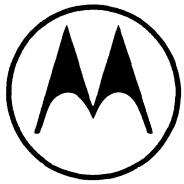
DL132R1

MOTOROLA INC.

SINGLE-CHIP MICROCOMPUTER DATA



MOTOROLA

MICROCOMPUTERS

Prepared by
Technical Information Center

This book is intended to provide the design engineer with the technical data needed to completely and successfully design a microcomputer based system. The data sheets for Motorola's microcomputer and peripheral components are included.

The information in this book has been carefully checked; no responsibility, however, is assumed for inaccuracies. Furthermore, this information does not convey to the purchaser of microelectronic devices any license under the patent rights of the manufacturer.

Additional information about memory products, technical training, and system development products is also provided. For further marketing and applications information, please contact:

Motorola Inc.
MOS Integrated Circuits Group
Microprocessor Division
Austin, Texas
(512)928-6800

Series C
Second Printing

Printed in U.S.A.

Product Preview data sheets herein contain information on a product under development. Motorola reserves the right to change or discontinue these products without notice.

Advance Information data sheets herein contain information on new products. Specifications and information are subject to change without notice.

TABLE OF CONTENTS

MASTER INDEX

This index includes all devices in Motorola 8-bit Microcomputer and Microprocessor/Peripheral product line. Devices with MPU in the page number column are fully characterized in the separate **Microprocessor Data Book.**

Motorola's Microcomputer Families

1

MOTOROLA'S MICROPROCESSOR AND MICROCOMPUTER FAMILIES

Serving as the "heart" of every microcomputer system is a microprocessor. Motorola manufactures the industry's most complete selection of solid-state microcomputer components to provide the performance you need and the design flexibility you want.

The family concept has been extremely popular in the microprocessor industry. Motorola pioneered this family concept with the introduction of the M6800 Family in 1974. Since then the MPU/MCU Family has evolved in several directions, as shown in Figure 1-1, in order to fill expanding use concepts. In addition, the basic M6800 Family has been enhanced. A large number of peripheral devices have been developed to support the expanding family of microprocessors and microcomputers.

SINGLE-CHIP MICROCOMPUTERS (MCUs) THE M3870 AND THE M6801 — M6804 — M6805 FAMILIES

Take a basic MPU; add an on-chip clock oscillator and timer, put in enough Read-Only Memory (ROM) to handle the program routines for dedicated application, and enough Read/Write (RAM) Memory capacity to handle the associated data manipulations; cap it off with sufficient input/output capability to interface with a number of parallel and serial oriented peripherals and you have a single-chip microcomputer.

The single-chip system doesn't necessarily have all the flexibility of a multi-chip system, but with adequate capacity to handle a specific requirement, it can save both component cost and equipment manufacturing cost. Motorola offers single-chip microcomputers across a broad spectrum of processor performance and system functionality. Motorola's first high volume production single-chip MCU is the second source of the popular 3870. The M6801 Family includes the high performance single-chip MCU, plus EPROM and ROM-less versions. The rapidly expanding M6805 Family includes a number of memory and package sizes with various special I/O functions, in both HMOS and CMOS. The M6804 Family now provides the 8-bit processing capabilities that compete in the 4-bit price arena!

PERFORMANCE — Processor performance, or program efficiency, for the application is an important single-chip MCU selection criteria. The M6801 Family is the throughput leader with 16-bit data operations, binary mulitply, and an average of only 3.7 cycles per instruction. Bit modify and test instructions and powerful indexing modes put the M6805 Family in second place on the performance scale. The MC3870 also offers a very successful 8-bit architecture. The MC6804 Family offers the proven capability of the M6800-based instruction set.

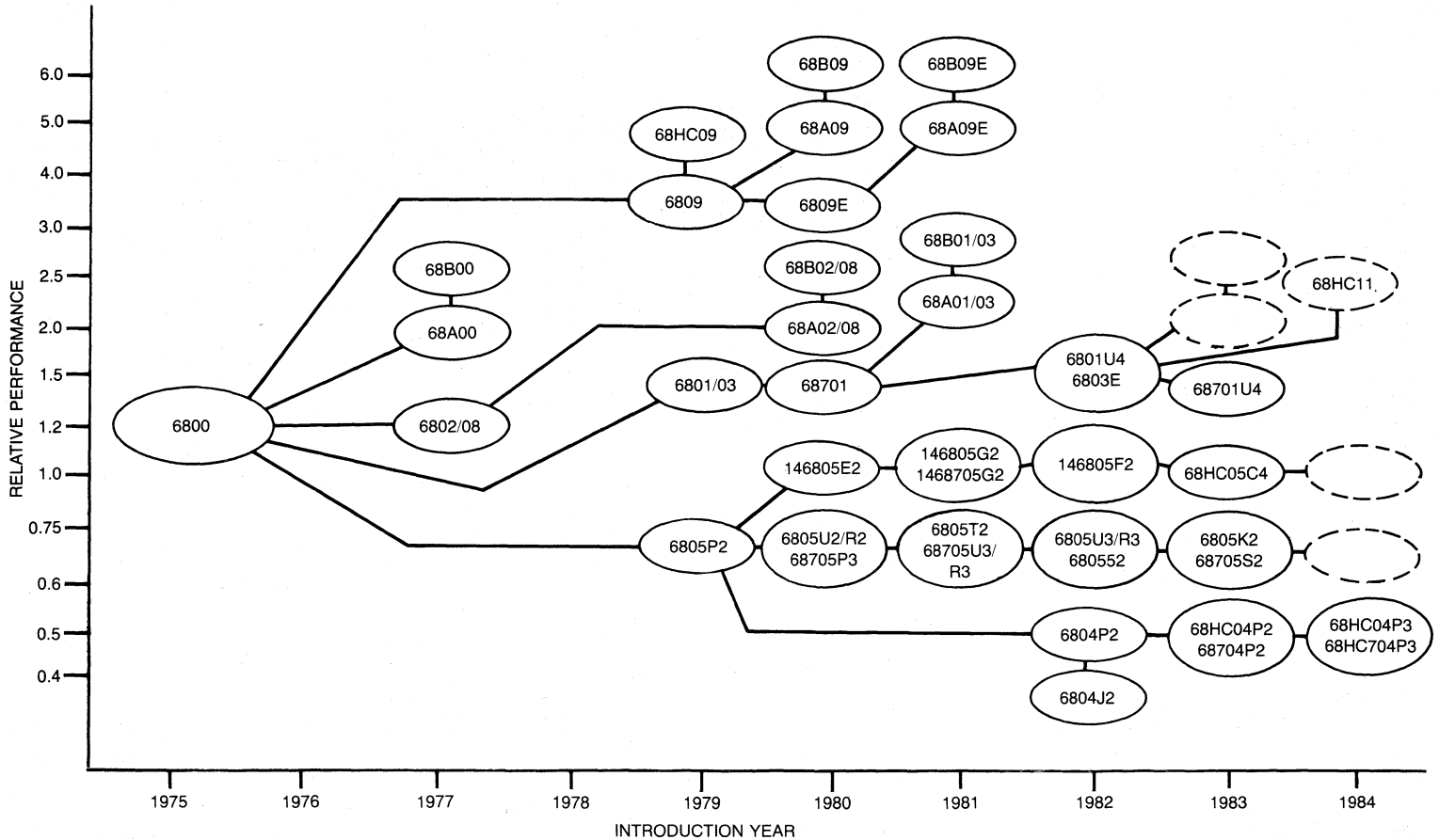


FIGURE 1-1. GENEALOGY OF THE COHESIVE M6800 MICROPROCESSOR/MICROCOMPUTER FAMILY

TECHNOLOGY — The very high production volumes of high-density NMOS (HMOS) permit low cost single-chip solutions. CMOS, as a relatively new microcomputer technology, offers very low power consumption and wide power supply tolerance at performance levels similar to HMOS. The M6801 Family, M6805 Family, and MC3870 are produced in HMOS while the M6805 Family makes CMOS benefits available. The M6805 Family is the first microcomputer that allows you to look at the technology trade-offs independent of the architectural and supplier choices. The new M6804 Family is available in both the HMOS and CMOS technology.

SINGLE-CHIP MICROCOMPUTERS, SELECTOR GUIDE BY TECHNOLOGY

ROM SIZE — The mask ROM capacities of the present single-chip MCUs range from 1K byte for the M6805 and M6804 Families, up to 4K bytes on the M6801 Family version. However, the M6801 and M6805 Families may in the future be implemented with as much as 64K bytes of on-chip ROM without any architectural changes. In selecting the ROM size, the ROM usage efficiency of the instruction set should be considered, along with the application to be programmed. The architecture of the MC3870 class offers short one- and two-byte instructions. The M6801 and M6805 Families use many multi-function instructions such as bit manipulation, memory modification, indexing, and multiply to do the function of two or more instructions in traditional MCUs.

NON-MASK-ROM VERSIONS — EPROM versions and/or ROM-less versions of practically all single-chip MCUs are offered. They serve for limited to high volume applications, prototype debugging, and field trials. EPROM versions are available in the M6805 and M6801 Families. ROM-less versions are offered in the M6801 and M6805 Families.

RAM SIZE — On-chip RAM sizes range from 32 bytes in the M6804 Family to 192 bytes in the M6801 Family. Between these present limits are the M6805 Family versions of 96, 112, and 176 bytes. Architectures such as the M6801 and M6805 Families which permit multi-level subroutines plus ROM and RAM data tables allow you to trade-off ROM and RAM utilization. ROM usage can be minimized with subroutines and look-up tables, while RAM use can be optimized with ROM tables and fewer subroutines.

DIGITAL I/O — Single-chip MCUs are available in 40-pin dual-in-line packages as well as the smaller (and lower cost) 28-pin packages. All these MCU families include 40-pin versions, while the M6805 Family has 28-pin members. Five to seven pins serve power and control functions permitting up to 23 I/O pins in a 28-pin package and up to 34 I/O pins in 40-pin versions (including interrupts, timers, and special I/O functions). All of the MCUs offer essentially any mix of inputs and outputs. Higher output drive current is available in the M6805 Family.

EXPANSION BUS — The ROM-less versions include a bus to access off-chip program memory and additional I/O. However, the M6801 Family single-chip MCUs also include three bus structure modes for off-chip expansion. The three bus modes permit the number of bus pins to be otimized for the amount of address space needed off-chip.

INTERRUPTS — When an application program must synchronize with two or more external events, interrupt hardware in some form is usually necessary. The M6801 and M6805 Families include fully automatic interrupts (registers are saved) with programmable vectors for both external pins and internal timers. The MC3870 interrupt scheme requires more program overhead.

TIMERS — On-chip timers are the most frequently used special I/O function. Timers may generate interrupts to a program at a periodic rate, measure external values, count external events, and generate measured output values. The M6801 Family includes a 16-bit timer that may be used to perform three of the above functions simultaneously. The M6805 Family timer consists of a programmable 8-bit counter and a selectable 7-bit prescaler. The MC3870 timer is 8 bits with a decimal prescaler.

SPECIAL FUNCTIONS — Various members of the MCU families include additional I/O functions. For example, the MC6801 Family includes a full 8-bit UART with baud rate generator on-chip. A 4-channel 8-bit A/D converter is included on a few M6805 Family versions. The digital portion of an RF frequency synthesizer is added to an M6805 Family member.

DEVELOPMENT SUPPORT — All three families are fully supported on the EXORciser development system. Included are assemblers, keyboard debugging including breakpoints, user system emulation, and stand-alone emulation. The M6801 Family has the added benefit of various high level languages and compatibility with MC6800 programs.

THE CMOS M6805 COMPONENTS

Motorola offers an 8-bit CMOS processor in the MC146805E2. The CMOS portion of the M6805 Family of 8-bit microprocessors, peripherals, and single-chip microcomputers combines the low power characteristic of CMOS, with the application flexibility of the M6800 Family.

The M6805 Family has evolved from the M6800 Family. The M6805 Family includes similar programmable bidirectional I/O, flexible memory organization, many memory reference instructions, interrupts, and multi-level subroutine nesting. ROM use efficiency, bit manipulation instructions, and improved table look-up indexing are M6805 Family enhancements of the M6800 heritage.

The benefits of CMOS are added to Motorola's microprocessor repertoire. Low operating power and even lower standby power consumption permit battery operation, cut cooling costs, and reduce power supply expense. The wider operating voltage range of CMOS offers higher noise immunity and easier switching to standby power. Static CMOS parts permit true standby operation plus power optimization with lower frequencies and voltages.

PROGRAMMING — The enhanced M6800 architectural features make the M6800 Family easy to program. The stack pointer permits up to 32 subroutine levels. Three ROM-efficient indexed addressing modes allow for look-up tables anywhere in memory. Any I/O pin or RAM bit may be modified with a single instruction. A branch may be taken depending upon the bit state of any I/O pin or RAM bit with only a single instruction. RAM, ROM, and I/O registers are all accessed with the same powerful memory addressing instructions. An efficient instruction set permits programs to be written faster, more easily optimized, and, therefore, more reliable.

INTERRUPTS — Real-time applications require sensing, measuring, and controlling system events. Five vectored interrupts, which stack the program registers, are included in M6805 Family processors to implement these applications. For time dependent tasks, a programmable 8-bit counter generates an interrupt when zero is reached. The timer includes a program-selectable 7-bit prescaler and a software selectable input. The timer input may be an external signal for pulse width measurement, or the on-chip oscillator. An external interrupt pin is also provided. Software techniques for external event synchronization are not needed.

MOTEL — The MOTEL concept (for MOtorola and InTEL bus compatibility) allows both types of processors to be interchanged on a bus without changing the design of the peripheral/memory system. The MOTEL circuit automatically detects which type of processor is connected, and interprets the bus control signals appropriately. The MCM65516 2K CMOS ROM, MC146818 Real-Time Clock plus RAM, and MC146823 Parallel Interface incorporate the MOTEL concept to provide a high degree of system flexibility.

SINGLE-CHIP CMOS MICROCOMPUTERS — Dedicated single-chip MCUs are also included in the M6805 Family. The MC146805F2 has 1K byte of on-chip ROM, while the MC146805G2 has a 2K ROM. The MC146805G2 also includes 112 RAM bytes, 32 input/output lines, programmable timer, external and timer interrupts, and high current output pins. The 1K MC146805F2 has the same interrupt features but fewer I/O lines, 28 pins, and less RAM, 64 bytes. The MC146805E2 microprocessor serves as the ROM-less prototyping part for both single-chip MCUs. The MC68HC05C4 has 32 I/O lines and 176 bytes of RAM. The MC68HC11A4 has A/D, 512 bytes of EEPROM, 256 bytes of RAM and 40 I/O lines. The MC68HC04P2 has 32 bytes of RAM and 20 I/O lines.

PERIPHERALS — Two types of CMOS peripherals are being added to Motorola's CMOS family. Parallel bus-oriented peripherals support microprocessors such as the MC146805E2,

while single-chip microcomputers are supported by port-oriented I/O, usually using serial data transfer. The MC146823 Parallel Interface offers three 8-bit ports (24 lines) of digital interfacing, including port latch control signals, to multiplexed-bus microprocessors such as the MC146805E2. The MC146818 Real-Time Clock plus RAM relieves the processor of maintaining the time and date, generates timed interrupts, and includes 50 bytes of CMOS RAM. Program memory is provided by the completely bus compatible MC65516 2K CMOS ROM. Other support circuits include LCD drivers (MC145000, MC145001, MC144115, and MC144117), LED drivers (MC14499 and MC144100), D/A converters (MC144110 and MC144111), A/D subsystem (MC14443 and MC14447), latches (MC14099, MC14597, MC14598, and MC14599), remote I/O (MC14469) and frequency synthesizers (MC14156 and MC145144).

POWER SAVINGS — Energy efficiency is, of course, the chief CMOS attraction. CMOS MPUs are seriously considered anywhere a battery is used, whether it be the primary or a back-up power source. The operating current can be orders-of-magnitude lower. Standby modes can have power usages order-of-magnitude lower yet. Since the M6805 Family is static in design, low-speed operating current is extremely low.

STATIC DESIGN — The clock of a static CMOS microprocessor may be at any frequency below the specified maximum. CMOS users frequently lower the frequency, to conserve power, approaching the point where the processor is fully loaded during the worst-case program cycle. A static MPU allows operation at 1 kHz or 10 kHz in applications where battery drain is critical, and the workload light. A static processor can also be stopped during any cycle without losing any volatile information, which assures extremely low standby current.

PROGRAM CONTROL OF POWER — Typical CMOS microprocessor applications require considerable attention to minimizing power consumption. The M6805 and M6804 Families of CMOS processors include program control of power usage, as well as the traditional external power optimizing tools. The program may initiate either of two standby modes, called Stop and Wait, which halt program execution. The external or timer interrupts automatically turn the processor back on to allow execution to resume. Why not save power when the program has no work to do? The program can be restarted when there is work that needs doing. Battery drain is the *average* of operating and standby current for the average work duty cycle.

LOW POWER DISSIPATION — A major side benefit of low power usage is that the heat dissipated is also low. The costs of cooling equipment is not needed. Fan noise in an office environment, as well as fan unreliability, need not be endured. Systems may be enclosed in smaller housings. Air tight systems need not have special heat conducting mechanisms.

WIDER VOLTAGE RANGE — The initial CMOS MPU products are characterized to operate from 3.0 to 6.0 voltages. The voltage range is being extended to higher voltages in upcoming versions. The wider voltage range permits lower cost power regulation, easier switching to back-up sources, and lower cost batteries. The higher voltage parts add noise immunity to the wide voltage range benefits.

SINGLE-CHIP MICROCOMPUTER FAMILIES FEATURES SPECTRUMS

The following illustrations (Figures 1-2 through 1-5) represent the microcomputer families and their features.

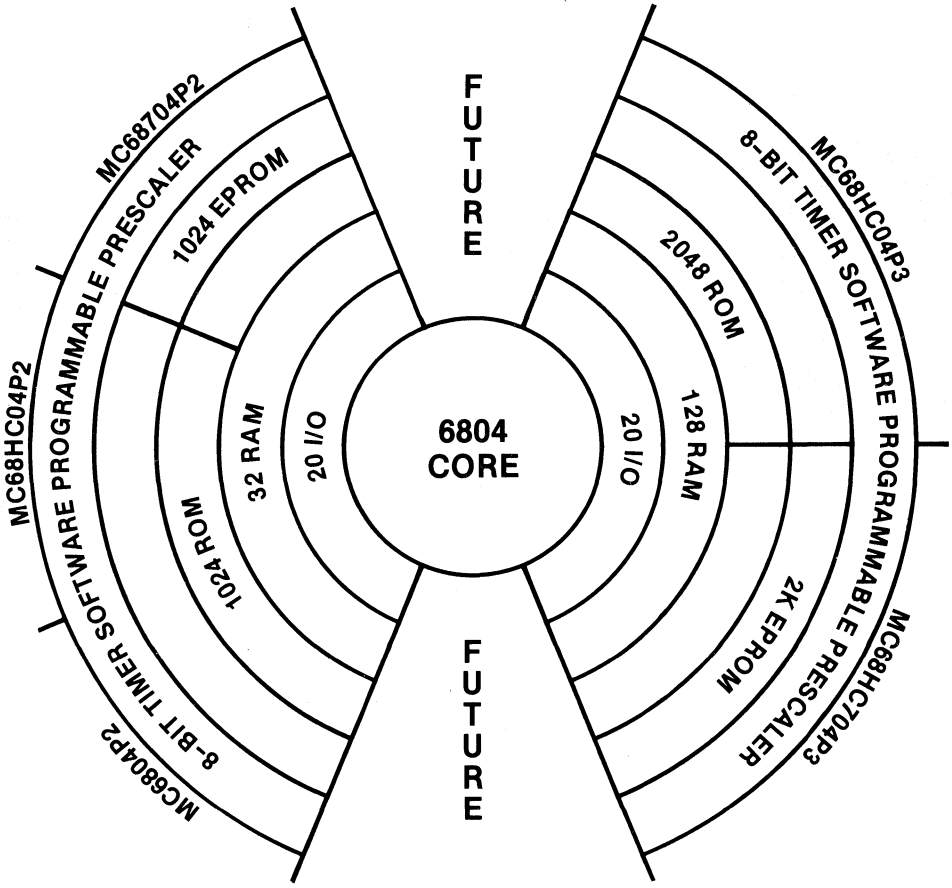
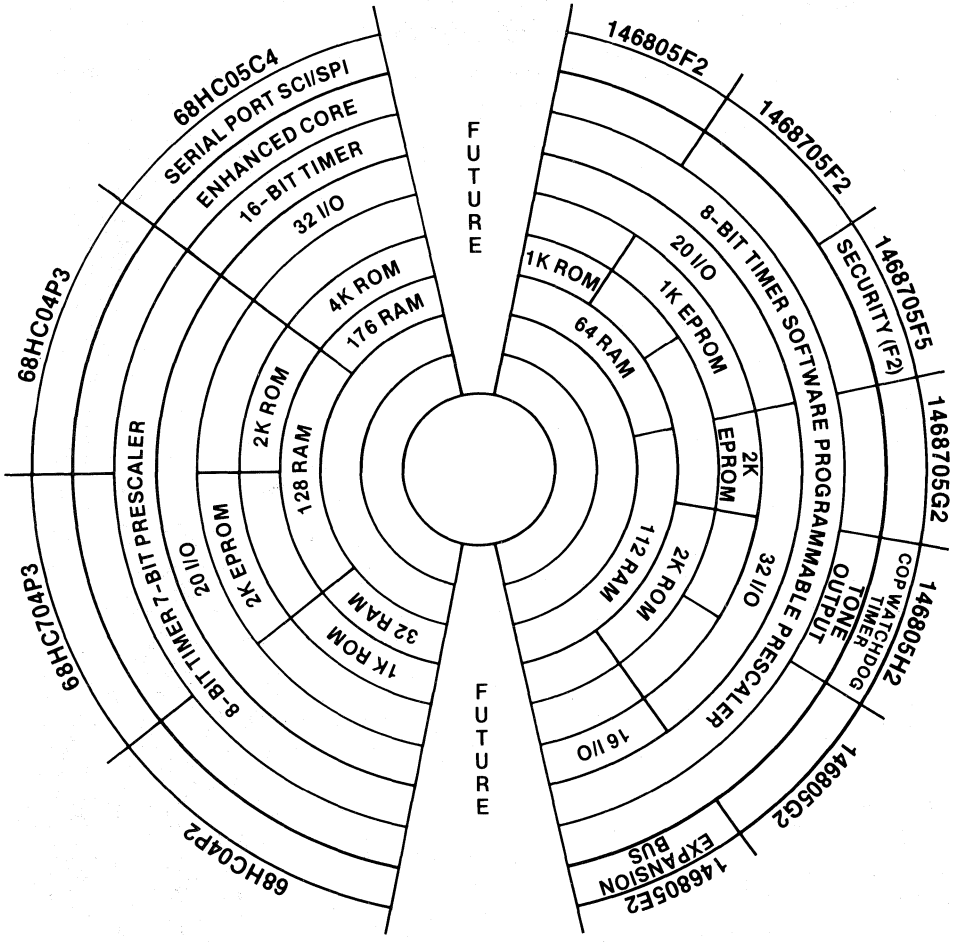


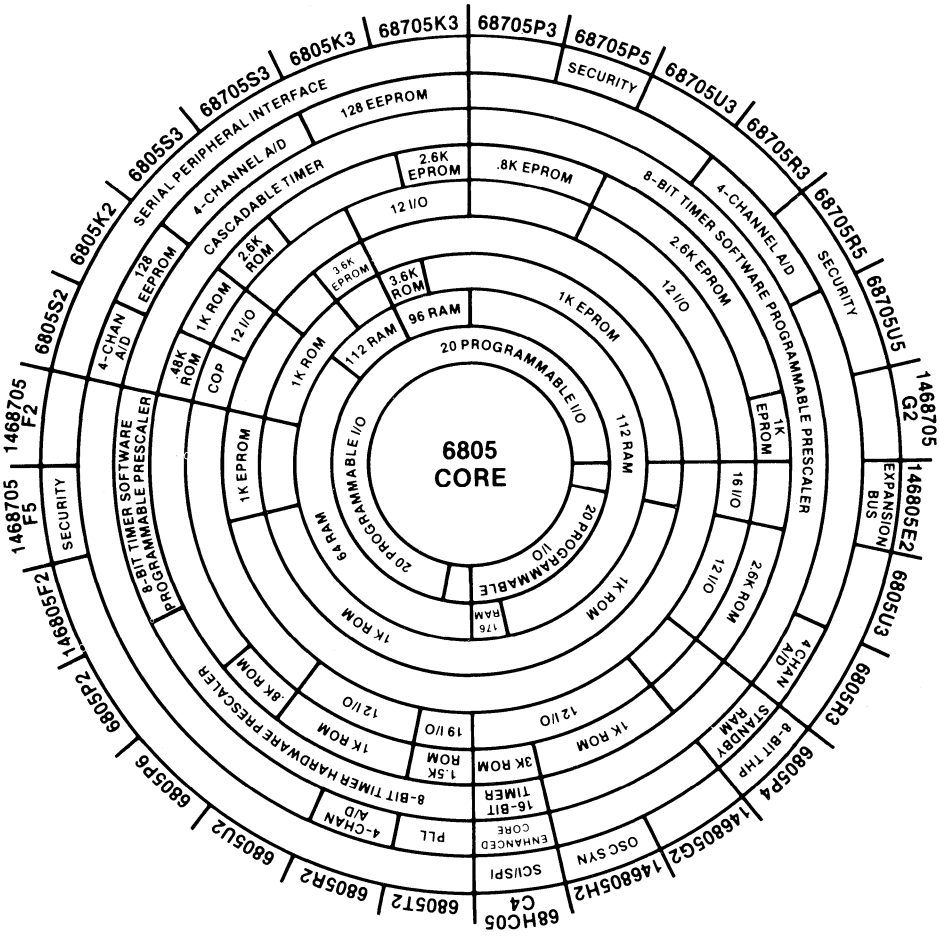
FIGURE 1-2. M6804 FAMILY SPECTRUM



LEGEND

SPI = SERIAL PERIPHERAL INTERFACE
SCI = SERIAL COMMUNICATION INTERFACE
COP = COMPUTER OPERATING PROPERLY

FIGURE 1-3. CMOS M6805 AND CMOS M6804 SPECTRUM

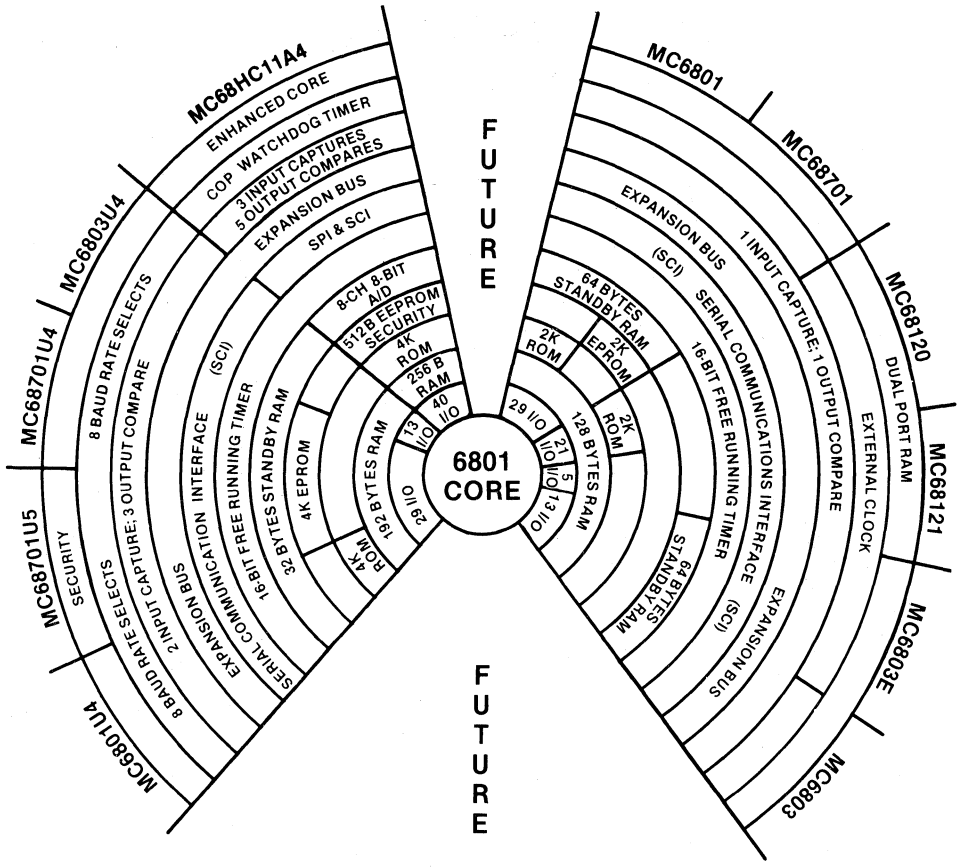


LEGEND

B = BYTE
CHAN = CHANNEL
COP = COMPUTER OPERATING PROPERLY
I/O = INPUT OUTPUT
OSC SYN = OSCILLATOR SYNTHESIZER
PLL = PHASE LOCK LOOP
SCI = SERIAL COMMUNICATION INTERFACE
SPI = SERIAL PERIPHERAL INTERFACE

8-BIT THP = 8-BIT TIMER HARDWARE PRESCALER

FIGURE 1-4. M6805 FAMILY SPECTRUM



LEGEND

B = BYTES
STBY = STANDBY
CH = CHANNEL
SPI = SERIAL PERIPHERAL INTERFACE
SCI = SERIAL COMMUNICATION INTERFACE
COP = COMPUTER OPERATING PROPERLY

FIGURE 1-5. M6801 FAMILY SPECTRUM

Reliability

2

RELIABILITY AND QUALITY MONITOR REPORT

OCTOBER 1983

Introduction

Motorola conducts extensive reliability tests to qualify devices, to evaluate process and material changes and to accumulate generic performance data. The results of these tests provide the basis for production decisions and the generation of reliability reports for customer use. The following report provides an overview of reliability testing on Motorola's MOS Microprocessor Components conducted during 1982. Included in the report are summary results of dynamic life testing and thermal performance testing for plastic and ceramic packaged devices, and moisture performance testing for plastic parts. Results of the tests are detailed below.

Dynamic Life

Dynamic life, or high temperature operating life, is performed to accelerate failures resulting from thermally activated defects. Failure mechanisms detected during life test include die related defects which occur during wafer processing and both die and package related defects which occur during assembly.

Stress is generated through the application of a 5 volt dynamic bias and an ambient temperature of 125°C. A dynamic bias is considered more effective than static bias for LSI Microprocessor devices because a large percentage of the chip can be continuously exercised. During life test, devices are exercised using a common mid-range frequency clock signal which is typically 500KHz or 1MHz.

Devices are electrically tested after 168, 504, and 1008 hours using computer controlled testers which employ functional patterns under worst-case supply and clock conditions. Pass/fail criteria are established for each circuit type based on functionality and data sheet limits for AC and DC parameters. Devices which fail to meet a test criterion are segregated by failure mode and data logged, and failure analysis is performed, when appropriate, to establish associated failure mechanisms.

Life test failure rates are calculated using the Chi-Square distribution and a 90% confidence level (see Appendix A). This 90% confidence level is more stringent than the 60% level used in the 1981 report. The accompanying increase in failure rates for individual device types is a result of tightening the confidence level and does not indicate a reduction in the reliability of the devices. Tables 1 and 2 summarize the 1982 dynamic life test data for MOS Microprocessors.

TABLE 1.
SUMMARY OF DYNAMIC LIFE TEST RESULTS

Technology	Device Type	Test Devices	125°C Device Hours	70°C Equivalent Device Hours	Failures	Failure Rate* FITs
NMOS	MC6800	45	45,360	2.2×10^6	0	1050
	MC6810	90	89,040	4.6×10^6	2	1150
	MC6821	448	451,584	24.1×10^6	0	100
	MC6822	83	83,664	4.9×10^6	0	470
	MC6840	45	45,360	2.5×10^6	0	920
	MC6844	45	45,360	2.7×10^6	0	860
	MC6845	346	346,752	19.5×10^6	2	270
	MC68652	45	45,360	1.9×10^6	0	1200
	MC68653	134	135,072	5.3×10^6	0	440
	MC68661	45	45,360	2.5×10^6	0	920
TOTAL		1,326	1,332,912	70.2×10^6	4	110
HMOS	MC6801	704	702,672	27.1×10^6	3	250
	MC6805P2	224	212,352	9.7×10^6	0	240
	MC6805R2	171	170,520	10.1×10^6	1	370
	MC6805U2	86	80,808	3.0×10^6	0	770
	MC6809	225	225,960	6.3×10^6	1	580
	MC68000	262	262,080	15.0×10^6	2	350
	MC68008	168	169,344	6.8×10^6	0	340
	MC68230	126	120,456	7.0×10^6	3	960
	MC68451	88	88,704	4.8×10^6	0	480
	MC68705P3	268	265,248	15.3×10^6	2	340
TOTAL		2,322	2,298,144	105.1×10^6	12	170
CMOS	MC141200	135	135,576	14.1×10^6	1	270
	MC146805E2	89	83,352	8.8×10^6	0	260
	MC146805G2	178	171,192	17.2×10^6	3	390
	MC146818	89	88,872	7.4×10^6	0	310
TOTAL		491	478,992	47.5×10^6	4	170
GRAND TOTAL		4,139	4,110,048	222.8×10^6	20	120

*90% Confidence Level

TABLE 2.
MICROPROCESSOR FAMILY DYNAMIC LIFE TEST RESULTS

	Total Devices	125°C Device Hours	70°C Equivalent Device Hours	Failures	Failure Rate* FITs
WAFER PROCESS TECHNOLOGY					
NMOS	1,326	1,332,912	70.2×10^6	4	110
HMOS	2,322	2,298,144	105.1×10^6	12	170
CMOS	491	478,992	47.5×10^6	4	170
PACKAGING SYSTEM TECHNOLOGY					
Ceramic	1,875	1,858,176	104.3×10^6	12	170
Plastic	2,264	2,251,872	118.5×10^6	8	110
TOTAL	4,139	4,110,048	222.8×10^6	20	120

*90% Confidence Level

SUMMARY:

The overall life test results for 1982 show a very significant improvement over our 1981 data base (Reliability Report 8238). For 1982 we tightened our confidence level from 60% to 90%. The failure rate for 1982 was 120 FITs at a 90% confidence level as compared with 250 FITs at 90% confidence level for 1981. The major effect of tightening the confidence level from 60% to 90% is to increase the predicted failure rate of individual devices with limited device hours. For example, the predicted failure rate for the MC6800 using 60% confidence is 420 FITs. The predicted failure rate for this same device using the 90% confidence is 1050 FITs, or more than double. This makes a statistically significant comparison of the individual device failure rates very difficult. It is more beneficial to examine the failure rate of the process technologies (NMOS, HMOS, CMOS) or the packaging technologies (plastic and ceramic) in which there are a considerable number of device hours which reduce the impact of the confidence level change. Even with the statistical tightening for 1982, the process and package technologies have achieved a reliability improvement as measured by dynamic life test when compared with the 1981 data base.

Plastic Package Environmental Performance

The use of plastic encapsulation for packaging of integrated circuits has met with widespread customer acceptance throughout the semiconductor industry because it is lighter, less expensive, and more resistant to physical damage than ceramic packaging. However, there are several reliability concerns in plastic packages: contamination, moisture resistance, wirebond integrity, and thermal performance. Dynamic life test results show no significant difference between plastic and ceramic device performance; this demonstrates that Motorola's careful selection of materials and rigid control of processes has eliminated any plastic-related performance degradation. The following section addresses the other reliability concerns of plastic parts: corrosion, wirebond integrity, and thermal performance.

Moisture Related Performance

In plastic integrated circuits, moisture present in the package can cause an increase in the corrosion rate of the die metallization, if ionic contaminants are present, resulting in failures when the device is in use. Moisture may reach the interconnect metallization along the leadframe-molding compound interface or through the bulk of the plastic. The combination of moisture, ionic contaminants carried in with the moisture or present in the plastic, and an electric field creates an electrolytic cell which becomes a corrosion site.

To help prevent corrosion problems, Motorola uses a molding compound which forms a compressive bond around the leadframe which, when cured, produces a tight seal to minimize microgaps. Tighter control of contamination sources throughout the manufacturing process, improvements in passivation and improved metallization techniques have resulted in lower defect density and more complete passivation coverage, keeping moisture from penetrating to the die surface.

Two accelerated tests are used by Motorola to assess the level of performance achieved by the combined application of these corrosion-prevention measures: Autoclave and Temperature Humidity Bias (T.H.B.). 1982 moisture performance test results are detailed below.

Autoclave

Autoclave testing uses a combination of temperature, humidity, and pressure to accelerate moisture ingress along the leadframe-molding compound interface path. The absence of a bias keeps device power dissipation from acting as a moisture barrier, increasing the probability that moisture will reach the die if a part is defective.

Autoclave test conditions include 121°C, 100% relative humidity and 15 psig. Each test sample is selected from a separate assembly lot and subjected to a minimum of 96 hours of stress; complete parametric and functional tests are performed on all devices at each readpoint. In addition, some devices are stressed for an additional 48 hours. All electrical failures are included in the data base, not only those associated with corrosion on the die. Autoclave test results for 1982 are summarized in Table 3.

TABLE 3.
AUTOCLAVE TEST RESULTS
121°C 100% R.H. 15 psig

Hours	48	96	144
Failures/Sample	6/3083	1/3076	2/1399
Percent Defective	0.19	0.03	0.14
Cumulative Percent Defective	0.19	0.22	0.36

Temperature Humidity Bias

Temperature Humidity Bias (T.H.B.) testing is used to evaluate the moisture resistance of plastic devices by employing the severe conditions of 85°C, 85% relative humidity, and 5 volts to accelerate corrosion of the metallization. The biasing circuits used in T.H.B. testing create static electric fields between adjacent pins and metallization stripes, maximizing the effect of electrolytic cells while minimizing the power dissipation. A typical T.H.B. biasing scheme would include: all I/O or output pins either open or with resistive terminations; enable pins are disabled; and all other pins have alternate VDD and VSS on adjacent pins. As with autoclave, the expected failure mode is corrosion of the die metallization.

Each T.H.B. sample is sourced from a separate assembly lot and tested for a period of 1008 hours. Complete parametric and functional test programs are typically performed at the 168, 504, and 1008 hour read points using computer controlled testers. The pass/fail criteria used for life test are also employed with T.H.B. samples. A worst-case analysis is presented since all electrical failures are considered instead of only those associated with corrosion mechanisms. Results for 1982 are summarized in Table 4.

TABLE 4.
TEMPERATURE HUMIDITY BIAS TEST RESULTS
85°C 85% R.H. 5.0 VOLTS

Hours	168	504	1008
Failures/Sample	2/1456	4/1796	5/1781
Percent Defective	0.14	0.22	0.28
Cumulative Percent Defective	0.14	0.36	0.64

A Weibull plot (Figure 1) shows the continued improvement in T.H.B. performance as measured in 1979, 1980, 1981 and 1982.

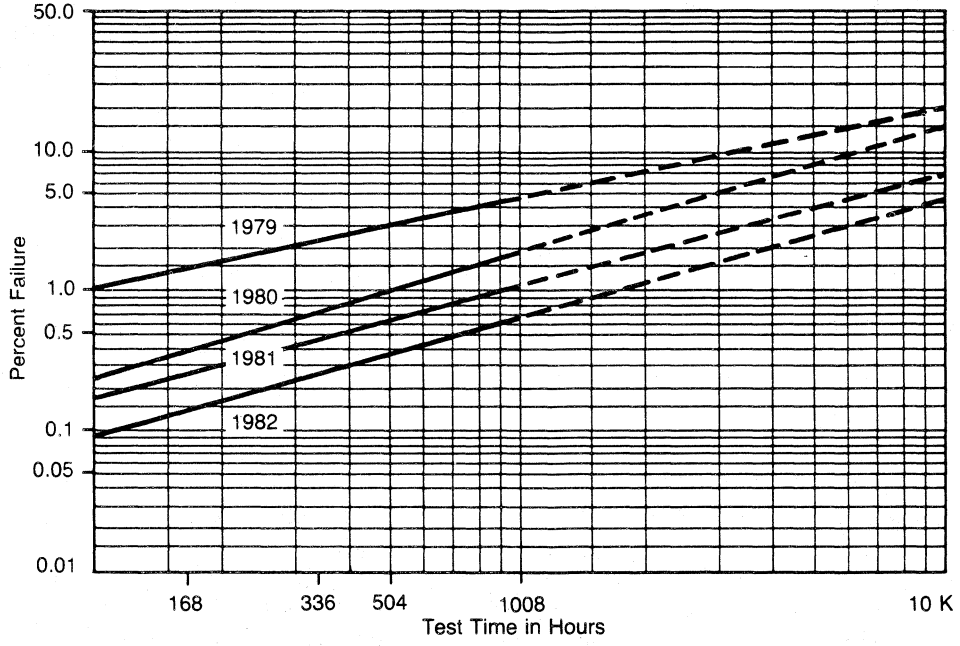


FIGURE 1. WEIBULL PLOT OF TEMPERATURE HUMIDITY BIAS TEST RESULTS

Thermal Cycling Performance

Thermal cycling accelerates the stressing effects of thermal expansion mismatch between the various components of the plastic and ceramic packaging systems through rapid successive excursions to high and low temperature extremes. Temperature cycle and thermal shock are two tests which are used to determine the effects of these stresses on package integrity, especially wire bond and die bond integrity. These types of failure modes follow the classical wearout mechanism pattern (i.e. an increasing failure rate with increased cycles of exposure.)

Temperature Cycle

The integrity of wire bonds and die bonds in plastic packages can be accurately evaluated through temperature cycle testing. *Military Standard 883B,* Method 1010.4, Condition C is employed to permit easy comparison of results with other industry sources.

Devices are inserted into the cycling system and held at −65°C for at least ten minutes. Following the cold dwell, devices are heated to 150°C during a transition time of five minutes maximum, after which devices dwell at 150°C for a minimum of ten minutes. They are then cooled during a similar transition period to −65°C after which the cycle is repeated. The system employs a circulating air environment to assure rapid stabilization at the specified temperature. The dwell at each extreme, plus the two transition times, constitutes one test cycle (approximately 30 minutes).

Electrical measurements and high temperature continuity tests are typically performed after 100, 500 and 1000 cycles. The predominant failure mechanism in the ceramic packaged product is wire bond breakage above the ball near the die where the heat and stress of the bonding process reduce the strength of the wire. The predominant temperature cycle activated failure mechanisms in plastic encapsulated circuits are die lift and die crazing/cracking due to inadequate die wetting/curing and mold compound stresses on the die, respectively. Results of the test are shown in Table 5.

TABLE 5.
TEMPERATURE CYCLE TEST RESULTS
−65°C to +150°C AIR TO AIR

Cycles	100	500	1000
Failures/Sample	7/3103	5/3081	8/3050
Percent Defective	0.23	0.16	0.26
Cumulative Percent Defective	0.23	0.39	0.65

Thermal Shock

Thermal shock is an environmental test performed in accordance with *Military Standard 883B,* Method 1011.3, Condition C. The objective of this test is the same as that for temperature cycle — to emphasize differences in expansion coefficients for components of the packaging system. However, thermal shock provides a more severe stress than temperature cycle in that the devices are exposed to a more sudden change in temperature due to the higher thermal conductivity and heat capacity of the liquid ambient.

Devices are placed in a fluorocarbon bath cooled to −65°C. After being held in the cold chamber for at least five minutes, the sample is transferred in less than ten seconds to an adjacent chamber filled with fluorocarbon at 150°C and held for an equivalent time. The dwell time at each endpoint, plus the total transition time, constitutes one test cycle (approximately ten minutes). Thermal shock endpoint electrical measurements and high temperature continuity tests are typically performed at 100, 500, and 1000 cycles. Results of thermal shock tests performed in 1982 are shown in Table 6.

TABLE 6.
THERMAL SHOCK TESTS RESULTS
−65°C TO +150°C LIQUID TO LIQUID

Cycles	100	500	1000
Failures/Sample	1/941	1/967	9/955
Percent Defective	0.11	0.10	0.94
Cumulative Percent Defective	0.11	0.21	1.15

Conclusions

Reliability testing performed by Motorola MOS Microprocessor Division during 1982 has produced excellent results. The specific test results included in this report are representative of Motorola MOS Microprocessor components expected field performance. Failure rate estimates have been based on the outcome of tests and data analyses which are widely accepted. Life test failure rates on both ceramic and plastic packaged devices are significantly reduced over those reported previously. Moisture resistance testing indicates extremely high performance of Motorola MOS Microprocessor plastic encapsulated circuits. Thermal integrity testing shows that there are few failures, which typically occur only after extensive exposure to temperature extremes greater than those seen in field applications. The level of performance predicted by these test results is among the best available in the industry and far exceeds the requirements of most applications. Comparison to previous reports (Reliability Report 8238) verifies a history of continuous improvement which has made Motorola MOS Microprocessor components the optimum choice for reliable performance.

Copies of this and other reliability reports may be obtained from your local Motorola representative. For additional information contact Microprocessor Reliability Engineering 512-928-6640 or write to:

MOS Microprocessor Reliability Engineering
Motorola Incorporated
3501 Ed Bluestein Blvd.
Austin, Texas 78721

APPENDIX A.
QUALITY AND RELIABILITY SYSTEM

A complete Reliability and Quality Assurance system is in place to monitor and control the performance of Motorola's MOS Microprocessor Components. Incoming Quality Control inspects starting wafers, masks, chemicals, package piece parts and molding compounds. Process Engineering and In-Process Quality Control perform step-by-step monitoring of the wafer process to check oxidation, diffusion, photolithography, ion implantation, polysilicon deposition, metallization, passivation, and other process operations. Final visual, class probe, and capacitance-voltage plots complete the wafer area inspections. Environmental monitors are also performed for air cleanliness, water quality, temperature and humidity.

In the assembly area, In-Process Quality Control performs monitors on equipment performance and gate inspections at the major process steps on all lots. The Outgoing Quality Control group continues this philosophy in the final test area by performing electrical and visual-mechanical gates on every lot. The electrical inspection, which consists of AC, DC and functional tests, is performed to a 0.1% (maximum) Acceptable Quality Level (AQL) sampling plan. The visual/mechanical inspection is also performed to a 0.1% AQL sampling plan. Any lot which fails either of these gates is returned to production for 100% rescreen. A Quality Engineering organization exists to approve final test programs and support the Outgoing Quality Control organization. Test programs are tailored to assure all required specifications are met or the devices are rejected.

The Reliability Engineering organization is responsible for performing qualifications of new designs and process changes prior to introduction. In addition, Reliability Engineering establishes and maintains monitor programs to assure processes stay in control once they are qualified. Results from these programs provide rapid feedback to correct problems as they occur.

Supporting these efforts is the Metrology Laboratory which includes both a Standards and a Calibration Laboratory to provide National Bureau of Standards traceability to all production measurements.

Also offering required support are a Chemical Laboratory with such equipment as a gas chromatograph/mass spectrograph and X-ray fluorescent systems for detailed incoming chemical analyses; a Surface Analysis Laboratory whose equipment includes a Scanning Electron Microscope (S.E.M.) and a Scanning Auger Microprobe (S.A.M.); and a Product Analysis Laborabory for detailed analyses of failure modes and mechanisms for Microprocessor devices.

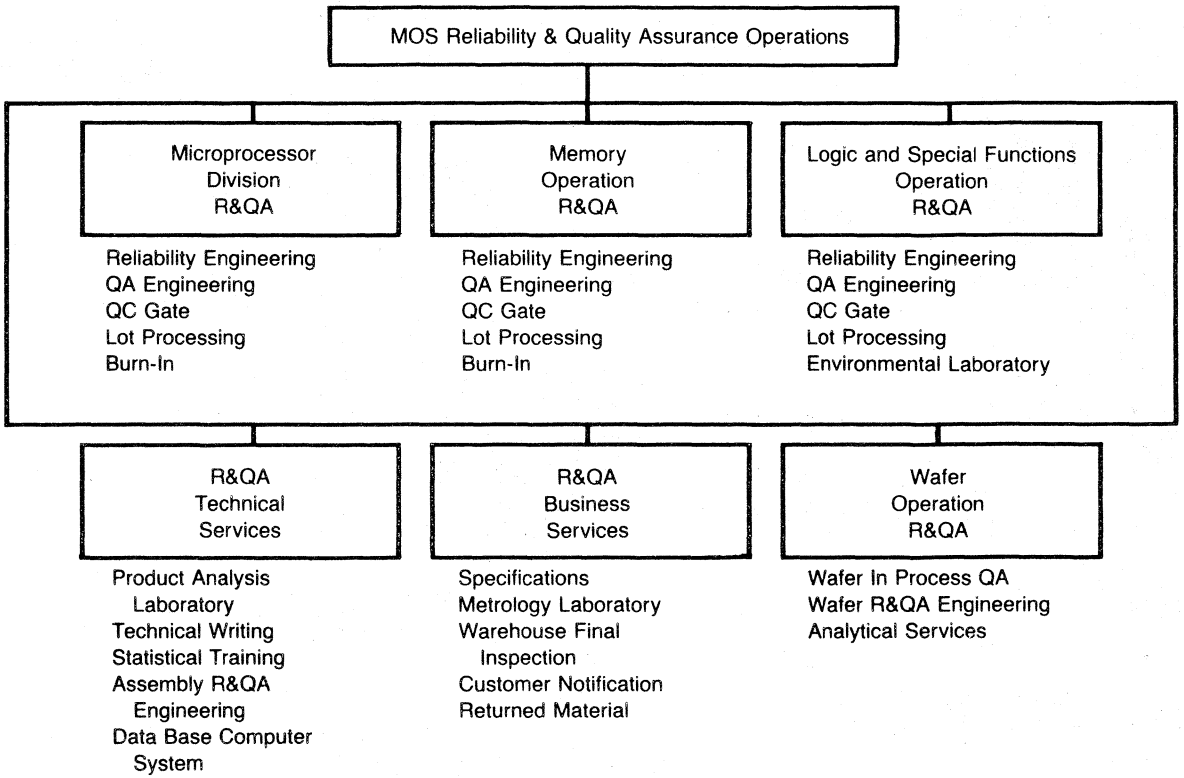


FIGURE A. RELIABILITY AND QUALITY ASSURANCE ORGANIZATION

APPENDIX B.
PACKAGING SYSTEMS

Motorola Microprocessor devices are produced in plastic, CERDIP and sidebraze packages. The ceramic package types are hermetically sealed to protect the integrated circuit from environmental factors and permit operation over extreme temperature ranges. Although plastic devices are not hermetic, modern epoxies exhibit extremely high moisture resistance, and long lifetimes may therefore be expected from these devices in typical environments.

Plastic

In recent years, plastic encapsulated devices have gained widespread acceptance throughout the electronics industry. Improvements in materials and process controls have resulted in significant improvements in reliability performance. In addition, plastic packages have the advantage of low cost and physical strength. Through careful selection of molding compound, leadframe material, and assembly methods, Motorola produces plastic packaged ICs with reliability suitable for nearly all applications.

Encapsulated integrated circuits incorporate the simplest processing and package construction of the various systems available. The die is attached to a leadframe, wire bonded and encapsulated using an epoxy novolac molding compound. The die may be attached to the leadframe by epoxy or by any of a variety of eutectic forming metal preforms. Wire bonding may be thermocompression or thermosonic, but the wire is always gold. This system has evolved from early industry experiments with aluminum ultrasonic wire bonding which experienced high rates of opens and intermittents. The encapsulant is the most critical component of the system since it controls contamination, moisture resistance, and stress effects. Epoxy novolacs have become the industry standard molding compound since they combine excellent characteristics in all these areas.

The plastic package is, by far, the most resistant to physical damage since the die is completely encapsulated and cavity hermeticity is not a concern. Since the package is light in weight and the plastic is less brittle than ceramic, chipping and cosmetic damage are not problems. The leadframe and plating are equivalent to CERDIP, and modern epoxies pose no danger from contamination.

In comparing plastic to ceramic packages, there are two characteristics to be considered: moisture resistance and thermal characteristics. Microprocessor plastic products perform very well on moisture resistance related tests. This is due to advances in molding compounds, and the characteristic low voltages and moderate power dissipation of Microprocessor products. In most instances, plastic devices will provide excellent performance, essentially equivalent to hermetic performance. Thermal resistance has been improved dramatically through the introduction of copper leadframes and heatspreaders. During 1982 and 1983, a large number of Microprocessor devices will be converted from Alloy 42 to copper leadframes to take advantage of the better thermal conductivity of copper. This results in lower junction temperatures, and subsequent improvements in electrical characteristics and reliability performance.

Another approach to lower thermal resistance for devices with high power dissipation is plastic assembly using a heatspreader. The heatspreader is an anodized aluminum piece part that sits below the plane of the leadframe. During the encapsulation process, the heatspreader is surrounded by plastic and becomes part of the package structure. Heatspreaders, when used in combination with Alloy 42 leadframes, yield a thermal resistance roughly equivalent to a copper leadframe plastic device, or to a ceramic device. Devices which contain a heatspreader employ the suffix "G" to designate this package type. The MC6801 Microprocessor Family has been offered in this package, and the 64-pin MC68000 16-bit Microprocessor is being offered in a heatspreader package.

Ceramic

Many users of integrated circuits continue to have requirements or preferences for hermetically sealed ceramic packages. These requirements are usually based on applications in a highly humid environment, increased temperature range or high power dissipation. Motorola produces two different types of ceramic packaged devices: CERDIP and sidebraze.

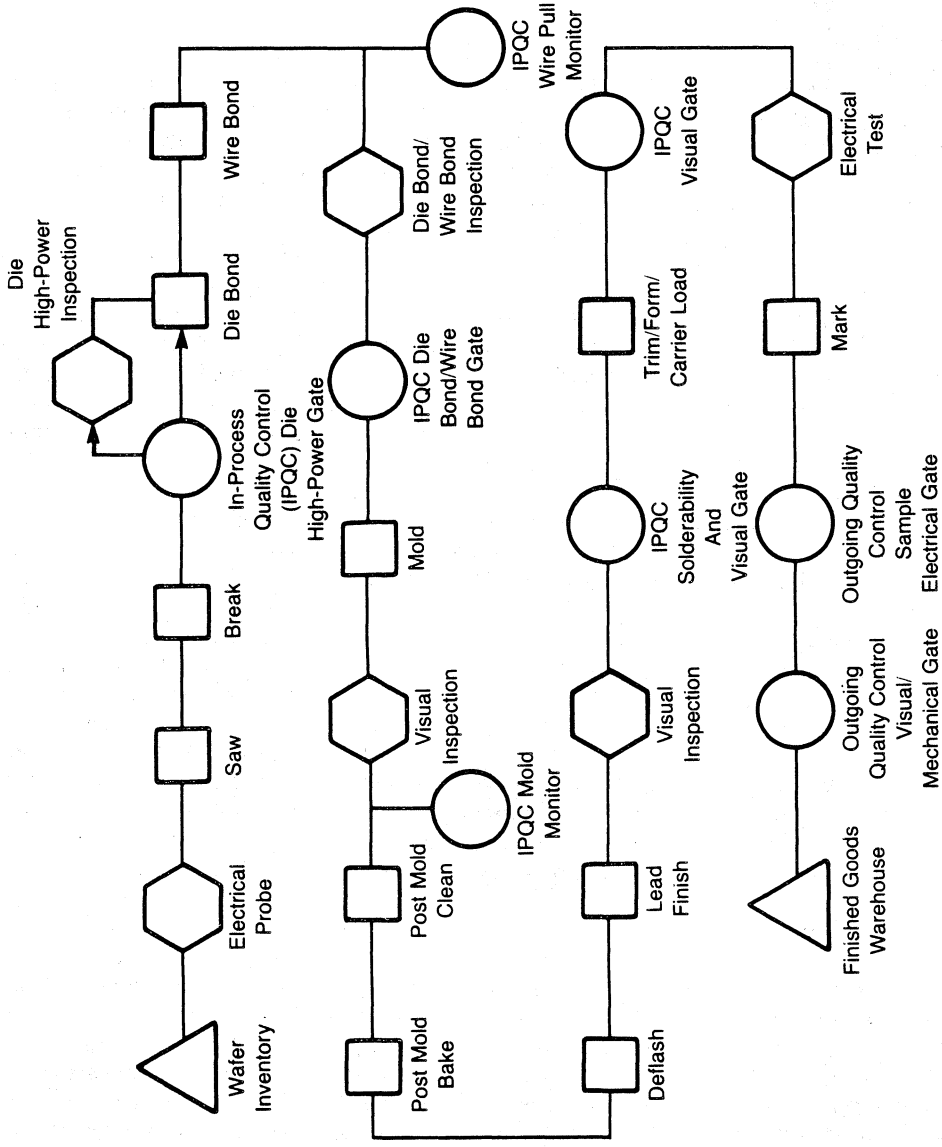


FIGURE B1. HERMETIC PROCESS FLOW

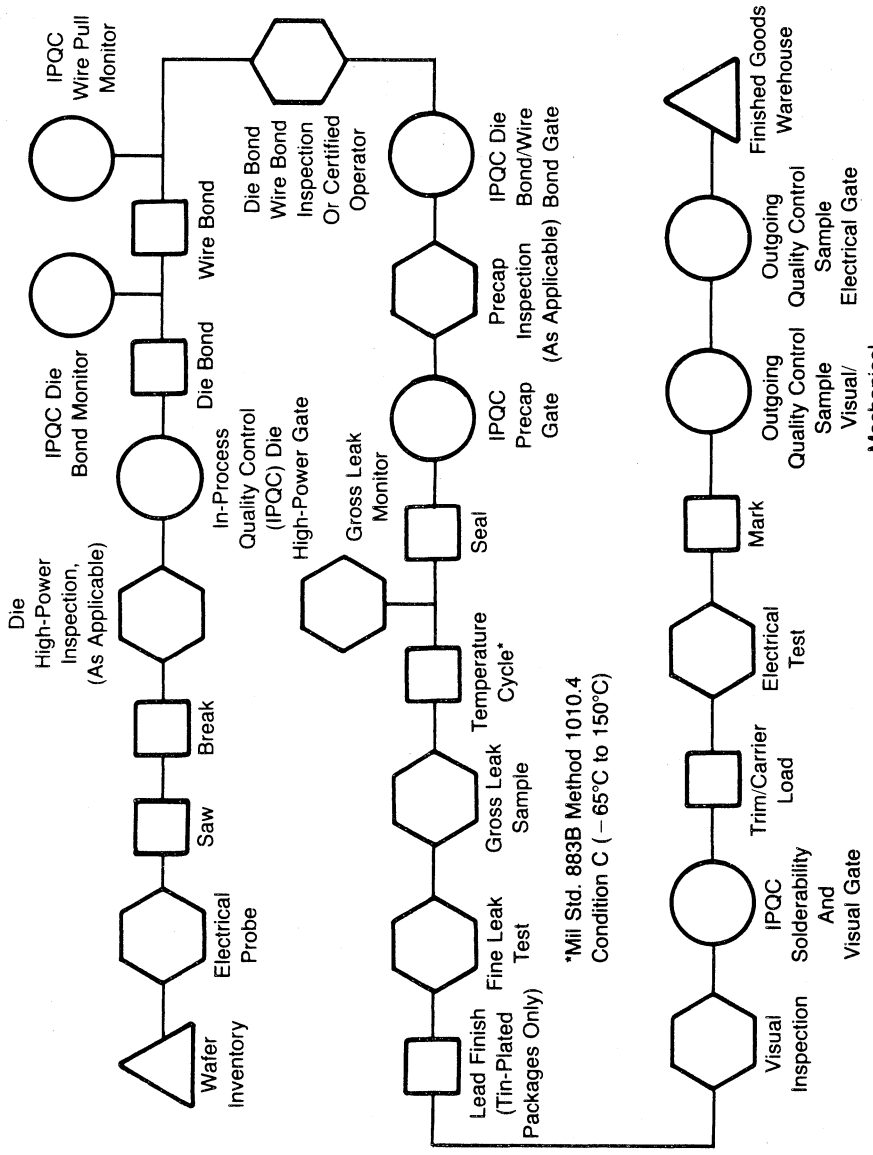


FIGURE B2. PLASTIC PROCESS FLOW

The sidebraze, or solder seal, package is composed of three layers of alumina which are screened with refractory metal such as tungsten or moly manganese and fired together to form the package body with a cavity for the die. The refractory metal is then plated and Alloy 42 leadframes are brazed to the bottom, sides or top of the package, depending on the vendor. The advantage of the sidebraze version is accurate lead alignment without the need for forming. The final piece part operation is plating which may be gold, or tin with a selective gold plate in the cavity. Although epoxy die bonding is feasible in this package — due to the higher sealing temperature, most manufacturers, including Motorola, employ a eutectic bond. Both aluminum ultrasonic wire bonding and gold thermocompression bonding are used.

Some tradeoffs exist in the performance characteristics of the two hermetic packages as they are offered by Motorola. Both typically are ceramic, hermetic, employ a eutectic die bond, use ultrasonic aluminum wire bonding, and have tin plating. The thermal resistance of the packages is very similar, with the sidebraze having a slight advantage. Both packages perform well on the standard thermal and mechanical environmental tests, but each is susceptible to handling damage. Loose shipping rail packaging or high velocity impacts during testing can chip the sidebraze package and sever the interlayer metallization. This type of handling will not affect the 10-mil-thick leadframe of the CERDIP package, but hermeticity failures can occur. The CERDIP package is slightly thicker and heavier, but no conductive surfaces are exposed so the shorting potential in dense packaging is reduced. Extensive testing of 24, 28, and 40 lead CERDIP and sidebraze devices has indicated no significant differences in reliability.

Some Microprocessor devices are now being offered in Leadless Chip Carriers (LCC). The primary advantage of LCCs is increased device density at the board or substrate level. Motorola currently uses a 40-pin LCC that is essentially identical to the sidebraze dual-in-line in construction characteristics and assembly methods. Some MC68000 16-bit family devices will be offered in higher terminal count LCCs, up to 68 terminals. Future plans include LCCs with single layer construction and other package types offering higher packing density at the system level.

APPENDIX C.
FAILURE RATE CALCULATIONS

Environmental tests are designed to measure device resistance to unusual and severe stress, not expected under normal operating conditions. Device performance under these conditions is expressed as a percent of devices defective and compared to previous results. Life tests, on the other hand, accelerate the use conditions of the device with temperature and voltage in a manner which is more quantitatively correlatable to system operation. Life test failure rates are expressed as failures per unit time and are calculated using established principles of probability and statistics.

The principles of reliability engineering have indicated that failure rates for semiconductor devices will take the form of the "bathtub" curve (Figure C1).

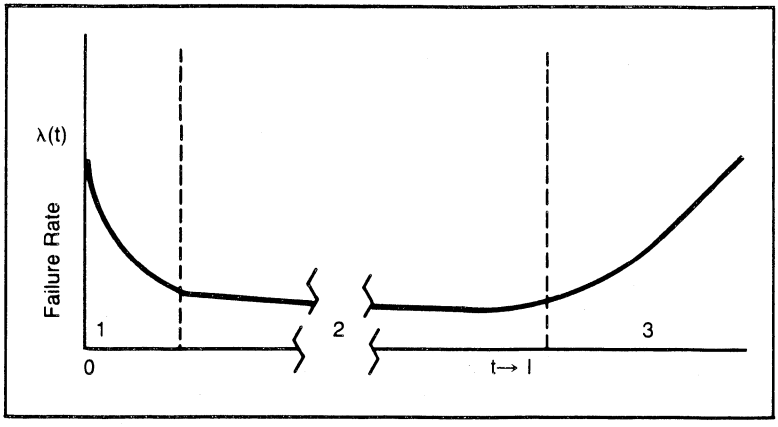


FIGURE C1. DEVICE FAILURE RATE AS A FUNCTION OF TIME.

The following three regions are represented in the curve:

1. *Infant Mortality* — a region of high but rapidly declining failure rates, usually associated with manufacturing defects.
2. *Random Failures* — a region of low, random failures caused by more subtle defects. This area of the curve represents the useful part of device life.
3. *Wearout* — a region of rapidly rising failure rates related to device wearout. Most semiconductors will not reach this stage before they are replaced because of changes in technology.

Techniques for calculating life test failure rates assume that the devices being tested have passed infant mortality and entered the stable random failure portion of the life curve. Failures which occur in this area are few and are known to approximate specific probability distributions. These probability distributions are used to calculate sample failure rates which can be projected to the population in general through the application of confidence limits. Techniques used to calculate life test failure rates for microprocessors are discussed below.

A failure rate for any sample of life tested devices can be determined by dividing the number of failures by the number of device hours. However, this rate will apply to that sample only. If you are interested in projecting from the sample to the populations in general, you must establish confidence limits. The application of confidence limits is a statement of how "confident" you are that the sample failure rate approximates that for the population in general. To obtain rates with different confidence levels it is necessary to make use of specific probability distributions which take the same form as the actual failure distribution.

It has been determined that failures in semiconductors that have entered the middle portion of the bathtub curve will approximate a Poisson distribution; this distribution applies when one has a large sample with an extremely small number of events of interest, such as device failures. Given a Poisson failure process, a Chi-Square distribution can be used to establish confidence limits for failure rates. Reliability Engineering has determined that the following general formula, which utilizes values from a Chi-Square table, can be used to calculate failure rates for semiconductors:

$$\lambda = \frac{1 \times 10^5}{MTTF} = \frac{\chi^2\,(\alpha,\ d.f.)}{2t} \qquad (1)$$

where:

λ = Failure Rate, %/1000 Hours
MTTF = Mean Time To Failure (Hours)
χ^2 = Chi-Square Function

$$\alpha = \frac{100 - \text{Confidence Limit}}{100}$$

d.f. = Degrees of Freedom = 2r + 2
r = Number of Failures
t = Device Hours

To calculate the failure rate, first determine the level of confidence you require and calculate degrees of freedom. Select the Chi-Square value from a Chi-Square distribution table with the appropriate degrees of freedom and confidence level. Divide that value by twice the actual device hours, at the temperature of interest.

The above formula applies for calculating a device failure rate, provided that the test is conducted at system temperature. However, since we are unable to observe long-term effects which develop over time, the test is accelerated through the application of a high temperature. In order to calculate a failure rate at the ambient temperature of a system, a factor must be supplied to compensate for the acceleration. The factor (Fa) which equates test temperature with rated temperature is derived from the Arrhenius relationship:

$$F_a = \exp\left((\theta/k) \cdot \left(\frac{1}{T_r} - \frac{1}{T_t}\right)\right) \qquad (2)$$

where:

F_a = Acceleration Factor
θ = Activation Energy, eV
k = Boltzman's constant, 8.62×10^{-5} eV/°K
T_r = Junction Temperature, °K at the Rated Ambient of 70°C
T_t = Junction Temperature, °K at the Life Test Ambient of 125°C

Motorola uses 70°C for the system temperature (To) to more closely approximate the actual temperature of the device during system operation and to supply a degree of conservatism to the failure rate calculation.

Motorola uses an activation energy (θ) value of 1.0 electron-volt. A 1.0 eV was selected as an average value because a variety of different failure mechanisms exist for microprocessor and other VLSI devices, with activation energies ranging from 0.40 eV for oxide related failures to 1.0 eV or greater for contamination and metal related failures. Tr and Tt of the equation are the average junction temperatures present at the rated and test ambients. Motorola uses junction, rather than ambient temperature, because they produce acceleration factors that are more conservative and representative of actual conditions. These temperatures are calculated as follows:

$$T_J = T_A + P_D \cdot \theta_{JA} \quad (3)$$

where:

T_J = Junction Temperature, °C
T_A = Ambient Temperature, °C
P_D = Average Power Dissipation, Watts
θ_{JA} = Thermal Resistance — Junction to Ambient, °C Per Watt

Once this step has been completed, the acceleration factor can be calculated and applied as a multiplier to the number of device test hours under accelerated test conditions to determine the equivalent number of hours at rated operating conditions. To determine the failure rate at the operating temperature, use equation (1) substituting the equivalent device hours at rated temperature for t in the equation.

Formula 1 provides a failure rate expressed in percent per thousand hours. This number, stated as a percentage per each thousand hours of operation, is one way Motorola Reliability Engineering expresses failure rates for Microprocessors. One other way of expressing failure rates is Failures In Time (FITs) which refers to failed units per 10^9 device hours (1 FIT = $\lambda \times 10^4$).

Mean Time To Failure (MTTF) is another parameter frequently used to express failure rates. MTTF is the average time to a failure of a non-repairable item such as a semiconductor and is expressed as the reciprocal of the failure rate:

$$MTTF = \frac{1}{\lambda} \quad (4)$$

APPENDIX D.
ELECTRICAL TESTING AND FAILURE CHARACTERISTICS

The electrical measurements performed on reliability test samples were obtained using computer controlled testers and programs employing exhaustive functional routines under worst-case supply and clock conditions. Devices which do not meet a test criterion, including those failing for parametric reasons, are first segregated into "bin outs" defined by the test program. A data log is obtained from which each failing device is then assigned to one of six failure mode categories. An analysis to determine specific failure mechanisms is performed when the level or pattern of failure indicates that it is appropriate. T.H.B. rejects are routinely decapsulated and inspected for corrosion of the metallization.

The electrical test programs are typically constructed in the following manner:

1. "Opens" test
2. "Shorts" test
3. Input Leakage
4. Functionality using nominal supply and input voltage levels and low frequency clock conditions
5. Functionality to data sheet parametric limits using worst-case combinations of VDD level and clock frequency
6. Three-state leakage
7. Output buffer current drive capability
8. Power dissipation test

Failure modes categorized according to these tests do not always indicate a specific problem and individual test programs may deviate from the sequence shown above as required for complete testing of the specific device type. Microprocessors and other LSI logic circuits do not readily lend themselves to the identification of failure modes since their complexity creates an astronomical number of possible combination, some of which are very subtle. Attempts to categorize these modes by the test sequence invariably result in groupings which are not mutually exclusive or related to physical mechanisms.

The distribution of failure modes and mechanisms observed during life testing appears to be the result of random manufacturing anomalies and does not, therefore, indicate trends correlatable to specific process or design deficiencies. These results are consistent with careful attention to process controls and reflect Motorola's high priority for quality and reliability.

TABLE D1.
FAILURE MODE CLASSIFICATION

A. **OPENS** — No electrical connection between an external terminal and corresponding die circuitry (possible intermitent). MOS inputs are normally high impedance parts and opens are detected by forward-biasing the substrate diode.

B. **SHORTS** — An unintended resistive path of relatively low value between one terminal and any other terminal.

C. **FUNCTIONAL** — A failure of one or more output terminals to respond with a correct logical state under nominal supply, clock, and VIH/VIL levels; a violation of the internal Boolean relationship defined by the circuit design.

D. **INPUT LEAKAGE** — A current of either polarity which exceeds data sheet limits for input terminals. Large values of leakage are classified as shorts.

E. **THREE-STATE LEAKAGE** — A current of either polarity which exceeds data sheet limits for I/O terminals when under three-stated conditions. This parameter is also timing dependent and, when catastrophic, is classified as a functional failure mode.

F. **PARAMETRIC** — A broad classification of non-catastrophic failure modes which excludes leakages but includes:

1. Failure to respond at one or more output terminals with a correct logical state under worst-case supply, clock, and VIH/VIL conditions; usually the result of excessive propagation delays, improper VOH/VOL levels, or a dynamic logic state which should be static, etc. Must be 100% functional under nominal conditions and may be associated with leakage currents not previously detected.
2. Excessive power dissipation. For CMOS Microprocessors, leakage currents can be a significant contributing factor for this failure mode. Device is 100% functional.
3. Incorrect output analog voltage or current level not resulting in a functional failure.

APPENDIX E.
MICROPROCESSOR AVERAGE JUNCTION TEMPERATURES AND GATE COUNTS

MOS Technology	Device Type	Average Junction Temperature @T_A = 70°C			Equivalent Number of Gates
		Ceramic	Plastic A42	Plastic Cu	
NMOS	MC6800	83	92		1,367
	MC6802/08	91	116		3,633
	MC6810	83	92		1,083
	MC6821	79	92	81	450
	MC6844	85	103	88	1,000
	MC6845	89	105	90	750
	MC6846	89	109	91	3,755
	MC6847	83	94	84	833
	MC6850	81	92	85	580
	MC6852	83	91	84	907
	MC6854	89	101	91	1,400
	MC68488	85	98	86	893
	MC68652	86	106	88	6,442
	MC68653				3,200
	MC68661	85	102	91	4,200
	MC68701	99			11,267
HMOS	MC6801	95	96*	97	8,533
	MC6805P2	88	106	95	4,833
	MC6805R2/U2	82	108	87	6,430
	MC6809/E	92	117	96	3,000
	MC6829	92	117	96	3,293
	MC68000	97	95*		12,667
	MC68008	107			12,667
	MC68120	96			9,644
	MC68451				12,233
	MC68705P3	88			8,833
	MC68705R3	89			14,433
CMOS	MC141000	71	72		2,425
	MC141200	71	72		2,425
	MC146805E2	71	72		4,333
	MC146805F2	71	72		5,633
	MC146805G2	71	72		5,800
	MC146823	71	72		867

NOTES: * Plastic package with molded-in heatspreader.
A42 Plastic package with Alloy 42 leadframe.
Cu Plastic package with copper leadframe.

APPENDIX F.
RELIABILITY AND QUALITY MONITOR PROGRAM

The Motorola MOS Microprocessor Reliability and Quality Monitor Program is designed to generate an ongoing data base of reliability and quality performance for various categories of Microprocessor products. The primary purpose of the program is to identify negative trends in the data so that immediate corrective action can be taken. The program also allows Motorola to develop a large data base of reliability and quality results that can be reported quarterly to customers.

For the reliability monitor tests, each quarter sample group is pulled from major categories of product representing a matrix of processing and packaging technologies (see Sample Group chart). Product mix, sample availability and equipment capacity may cause the specific sample group pulled for a given quarter to vary from the chart shown. Each sample group has a specific set of reliability tests associated with it that are appropriate for that product type based on our history for that classification. At the end of each quarter, results are reported for all sample groups that have completed testing.

The quality results that are reported are the electrical and visual/mechanical AOQ (Average Outgoing Quality, given in parts per million defective) for the Microprocessor Division. This data represents the summary of results from the QC gate operation performed on every lot during the quarter. Electrical AOQ represents any AC, DC, or functional failure at any temperature (each lot is typically gated at two temperatures: hot and either room or cold). Visual/mechanical AOQ represents failures such as bent leads, incorrect marking, marking permanency problems, and cracked packages. The AOQ reported is the product of the process average (ratio of defective devices to largest sample size) and the lot acceptance rate.

Following are brief descriptions of the various reliability tests included in this program:

High Temperature Operating Life

High temperature operating life (H.T.O.L.) testing is performed to accelerate failure mechanisms which are thermally activated through the application of extreme temperatures and the use of dynamic operating conditions. The temperature and voltage conditions used in the stress are typically 125°C with a bias level at the maximum data sheet specification limit of 5.5 volts. All devices used in HTOL test are sampled directly after final electrical test with no prior burn-in or other pre-screening. Testing is performed per Mil Std 883B, Method 1005, with all stressing dynamic and minimum test duration 1008 hours. Some sample groups will be extended beyond 1008 hours, some run at temperatures higher than 125°C, and some at voltages higher than maximum rated voltage to look for the effects of these variations.

Device equivalent hours assume the Arrhenius relationship using an activation energy of 1.0 eV to extrapolate from the device junction temperature at 125°C to the junction temperature at 70°C. Failure rates given in FITs are derived using the Chi-Square distribution to a 90% confidence limit. A FIT is 1 failure per 10^9 device hours or 0.0001%/1000 Hours.

Temperature Humidity Bias

Temperature Humidity Bias (T.H.B.) is an environmental test performed at a temperature of 85°C and a relative humidity of 85%. The test is designed to measure the moisture resistance of plastic encapsulated circuits. A nominal voltage of 5 volts static bias is applied to the device to create the electrolytic cells necessary to accelerate corrosion of the metallization. Testing is performed per JEDEC Standard 22, Method A101. Most groups are tested to 100 hours with some groups extended beyond to look for longer term effects.

Autoclave

Autoclave, like T.H.B., is an environmental test which measures device resistance to moisture penetration along the leadframe-plastic interface. Conditions employed during the test include 121°C, 100% relative humidity, and 15 psig. Corrosion of the die is the expected failure mechanism. Autoclave is a highly accelerated and destructive test performed per JEDEC Standard 22, method A102. Testing is routinely performed for 144 hours.

Temperature Cycle

Temperature cycle testing accelerates the effects of thermal expansion mismatch among the different components within a specific packaging system. This test is typically performed per Mil Std 883B, Method 1010, Condition C (−65°C to +150°C), or JEDEC Standard 22, Method A104, Condition B (−40°C to +125°C). During temperature cycle testing, devices are inserted into a cycling system and held at the cold dwell temperature for at least ten minutes. Following this cold dwell, the devices are heated to the hot dwell where they remain for another ten minute minimum time period. The system employs a circulating air environment to assure rapid stabilization at the specified temperature. The dwell at each extreme, plus the two transition times of five minutes each (one up to the hot dwell temperature, another down to the cold dwell temperature), constitute one cycle. Test duration is for 1000 cycles with some tests extended to look for longer term effects.

Thermal Shock

The objective of thermal shock testing is the same as that for temperature cycle testing — to emphasize differences in expansion coefficients for components of the packaging systems. However, thermal shock provides additional stress in that the device is exposed to a sudden change in temperature due to the transfer time of ten seconds maximum as well as the increased thermal conductivity of a liquid ambient. This test is performed per Mil Std 883B, Method 1011, Condition C (−65°C to +150°C). Devices are placed in a fluorocarbon bath and cooled to −65°C. After being held in the cold chamber for five minutes minimum, the devices are transferred to an adjacent chamber filled with fluorocarbon at +150°C for an equivalent time. Two five-minute dwells plus two ten-second transitions constitute one cycle. Test duration is normally for 1000 cycles with some tests being extended to look for longer term effects.

Data Retention

Data retention testing or high temperature storage is performed to measure the stability of programmed EPROM and EEPROM devices during storage at elevated temperatures with no electrical stress applied. The devices are exposed to an ambient environment of 150°C per Mil Std 883B, Method 1008, Condition C. An acceleration of charge loss from the storage cell is the expected result. All groups are typically tested to 1008 hours.

RELIABILITY AND QUALITY MONITOR PROGRAM

SAMPLE GROUPS

Category Name	Typical Product Types	Minimum Number of Sample Groups/Qtr	Test Performed	
			No. Samples	(Typ.)
NMOS Plastic	6800 Family 3870, 6800, 6810 6821, 6845, Custom	8	HTOL THB Autoclave TC/TS	45 Pcs 34 Pcs 22 Pcs 38 Pcs
HMOS Plastic	6801 Family 6805 Family 6809 Family	4	HTOL THB Autoclave TC/TS	45 Pcs 34 Pcs 22 Pcs 38 Pcs
CMOS Plastic	CMOS Family 146805E2 146805G2	4	HTOL THB Autoclave TC/TS	45 Pcs 34 Pcs 22 Pcs 38 Pcs
68000 Family Plastic (HMOS)	68000	2	HTOL THB Autoclave TC/TS	45 Pcs 36 Pcs 38 Pcs 38 Pcs
CERDIP (NMOS or HMOS)	6800 Family 3870, 6800, 6810, 6821, 6845, 6801, 6805, 6809	2	HTOL TC/TS	45 Pcs 38 Pcs
Side Braze	6800 Family 3870, 6800, 6810 6821, 6845, 6810, 6805, 6809	2	TC/TS	52 Pcs
Leadless Chip Carrier	146805E2 146805G2 CMOS Family	3	HTOL TC/TS	30 Pcs 38 Pcs
68000 Family Ceramic (HMOS)	68000	2	HTOL	45 Pcs
EPROM MCU (NMOS, HMOS or CMOS)	68701 68705 1468705G2	2	HTOL TC/TS Data Retention	45 Pcs 38 Pcs 45 Pcs

APPENDIX G.
QUALITY PERFORMANCE

The chart below gives the goals and actuals for the Microprocessor Division Electrical and Visual/Mechanical AOQ (Average Outgoing Quality, given in parts per million defective). This data represents the summary of results from the QC gate operations performed on every lot. Electrical AOQ represents any AC, DC, or functional failure at any temperature (each lot is typically gated at two temperatures: hot, and either room or cold). Visual/Mechanical AOQ represents failures such as bent leads, incorrect marking, marking permanency problems, and cracked packages. The AOQ reported is the product of the process average (ratio of defective devices to largest sample size) and the lot acceptance rate.

AVERAGE OUTGOING QUALITY

	Goal	Electrical AOQ (PPM) Actual	Visual/Mechanical AOQ (PPM) Actual
Total 1979	3000	(~) 4000	(~) 4500
Total 1980	2500	(~) 2000	(~) 2500
Total 1981	1500	1725	1920
1st Qtr 1982	1200	1045	1408
2nd Qtr 1982	1000	868	1934
3rd Qtr 1982	800	492	1062
4th Qtr 1982	600	636	651
1st Qtr 1983	500	326	405
2nd Qtr 1983	450	341	267
3rd Qtr 1983	400	313	251
4th Qtr 1983	350		
1st Half 1984	275		
2nd Half 1984	275		
1st Half 1985	175		
2nd Half 1985	125		
1986	100		

Data Sheets

3

MOTOROLA

MC1372

COLOR TV VIDEO MODULATOR

. . . an integrated circuit used to generate an RF TV signal from baseband color-difference and luminance signals.

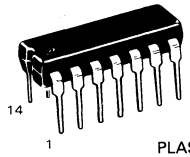
The MC1372 contains a chroma subcarrier oscillator, a lead and lag network, a quasi-quadrature suppressed carrier DSB chroma modulator, an RF oscillator and modulator, and an LSTTL compatible clock driver with adjustable duty cycle.

The MC1372 is a companion part to the MC6847 Video Display Generator, providing and accepting the correct dc interconnection levels. This device may also be used as a general-purpose modulator with a variety of video signal generating devices such as video games, test equipment, video tape recorders, etc.

- Single 5.0 Vdc Supply Operation for NMOS and TTL Compatibility
- Minimal External Components
- Compatible with MC6847 Video Display Generator
- Sound Carrier Addition Capability
- Modulates Channel 3 or 4 Carrier with Encoded Video Signal
- Low Power Dissipation
- Linear Chroma Modulators for High Versatility
- Composite Video Signal Generation Capability
- Ground-Referenced Video Prevents Overmodulation

COLOR TV VIDEO MODULATOR CIRCUIT

SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 646-05

Pin Connections

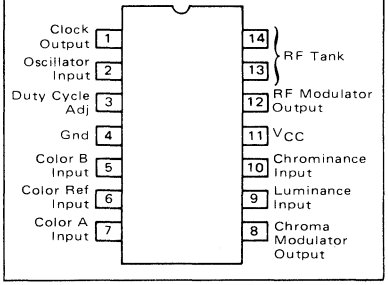
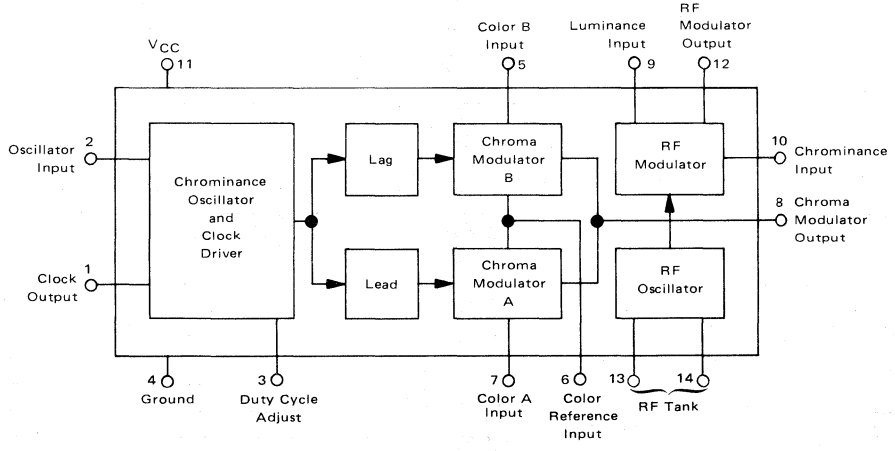


FIGURE 1 – BLOCK DIAGRAM



MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Value	Unit
Supply Voltage	8.0	Vdc
Operating Ambient Temperature Range	0 to +70	°C
Storage Temperature Range	−65 to +150	°C
Junction Temperature	150	°C
Power Dissipation, Package Derate above 25°C	1.25 13	Watts mW/°C

RECOMMENDED OPERATING CONDITIONS

Supply Voltage	5.0	Vdc
Luma Input Voltage — Sync Tip Peak White	1.0 0.35	Vdc
Color Reference Voltage	1.5	Vdc
Color A, B Input Voltage Range	1.0 to 2.0	Vdc

ELECTRICAL CHARACTERISTICS (V_{CC} = +5 Vdc, T_A = 25°C, Test Circuit 1 unless otherwise noted)

Characteristic	Min	Typ	Max	Unit
Operating Supply Voltage	4.75	5.0	5.25	Volts
Supply Current	–	25	–	mA
CHROMA OSCILLATOR/CLOCK DRIVER (Measured at Pin 1 unless otherwise noted)				
Output Voltage (V_{OL}) (V_{OH})	– 2.4	– –	0.4 –	Vdc
Rise Time (V1 = 0.4 to 2.4 Vdc)	–	–	50	ns
Fall Time (V1 = 2.4 to 0.4 Vdc)	–	–	50	ns
Duty Cycle Adjustment Range (V3 = 5.0 Vdc) (Measured at V1 = 1.4 V)	70	–	30	%
Inherent Duty Cycle (No connection to Pin 3)	–	50	–	%
CHROMA MODULATOR (V5 = V6 = V7 = 1.5 Vdc unless otherwise noted)				
Input Common Mode Voltage Range (Pins 5, 6, 7)	0.8	–	2.3	Vdc
Oscillator Feedthrough (Measured at Pin 8)	–	15	31	mV(p-p)
Modulation Angle [θ8(V7 = 2.0 Vdc) − θ8(V5 = 2.0 Vdc)]	85	100	115	degrees
Conversion Gain [V8/(V7 − V6); V8/(V5 − V6)]	–	0.6	–	V(p-p)/Vdc
Input Current (Pins 5, 6, 7)	–	–	−20	μA
Input Resistance (Pins 5, 6, 7)	100	–	–	kΩ
Input Capacitance (Pins 5, 6, 7)	–	–	5.0	pF
Chroma Modulator Linearity (V5 = 1.0 to 2.0 V; V7 = 1.0 to 2.0 V)	–	4.0	–	%
RF MODULATOR				
Luma Input Dynamic Range (Pin 9, Test Circuit 2)	0	–	1.5	Volts
RF Output Voltage (f = 67.25 MHz, V9 = 1.0 V)	–	15	–	mVrms
Luma Conversion Gain (ΔV12/ΔV9; V9 = 0.1 to 1.0 Vdc) Test Circuit 2	–	0.8	–	V/V
Chroma Conversion Gain (ΔV12/ΔV10; V10 = 1.5 Vp-p; V9 = 1.0 Vdc) Test Circuit 2	–	0.95	–	V/V
Chroma Linearity (Pin 12, V10 = 1.5 Vp-p) Test Circuit 2	–	1.0	–	%
Luma Linearity (Pin 12, V9 = 0 to 1.5 Vdc) Test Circuit 2	–	2.0	–	%
Input Current (Pin 9)	–	–	−20	μA
Input Resistance (Pin 10)	–	800	–	Ω
Input Resistance (Pin 9)	100	–	–	kΩ
Input Capacitance (Pins 9, 10)	–	–	5.0	pF
Residual 920 kHz (Measured at Pin 12) See Note 1	–	50	–	dB
Output Current (Pin 12, V9 = 0 V) Test Circuit 2	–	1.0	–	mA
TEMPERATURE CHARACTERISTICS (V_{CC} = 5 Vdc, T_A = 0 to 70°C, IC only)				
Chroma Oscillator Deviation (f_o = 3.579545 MHz)	–	±50	–	Hz
RF Oscillator Deviation (f_o = 67.25 MHz)	–	±250	–	kHz
Clock Drive Duty Cycle Stability	±5.0	–	–	%

NOTE 1. V9 = 1.0 Vdc, V_C = 300 mV(p-p) @ 3.58 MHz,
V_S = 250 mV(p-p) @ 4.5 MHz, Source Impedance = 75 Ω.

FIGURE 2 – TEST CIRCUIT 1

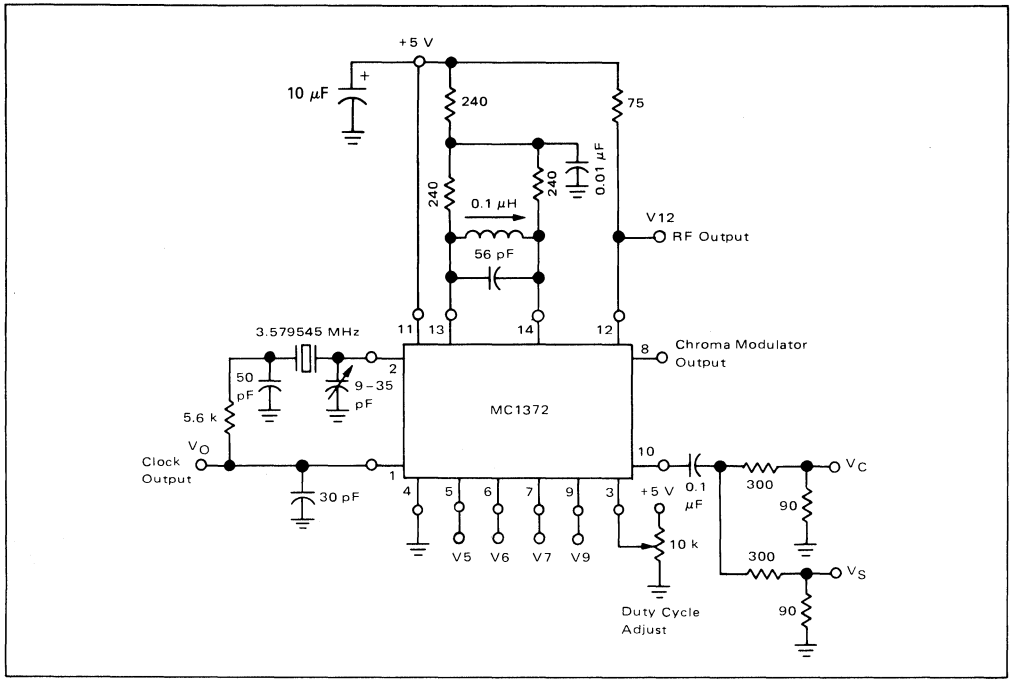


FIGURE 3 – TEST CIRCUIT 2

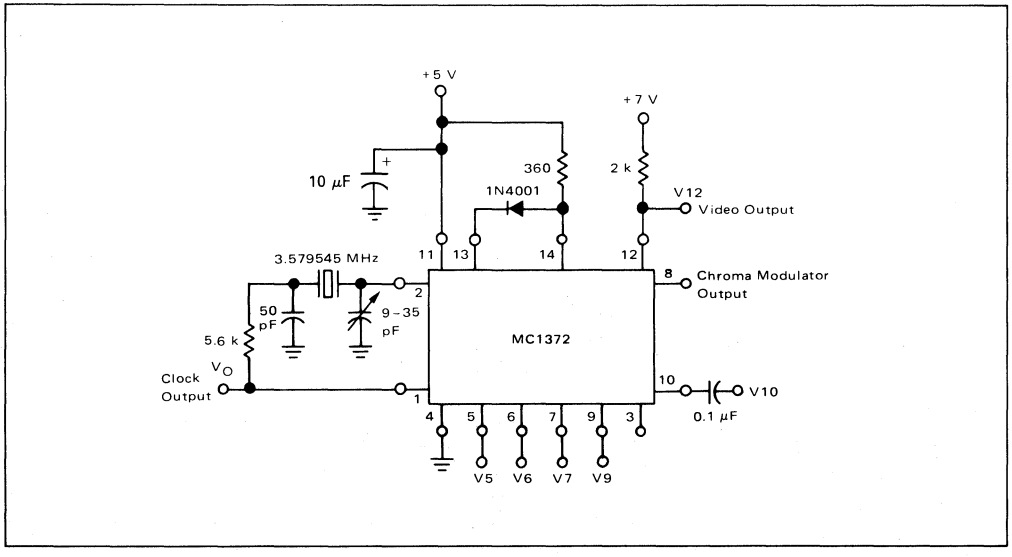
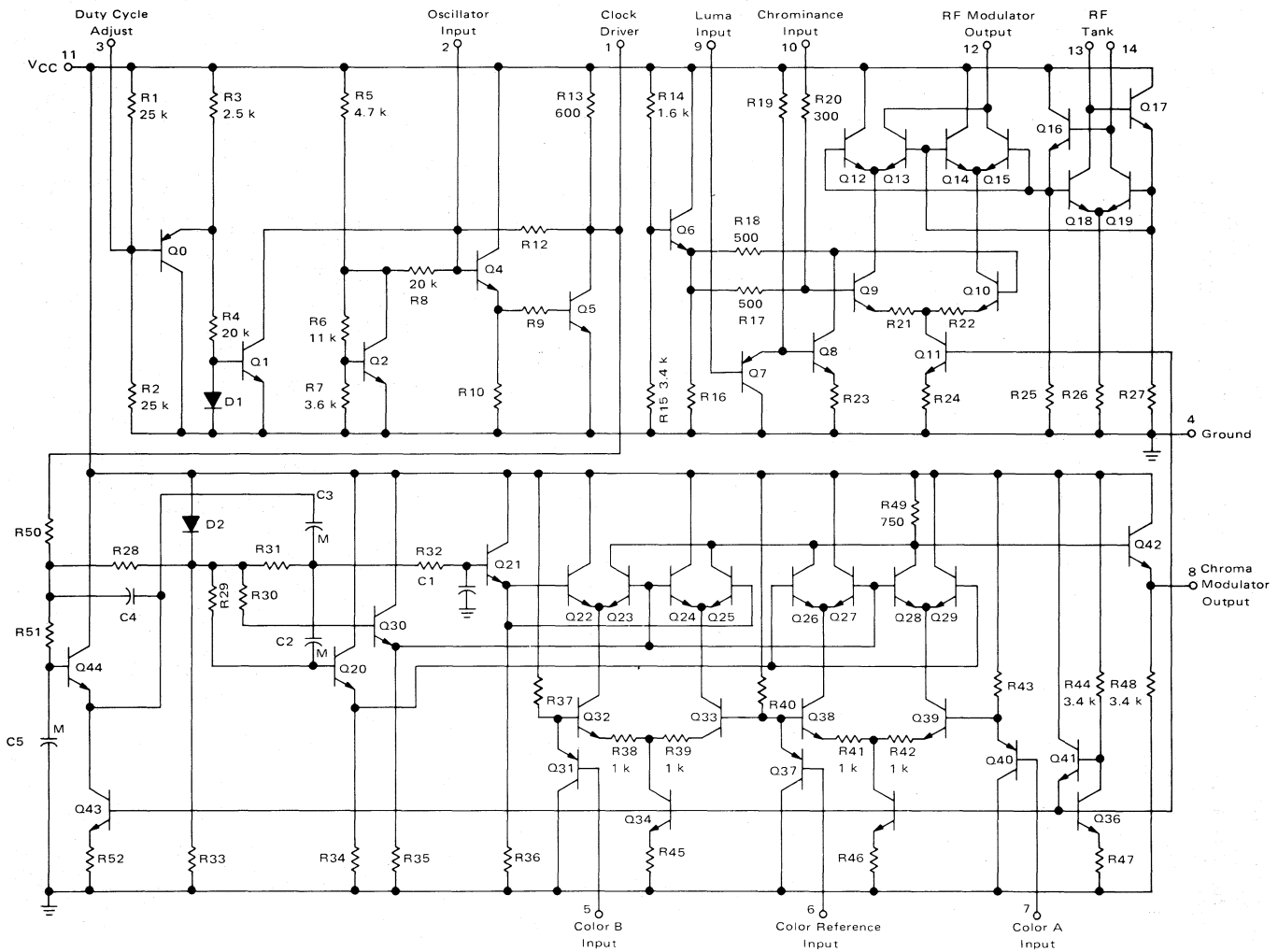


FIGURE 4 – SCHEMATIC DIAGRAM



OPERATIONAL DESCRIPTION

Pin 1 – Clock Output

Provides a rectangular pulse output waveform with frequency equal to the chrominance subcarrier oscillator. This output is capable of driving one LS-TTL load.

Pin 2 – Oscillator Input

Color subcarrier oscillator feedback input. Signal from the clock output is externally phase shifted and ac coupled to this pin.

Pin 3 – Duty Cycle Adjust

A dc voltage applied to this pin adjusts the duty cycle of the clock output signal. If the pin is left unconnected, the duty cycle is approximately 50%.

Pin 4 – Ground

Pin 5 – Color B Input

Dc coupled input to Chroma Modulator B, whose phase leads modulator A by approximately 100°. The modulator output amplitude and polarity correspond to the voltage difference between this pin and the Color Reference Voltage at Pin 6.

Pin 6 – Color Reference Input

The dc voltage applied to this pin establishes the reference voltage to which Color A and Color B inputs are compared.

Pin 7 – Color A Input

Dc coupled input to Chroma Modulator A, whose phase lags modulator B by approximately 100°. The modulator output amplitude and polarity correspond to the voltage difference between this pin and the Color Reference Voltage at Pin 6.

Pin 8 – Chroma Modulator Output

Low impedance (emitter follower) output which provides the vectorial sum of chroma modulators A and B.

Pin 9 – Luminance Input

Input to RF modulator. This pin accepts a dc coupled luminance and sync signal. The amplitude of the RF signal output increases with positive voltage applied to the pin, and ground potential results in zero output (i.e., 100% modulation). A signal with positive-going sync should be used.

Pin 10 – Chrominance Input

Input to the RF modulator. This pin accepts ac coupled chrominance provided by the Chroma Modulator Output (pin 8). The signal is reduced by an internal resistor divider before being applied to the RF modulator. The resistor divider consists of a 300 ohm series resistor and a 500 ohm shunt resistor. Additional gain reduction may be obtained by the addition of external series resistance to pin 10.

Pin 11 – V_{CC}

Positive supply voltage

Pin 12 – RF Modulator Output

Common collector of output modulator stage. Output impedance and stage gain may be selected by choice of resistor connected between this pin and dc supply.

Pins 13 and 14 – RF Tank

A tuned circuit connected between these pins determines the RF oscillator frequency. The tuned circuit must provide a low dc resistance shunt. Applying a dc offset voltage between these pins results in baseband composite video at the RF Modulator Output.

MC1372 CIRCUIT DESCRIPTION

The chrominance oscillator and clock driver consist of emitter follower Q4 and inverting amplifier Q5. Signal presented at clock driver output pin 1 is coupled to oscillator input pin 2 through an external RC and crystal network, which provides 180° phase shift at the resonant frequency. The duty cycle of the output waveform is determined by the dc component at pin 1 internally coupled through R12 to the base of Q4. As pin 1 dc voltage increases, a smaller portion of the sinusoidal feedback signal at pin 2 exceeds the Q4 base voltage of two times V_{BE} required for conduction. As the dc level is reduced, device Q4 and thus Q5 is turned on for a longer percentage of the cycle. Transistors Q0, Q1, Q2 and diode D1 provide the biasing network which determines the dc operating level of the oscillator. The transistor Q2 and resistors R5, R6, and R7 form a voltage reference of four times V_{BE} at the collector of Q2. The dc voltage at pin 1 is determined by the values of R4, R8, and R12 and the applied duty cycle adjust voltage at pin 3. Since these resistors are nominally equal, the voltage at pin 1 will always approximate the dc voltage at pin 3.

The oscillator signal at pin 1 is internally coupled to active filter Q44. This filter reduces the frequency content above 4 MHz. The output of the filter at the emitter of Q44 is ac coupled through C3 to the input of the lead/lag network. R32 and C1 provide approximately 50° of phase lag, while C2 and R29 provide approximately 50° of phase lead. These two quasi-quadrature waveforms are used to switch chroma modulators B and A, respectively. The transistors Q22 through Q25 and Q32–Q33 form a doubly balanced modulator. The input signal applied at pin 5 is compared to the color dc reference voltage applied at pin 6 in differential amplifier Q32–Q33. The source current provided by transistor Q34 is partitioned in transistors Q32 and Q33 according to the differential input signal. The bases of transistors Q23 and Q24 are connected to the dc reference voltage at the emitter of Q30. The bases of transistors Q22 and Q25 are connected

to the phase delayed oscillator signal at the emitter of buffer transistor Q21. The differential signal currents provided by Q32 and Q33 are switched in transistors Q22 through Q25 and the resultant signal voltage is developed across R49. This signal has the phase and frequency of the oscillator signal at the emitter of Q21. The amplitude is proportional to the differential input signal applied between pins 5 and 6. Transistors Q26 through Q29 and Q38-Q39 form chroma modulator B. This modulator develops a signal voltage which is proportional to the differential voltage applied between pins 7 and 6. The phase and frequency of the output is equal to the phase advanced chroma oscillator at the emitter of buffer transistor Q20. Both chroma modulators A and B share the same output resistor, R49, so the output signal presented at the emitter of Q42 (pin 8) is the algebraic sum of modulators A and B.

The RF oscillator consists of differential amplifier Q18 and Q19 cross-coupled through emitter followers Q16 and Q17. The oscillator will operate at the parallel resonant frequency of the network connected between pins 13 and 14. The oscillator output is used to switch the doubly balanced RF modulator, Q9 through Q15. Transistors Q7 and Q8 provide level shifting and a high input impedance to the luminance input pin 9. The bases of transistors Q9 and Q10 are both biased through resistors R17 and R18, respectively, to the same dc reference voltage at Q6 emitter. The base voltage at Q10 may only be offset in a negative direction by luminance signal current source Q8. This design insures that overmodulation due to the luminance signal will never occur. The chrominance signal developed at pin 8 is externally ac coupled to pin 10 where it is reduced by resistor dividers R20 and R17, and added to the luminance signal in Q9. The resultant differential composite video currents are switched at the appropriate RF frequency in Q12 through Q15. The output signal current is presented at pin 12.

Transistors Q36, Q41 and resistors R44, R47 provide a highly stable voltage reference for biasing current sources Q43, Q34, Q35, and Q11.

MC1372 APPLICATION INFORMATION

Chrominance Oscillator

The oscillator is used as a clock signal for driving associated external circuitry, in addition to providing a switching signal for the chroma modulators. The IC uses an external crystal in a Colpitts configuration, as shown in Figure 5. Resistor R1 provides current limiting to reduce the signal swing. Capacitor C2 is adjusted for the exact frequency desired (3.579545 MHz).

In some applications, the duty cycle of the clock signal at pin 1 must be modified to overcome gate delays in associated equipment. The duty cycle may be adjusted by varying the dc voltage applied to pin 3. This adjustment may be made with the use of a potentiometer (10 kΩ) between supply and ground. With no connection to pin 3, the duty cycle is approximately 50%.

Chroma Modulator

The chrominance oscillator is internally phase shifted and applied to chroma modulators A and B. No external lead/lag networks are necessary. The phase relationship between the modulators is approximately 100^{o}, which was chosen to provide the best rendition of colors using equal amplitude color-difference signals. The voltage applied to pin 5, 6, or 7 must always be within the Input Common Mode Voltage Range. Since the amplitude of chrominance output is proportional to the voltage difference between pins 5 and 6 or 7 and 6, it is desirable to select the Color Reference Voltage applied to pin 6 to be midway between $V5_{max}$ and $V5_{min}$ (which should be $V7_{max}$ and $V7_{min}$). The Chroma B Modulator will be defined as a (B-Y) modulator if a burst flag signal is applied to the Color B Input (pin 5) at the appropriate time. This voltage should be negative with respect to the Color Reference Voltage, and typically has an amplitude equal to $1/2[V6-V5_{min}]$. Since the phase of burst is always defined as -(B-Y), the Chroma A Modulator approximates an (R-Y) modulator; however, the phase is offset by 10^{o} from the nominal 90^{o}, to provide the 100^{o} phase shift as discussed previously.

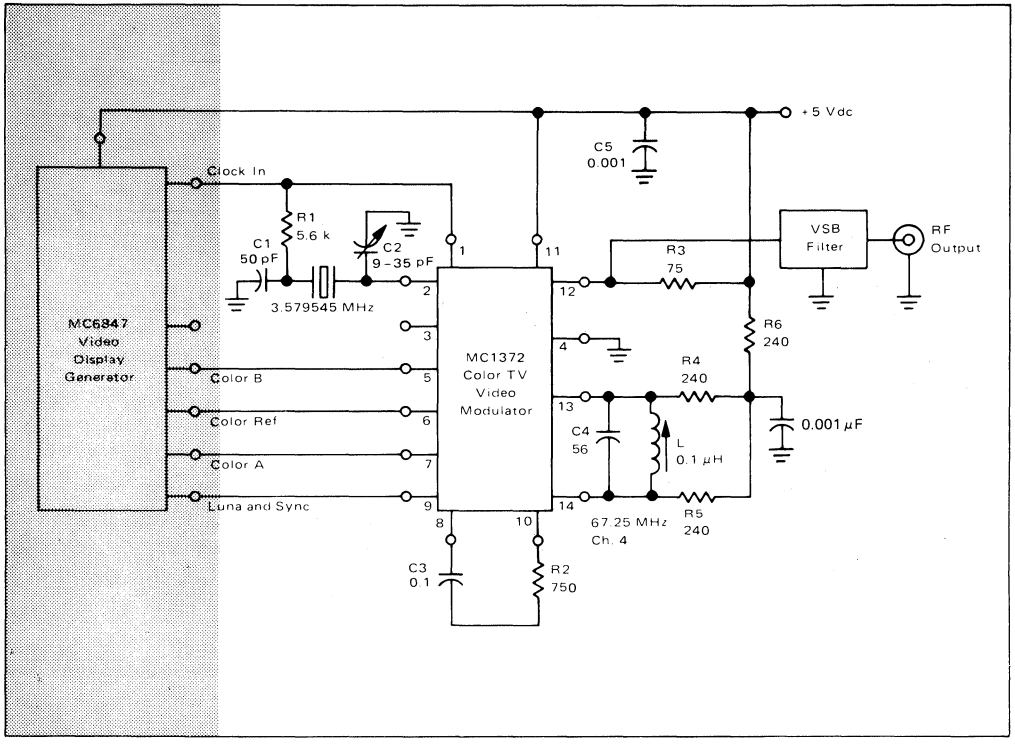
RF Modulator and Oscillator

The coil and capacitor connected between pins 13 and 14 should be selected to have a parallel resonance at the carrier frequency of the desired TV channel. The values of 56 pF and 0.1 μH shown in Figure 5 were chosen for a Channel 4 carrier frequency of 67.25 MHz. For Channel 3 operation, the resonant frequency should be 61.25 MHz (C = 75 pF, L = 0.1 μH). Resistors R4 and R5 are chosen to provide an adequate amplitude of switching voltage, whereas R6 is used to lower the maximum dc level of switching voltage below V_{CC}, thus preventing saturation within the IC.

Composite Luminance and Sync should be dc coupled to Luminance Input, pin 9. This signal must be within the Luma Input Dynamic Range to insure linearity. Since an increase in dc voltage applied to pin 9 results in an increase in RF output, the input signal should have positive-going sync to generate an NTSC compatible signal. As long as the input signal is positive, overmodulation is prevented by the integrated circuit.

Chrominance information should be ac coupled to Chrominance Input, pin 10. This pin is internally connected to a resistor divider consisting of a series 300 ohms and a shunt 500 ohms resistor. The input impedance is thus 800 ohms, and a coupling capacitor should be appropriately chosen.

FIGURE 5 – TYPICAL APPLICATION CIRCUIT



The Luminance to Chrominance ratio (L:C) may be modified with the addition of an external resistor in series with pin 10 (as shown in Figure 5). The unmodified L:C (A_O) is determined by the ratio of the respective Conversion Gain for equal amplitude signals (typically, 0.883 = −1.6 dB). The modified L:C will be governed by the equation $A_O(1 + R_{ext}/800)$ for equal amplitude input signals.

The internal chrominance modulators are not internally connected to the RF modulator; therefore, the user has the option of connecting an externally generated chrominance signal to the RF modulator. In addition, the RF modulator is wideband, and a 4.5 MHz FM audio signal may be added to the chrominance input at pin 10. This may be accomplished by selecting an appropriate series input resistor to provide the correct Luminance:Sound ratio.

The modulated RF signal is presented as a current at RF Modulator Output, pin 12. Since this pin represents a current source, any load impedance may be selected for matching purposes and gain selection, as long as the voltage at pin 12 is high enough to prevent the output devices from reaching saturation (approximately 4.5 V with components in Figure 5). The peak current out of pin 12 is typically 2 mA. Hence, a load resistance of up to 250 ohms may be safely used with a 5 V supply.

Composite Video Signal Generation

The RF modulator may be easily used as a composite video generator by replacing the RF oscillator tank circuit with a diode as shown in Figure 3. This results in the output modulator being biased so the summation of luminance and chrominance appears unswitched at pin 12. The polarity of the output waveform is controlled by the direction of the diode. *Inverted video:* Anode to pin 14, cathode to pin 13. *Non-inverted video:* Anode to pin 13, cathode to pin 14. Note that the supply resistor must always be connected to the anode of the diode.

The amplitude of signal may be increased by increasing the load resistor on pin 12 and returning it to a higher supply voltage. Any voltage up to the Absolute Maximum Rating may be used.

Applications with MC6847 Video Display Generator

The MC1372 may be easily interfaced to the MC6847 as shown in Figure 5. The dc levels generated and required by the VDG are compatible with the MC1372, so that pins 1, 5, 6, 7, and 9 may be directly coupled to the appropriate MC6847 pins. Both integrated circuits as well as any associated NMOS MPU may be driven from a common 5 Vdc supply.

Recommended Chroma-Luma Signals

A chroma modulation angle of 100° was chosen to facilitate a desirable selection of colors with a minimum number of input signal levels. The following table demonstrates applicable signal levels for a variety of colors.

RECOMMENDED CHROMA-LUMA SIGNALS

	Pin #9 Luminance Input (Vdc)	Pin #7 Color A (Vdc)	Pin #6 Color Ref. (Vdc)	Pin #5 Color B (Vdc)
Sync	1.0	1.5	1.5	1.5
Blanking	0.75	1.5	1.5	1.5
Burst	0.75	1.5	1.5	1.25
Black	0.70	1.5	1.5	1.5
Green	0.50	1.0	1.5	1.0
Yellow	0.38	1.5	1.5	1.0
Blue	0.62	1.5	1.5	2.0
Red	0.62	2.0	1.5	1.5
Cyan	0.50	1.0	1.5	1.5
Magenta	0.50	2.0	1.5	2.0
Orange	0.50	2.0	1.5	1.0
Buff	0.38	1.5	1.5	1.5

MOTOROLA

MC3440A
MC3441A
MC3443A

QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVERS

The MC3440A, MC3441A, MC3443A are quad bus transceivers intended for usage in instruments and programmable calculators equipped for interconnection into complete measurement systems. These transceivers allow the bidirectional flow of digital data and commands between the various instruments. Each of the transceiver versions provides four open-collector drivers and four receivers featuring input hysteresis.

The MC3440A version consists of three drivers controlled by a common Enable input and a single driver without an Enable input. Terminations are provided in the device.

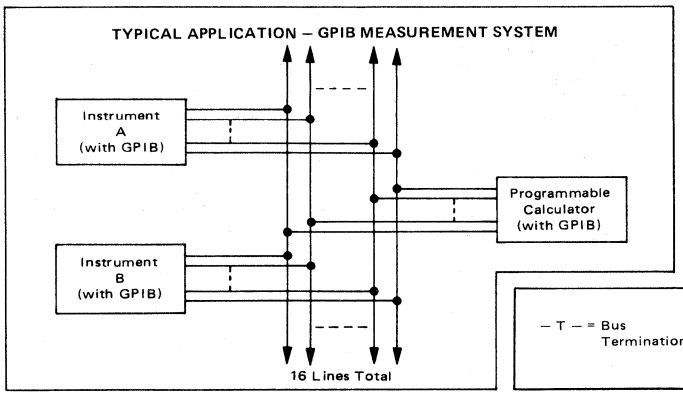
The MC3441A differs in that all four drivers are controlled by the common Enable input. Again, the terminations are provided.

The MC3443A is identical to the MC3441A except that the terminations have been omitted. As such it is pin compatible, and functionally equivalent to the SN75138. It does offer the advantage of receiver input hysteresis.

- Receiver Input Hysteresis Provides Excellent Noise Rejection
- Open-Collector Driver Outputs Permit Wire-OR Connection
- Tailored to Meet the Standards Set by the IEEE and IEC Committees on Instrument Interface (488-1978)
- Terminations provided (except MC3443A version)
- Provides Electrical Compatibility with General-Purpose Interface Bus

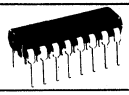
MAXIMUM RATINGS ($T_A = 25^{\circ}C$ unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Power Dissipation (Package Limitation) Derate above 25°C	P_D	830 6.7	mW mW/°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

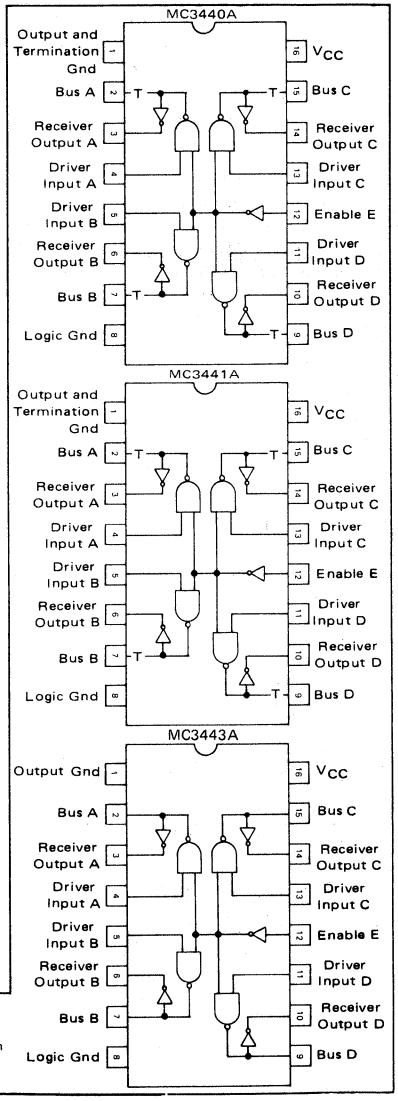


QUAD INTERFACE BUS TRANSCEIVERS

SILICON MONOLITHIC INTEGRATED CIRCUITS



P SUFFIX
PLASTIC PACKAGE
CASE 648-05



3

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, 4.5 V ≤ V_{CC} ≤ 5.5 V and 0 ≤ T_A ≤ 70°C, typical values are at T_A = 25°C, V_{CC} = 5.0 V)

Characteristic	Symbol	Min	Typ	Max	Unit
DRIVER PORTION					
Input Voltage – High Logic State	$V_{IH(D)}$	2.0	–	–	V
Input Voltage – Low Logic State	$V_{IL(D)}$	–	–	0.8	V
Input Current – High Logic State (V_{IH} = 2.4 V)	$I_{IH(D)}$	–	–	40	μA
Input Current – Low Logic State (V_{IL} = 0.4 V, V_{CC} = 5.0 V, T_A = 25°C) MC3443A / MC3440A, 3441A	$I_{IL(D)}$	–	–	–1.6 –0.25	mA
Input Clamp Voltage (I_{IK} = –12 mA)	$V_{IK(D)}$	–	–	–1.5	V
Output Voltage – High Logic State (1) (MC3440A, 3441A only) ($V_{IH(E)}$ = 2.4 V or $V_{IL(D)}$ = 0.8 V)	$V_{OH(D)}$	2.5	–	–	V
Output Voltage – Low Logic State ($V_{IH(D)}$ = 2.0 V, $V_{IL(E)}$ = 0.8 V, $I_{OL(D)}$ = 48 mA) ($V_{IH(D)}$ = 2.0 V, $V_{IL(E)}$ = 0.8 V, $I_{OL(D)}$ = 100 mA)	$V_{OL(D)}$	 – –	 – –	 0.5 0.80	V
Output Leakage Current – MC3443A Only ($V_{IH(E)}$ = 2.0 V or $V_{IL(D)}$ = 0.8 V)	$I_{OH(D)}$	–	–	250	μA
RECEIVER PORTION					
Input Hysteresis	–	400	580	–	mV
Input Threshold Voltage – Low to High Output Logic State (V_{CC} = 5.0 V, T_A = 25°C)	$V_{ILH(R)}$	0.8	0.98	–	V
Input Threshold Voltage – High to Low Output Logic State (V_{CC} = 5.0 V, T_A = 25°C)	$V_{IHL(R)}$	–	1.56	2.0	V
Output Voltage – High Logic State ($V_{IL(R)}$ = 0.8 V, $I_{OH(R)}$ = –400 μA)	$V_{OH(R)}$	2.4	–	–	V
Output Voltage – Low Logic State ($V_{IH(R)}$ = 2.0 V, $I_{OL(R)}$ = 16 mA)	$V_{OL(R)}$	–	–	0.5	V
Output Short-Circuit Current ($V_{IL(R)}$ = 0.8 V) (Only one output may be shorted at a time)	$I_{OS(R)}$	–20	–	–55	mA
BUS TERMINATION PORTION (Does not apply to MC3443A)					
Bus Voltage ($V_{IL(D)}$ = 0.8 V) (I_{BUS} = –12 mA) (No Load)	V_{BUS}	 – 2.50	 – –	 –1.5 3.70	V
Bus Current ($V_{IL(D)}$ = 0.8 V, V_{BUS} ≥ 5.0 V) ($V_{IL(D)}$ = 0.8 V, V_{BUS} ≤ 5.5 V) ($V_{IL(D)}$ = 0.8 V, V_{BUS} = 0.5 V) (V_{CC} = 0, 0 ≤ V_{BUS} ≤ 2.75 V) (MC3440A, 3441A only)	I_{BUS}	 0.7 – –1.3 –	 – – – –	 – 2.5 –3.2 +0.04	mA
TOTAL DEVICE POWER CONSUMPTION					
Power Supply Current ($V_{IH(D)}$ = 2.4 V, $V_{IL(E)}$ = 0 V)	I_{CC}	30	56	75	mA

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C)

Characteristic	Symbol	MC3440A,3441A Min	MC3440A,3441A Typ	MC3440A,3441A Max	MC3443A Min	MC3443A Typ	MC3443A Max	Unit
DRIVER PORTION								
Propagation Delay Time from Driver Input to Low Logic State Bus Output	$t_{PHL(D)}$	–	13	30	–	13	25	ns
Propagation Delay Time from Driver Input to High Logic State Bus Output	$t_{PLH(D)}$	–	17	30	–	17	25	ns
Propagation Delay Time from Enable Input to Low Logic State Bus Output	$t_{PHL(E)}$	–	25	40	–	25	32	ns
Propagation Delay Time from Enable Input to High Logic State Bus Output	$t_{PLH(E)}$	–	25	40	–	25	32	ns
RECEIVER PORTION								
Propagation Delay Time from Bus Input to High Logic State Receiver Output	$t_{PLH(R)}$	–	15	30	–	15	22	ns
Propagation Delay Time from Bus Input to Low Logic State Receiver Output	$t_{PHL(R)}$	–	15	30	–	15	22	ns

(1) 12 k resistor from the bus terminal to V_{CC} required on the MC3443A version.

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER INPUT (BUS) TO OUTPUT

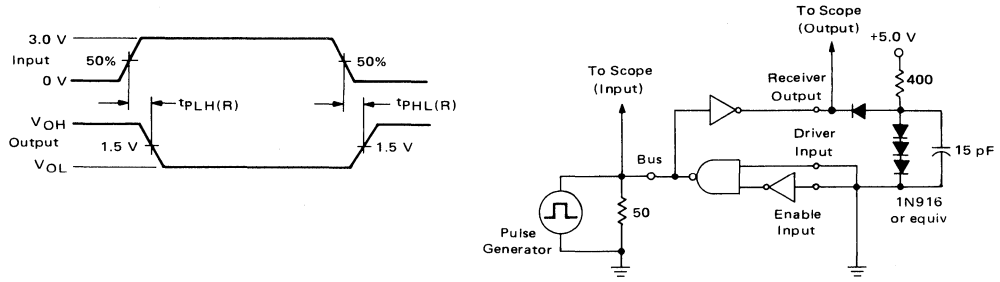


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)

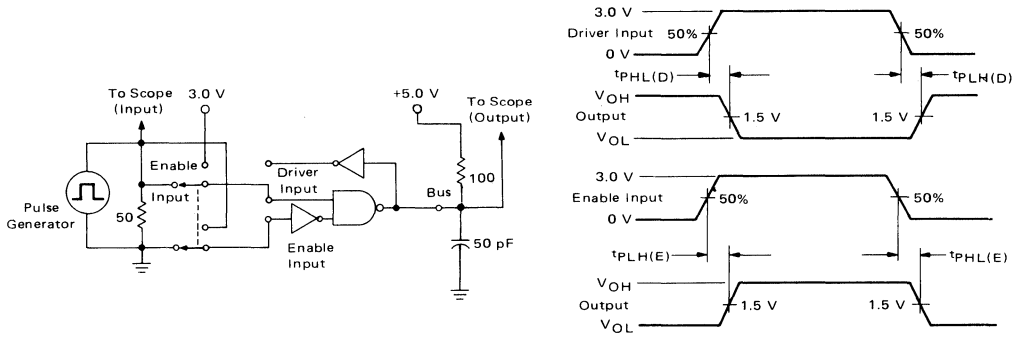
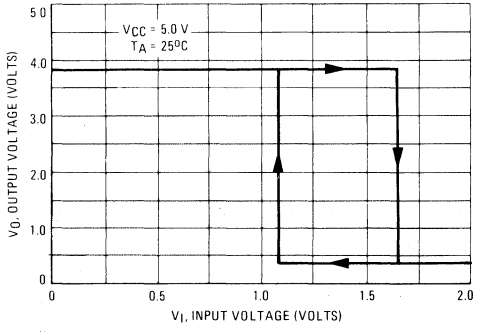
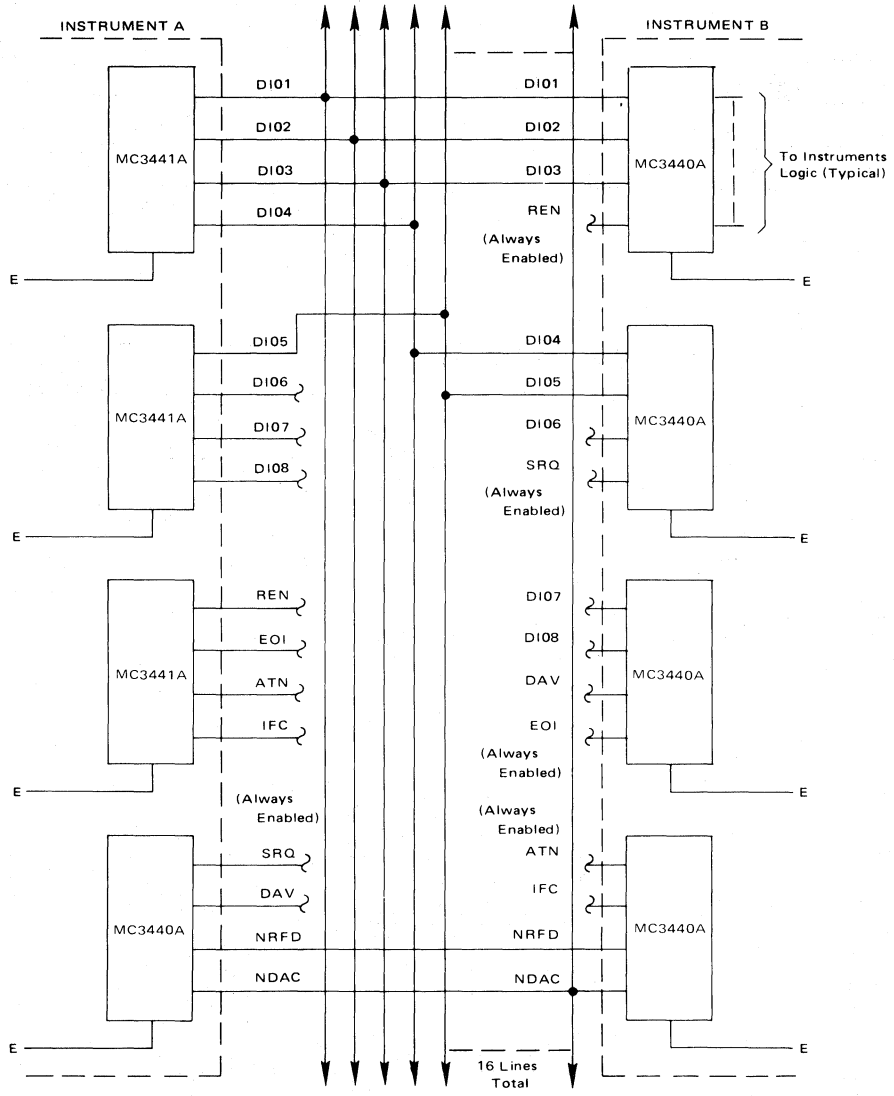


FIGURE 3 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS



GENERAL PURPOSE INTERFACE BUS APPLICATION



GPIB SIGNALS:

8 Line Data Bus: DI01 – DI08

5 General Interrupt Transfer Control Bus:
- REN – Remote Enable
- SRQ – Service Request
- EOI – End or Identify
- ATN – Attention
- IFC – Interface Clear

3 Data Byte Transfer Control Bus
- DAV – Data Valid
- NRFD – Not Ready for Data
- NDAC – Not Data Accepted

16 Total Signal Lines

MOTOROLA

MC3446A

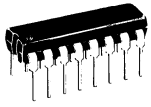
QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVER

The MC3446A is a quad bus transceiver intended for usage in instruments and programmable calculators equipped for interconnection into complete measurement systems. This transceiver allows the bidirectional flow of digital data and commands between the various instruments. The transceiver provides four open-collector drivers and four receivers featuring hysteresis.

- Tailored to Meet the IEEE Standard 488-1978 (Digital Interface for Programmable Instrumentation) and the Proposed IEC Standard on Instrument Interface
- Provides Electrical Compatibility with General-Purpose Interface Bus (GPIB)
- MOS Compatible with High Impedance Inputs
- Driver Output Guaranteed Off During Power Up/Power Down
- Low Power — Average Power Supply Current = 12 mA
- Terminations Provided

QUAD INTERFACE BUS TRANSCEIVER

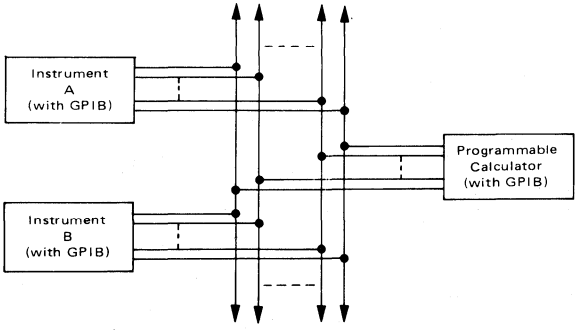
SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 648-05

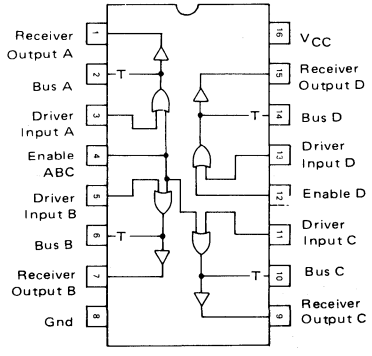
3

TYPICAL MEASUREMENT SYSTEM APPLICATION



16 Lines Total

PIN CONNECTIONS



— T — = Bus Termination

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS

(Unless otherwise noted, 4.5 V ≤ V_{CC} ≤ 5.5 V and 0 ≤ T_A ≤ 70°C, typical values are at T_A = 25°C, V_{CC} = 5.0 V)

Characteristic	Symbol	Min	Typ	Max	Unit
DRIVER PORTION					
Input Voltage – High Logic State	$V_{IH(D)}$	2.0	–	–	V
Input Voltage – Low Logic State	$V_{IL(D)}$	–	–	0.8	V
Input Current – High Logic State (V_{IH} = 2.4 V)	$I_{IH(D)}$	–	5.0	40	μA
Input Current – Low Logic State (V_{IL} = 0.4 V, V_{CC} = 5.0 V, T_A = 25°C)	$I_{IL(D)}$	–	-0.2	-0.25	mA
Input Clamp Voltage (I_{IK} = -12 mA)	$V_{IK(D)}$	–	–	-1.5	V
Output Voltage – High Logic State (1) ($V_{IH(S)}$ = 2.4 V or $V_{IH(D)}$ = 2.0 V)	$V_{OH(D)}$	2.5	3.3	3.7	V
Output Voltage – Low Logic State ($V_{IL(S)}$ = 0.8 V, $V_{IL(D)}$ = 0.8 V, $I_{OL(D)}$ = 48 mA)	$V_{OL(D)}$	–	–	0.5	
Input Breakdown Current ($V_{I(D)}$ = 5.5 V)	$I_{IB(D)}$	–	–	1.0	mA
RECEIVER PORTION					
Input Hysteresis	–	400	625	–	mV
Input Threshold Voltage – Low to High Output Logic State	$V_{ILH(R)}$	–	1.66	2.0	V
Input Threshold Voltage – High to Low Output Logic State	$V_{IHL(R)}$	0.8	1.03	–	V
Output Voltage – High Logic State ($V_{IH(R)}$ = 2.0 V, $I_{OH(R)}$ = -400 μA)	$V_{OH(R)}$	2.4	–	–	V
Output Voltage – Low Logic State ($V_{IL(R)}$ = 0.8 V, $I_{OL(R)}$ = 8.0 mA)	$V_{OL(R)}$	–	–	0.5	V
Output Short-Circuit Current ($V_{IH(R)}$ = 2.0 V) (Only one output may be shorted at a time)	$I_{OS(R)}$	4.0	–	14	mA
BUS LOAD CHARACTERISTICS					
Bus Voltage ($V_{IH(E)}$ = 2.4 V)	$V_{(BUS)}$	2.5	3.3	3.7	V
(I_{BUS} = -12 mA)		–	–	-1.5	
Bus Current ($V_{IH(O)}$ = 2.4 V, V_{BUS} ≥ 5.0 V)	$I_{(BUS)}$	0.7	–	–	mA
($V_{IH(D)}$ = 2.4 V, V_{BUS} = 0.5 V)		-1.3	–	-3.2	
(V_{BUS} ≤ 5.5 V)		–	–	2.5	
(V_{CC} = 0, 0 V ≤ V_{BUS} ≤ 2.75 V)		–	–	0.04	
TOTAL DEVICE POWER CONSUMPTION					
Power Supply Current	I_{CC}				mA
(All Drivers OFF)		–	12	19	
(All Drivers ON)		–	32	40	

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
DRIVER PORTION					
Propagation Delay Time from Driver Input to Low Logic State Bus Output	$t_{PHL(D)}$	–	–	50	ns
Propagation Delay Time from Driver Input to High Logic State Bus Output	$t_{PLH(D)}$	–	–	40	ns
Propagation Delay Time from Enable Input to Low Logic State Bus Output	$t_{PHL(E)}$	–	–	50	ns
Propagation Delay Time from Enable Input to High Logic State Bus Output	$t_{PLH(E)}$	–	–	50	ns
RECEIVER PORTION					
Propagation Delay Time from Bus Input to High Logic State Receiver Output	$t_{PLH(R)}$	–	–	50	ns
Propagation Delay Time from Bus Input to Low Logic State Receiver Output	$t_{PHL(R)}$	–	–	40	ns

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER INPUT (BUS) TO OUTPUT

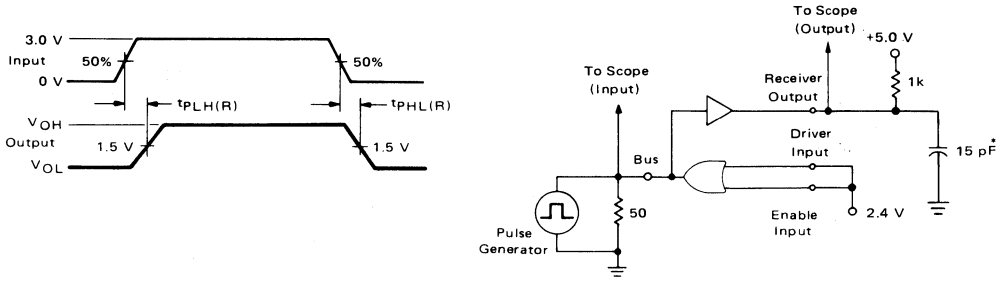
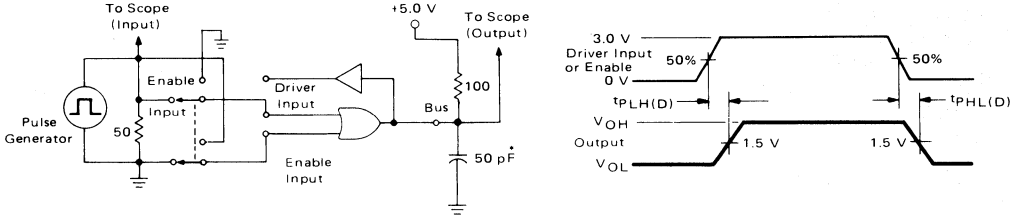


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)



* Includes Probe and Jig Capacitance

FIGURE 3 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

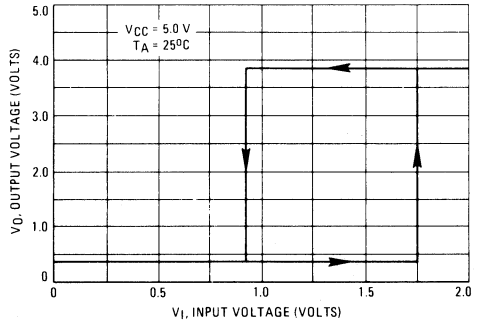
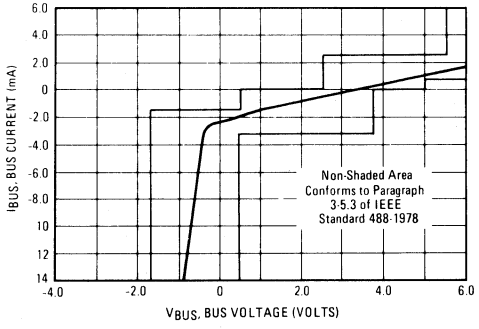


FIGURE 4 – TYPICAL BUS LOAD LINE



MOTOROLA

MC3447

BIDIRECTIONAL INSTRUMENTATION BUS (GPIB) TRANSCEIVER

This bidirectional bus transceiver is intended as the interface between TTL or MOS logic and the IEEE Standard Instrumentation Bus (488-1978, often referred to as GPIB). The required bus termination is internally provided.

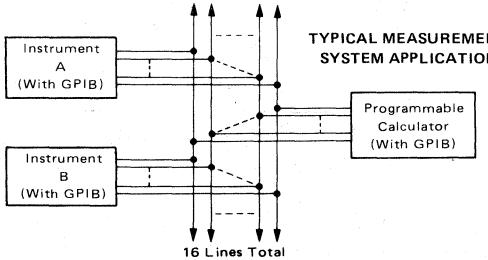
Low power consumption has been achieved by trading a minimum of speed for low current drain on non-critical channels. A fast channel is provided for critical ATN and EOI paths.

Each driver/receiver pair forms the complete interface between the bus and an instrument. Either the driver or the receiver of each channel is enabled by a Send/Receive input with the disabled output of the pair forced to a high impedance state. The receivers have input hysteresis to improve noise margin, and their input loading follows the bus standard specifications.

- Low Power — Average Power Supply Current = 30 mA Listening
  75 mA Talking
- Eight Driver/Receiver Pairs
- Three-State Outputs
- High Impedance Inputs
- Receiver Hysteresis — 600 mV (Typ)
- Fast Propagation Times — 15–20 ns (Typ)
- TTL Compatible Receiver Outputs
- Single +5 Volt Supply
- Open Collector Driver Output with Terminations
- Power Up/Power Down Protection (No Invalid Information Transmitted to Bus)
- No Bus Loading When Power is Removed From Device
- Required Termination Characteristics Provided

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

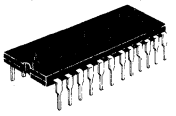
Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C



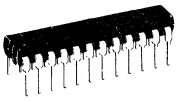
TYPICAL MEASUREMENT SYSTEM APPLICATION

OCTAL BIDIRECTIONAL BUS TRANSCEIVER WITH TERMINATION NETWORKS

SILICON MONOLITHIC INTEGRATED CIRCUIT

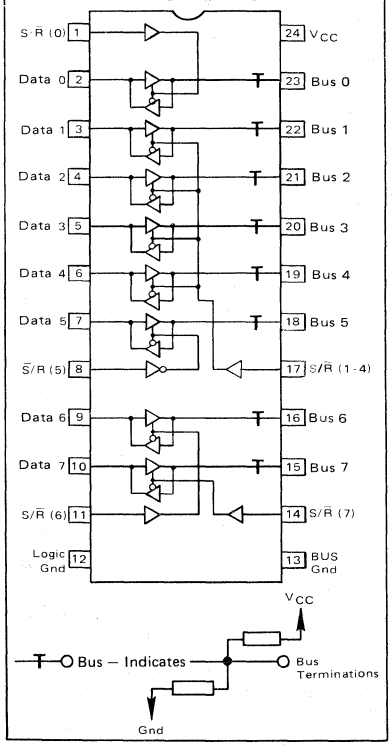


L SUFFIX
CERAMIC PACKAGE
CASE 623-05



P3 SUFFIX
PLASTIC PACKAGE
CASE 724-02

PIN ASSIGNMENTS



ELECTRICAL CHARACTERISTICS

(Unless otherwise noted 4.50 V ≤ V_{CC} ≤ 5.50 V and 0 ≤ T_A ≤ 70°C; typical values are at T_A = 25°C, V_{CC} = 5.0 V)

Characteristic — Note 2	Symbol	Min	Typ	Max	Unit
Bus Voltage					V
(Bus Pin Open) ($V_{I(S/\overline{R})}$ = 0.8 V)	$V_{(Bus)}$	2.5	—	3.7	
($I_{(Bus)}$ = −12 mA)	$V_{IC(Bus)}$	—	—	−1.5	
Bus Current	$I_{(Bus)}$				mA
(5.0 V ≤ $V_{(Bus)}$ ≤ 5.5 V)		0.7	—	2.5	
($V_{(Bus)}$ = 0.5 V)		−1.3	—	−3.2	
(V_{CC} = 0 V, 0 V ≤ $V_{(Bus)}$ ≤ 2.75 V)		—	—	+0.04	
Receiver Input Hysteresis ($V_{I(S/\overline{R})}$ = 0.8 V)	—	400	600	—	mV
Receiver Input Threshold					V
($V_{I(S/\overline{R})}$ = 0.8 V) Low to High	$V_{ILH(R)}$	—	1.6	2.0	
High to Low	$V_{IHL(R)}$	0.8	1.0	—	
Receiver Output Voltage — High Logic State ($V_{I(S/\overline{R})}$ = 0.8 V, $I_{OH(R)}$ = −200 μA, $V_{(Bus)}$ = 2.0 V)	$V_{OH(R)}$	2.4	—	—	V
Receiver Output Voltage — Low Logic State ($V_{I(S/\overline{R})}$ = 0.8 V, $I_{OL(R)}$ = 4.0 mA, $V_{(Bus)}$ = 0.8 V	$V_{OL(R)}$	—	—	0.5	V
Receiver Output Short Circuit Current ($V_{I(S/\overline{R})}$ = 0.8 V, $V_{(Bus)}$ = 2.0 V)	$I_{OS(R)}$	−4.0	—	−20	mA
Driver Input Voltage — High Logic State ($V_{I(S/\overline{R})}$ = 2.0 V)	$V_{IH(D)}$	2.0	—	—	V
Driver Input Voltage — Low Logic State ($V_{I(S/\overline{R})}$ = 2.0 V)	$V_{IL(D)}$	—	—	0.8	V
Driver Input Current — Data Pins ($V_{I(S/\overline{R})}$ = 2.0 V)					μA
(0.5 ≤ $V_{I(D)}$ ≤ 2.7 V)	$I_{I(D)}$	−100	—	40	
($V_{I(D)}$ = 5.5 V)	$I_{IB(D)}$	—	—	200	
Input Current — Send/Receive					μA
(0.5 ≤ $V_{I(S/R)}$ ≤ 2.7 V)	$I_{I(S/\overline{R})}$	−250	—	20	
($V_{I(S/\overline{R})}$ = 5.5 V)	$I_{IB(S/\overline{R})}$	—	—	100	
Driver Input Clamp Voltage ($V_{I(S/\overline{R})}$ = 2.0 V, $I_{IC(D)}$ = −18 mA)	$V_{IC(D)}$	—	—	−1.5	V
Driver Output Voltage — High Logic State ($V_{IS/\overline{R})}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V)	$V_{OH(D)}$	2.5	—	—	V
Driver Output Voltage — Low Logic State (Note 1) ($V_{I(S/\overline{R})}$ = 2.0 V, $V_{IL(D)}$ = 0.8 V, $I_{OL(D)}$ = 48 mA)	$V_{OL(D)}$	—	—	0.5	V
Power Supply Current					mA
(Listening Mode — All Receivers On)	I_{CCL}	—	30	45	
(Talking Mode — All Drivers On)	I_{CCH}	—	75	95	

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted)

Propagation Delay of Driver					ns
(Output Low to High)	$t_{PLH(D)}$	—	7.0	15	
(Output High to Low)	$t_{PHL(D)}$	—	16	30	
Propagation Delay of Receiver (Channels 0 to 5, 7)					ns
(Output Low to High)	$t_{PLH(R)}$	—	28	50	
(Output High to Low)	$t_{PHL(R)}$	—	15	30	
Propagation Delay of Receiver (Channel 6, Note 3)					ns
(Output Low to High)	$t_{PLH(R)}$	—	17	30	
(Output High to Low)	$t_{PHL(R)}$	—	12	22	

NOTES: 1. The IEEE 488-1978 Bus Standard changes $V_{OL(D)}$ from 0.4 to 0.5 V maximum to permit the use of Schottky technology.

2. Specified test conditions for $V_{I(S/\overline{R})}$ are 0.8 V (Low) and 2.0 V (High). Where $V_{I(S/\overline{R})}$ is specified as a test condition, $V_{I(\overline{S}/R)}$ uses the opposite logic levels.

3. In order to meet the IEEE 488-1978 standard for total system delay on the ATN and EOI channels, a fast receiver has been provided on Channel 6 (pins 9 and 16).

SWITCHING CHARACTERISTICS (continued) (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Propagation Delay Time — Send/Receiver to Data					ns
Logic High to Third State	$t_{PHZ(R)}$	—	15	30	
Third State to Logic High	$t_{PZH(R)}$	—	15	30	
Logic Low to Third State	$t_{PLZ(R)}$	—	15	25	
Third State to Logic Low	$t_{PZL(R)}$	—	10	25	
Propagation Delay Time — Send/Receiver to Bus					ns
Logic Low to Third State	$t_{PLZ(D)}$	—	13	25	
Third State to Logic Low	$t_{PZL(D)}$	—	30	50	

PROPAGATION DELAY TEST CIRCUITS AND WAVEFORMS

FIGURE 1 — BUS INPUT TO DATA OUTPUT (RECEIVER)

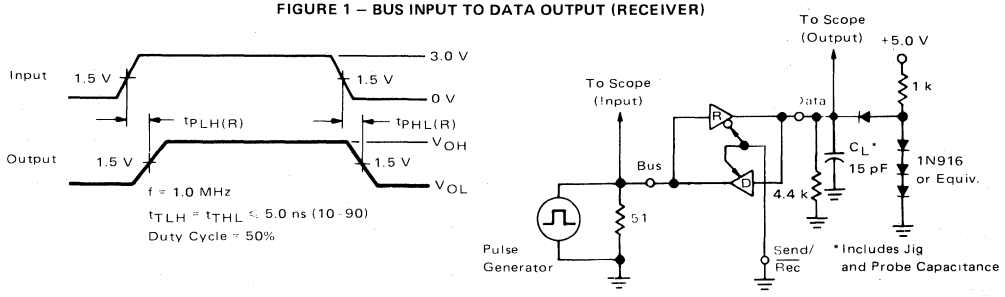


FIGURE 2 — DATA INPUT TO BUS OUTPUT (DRIVER)

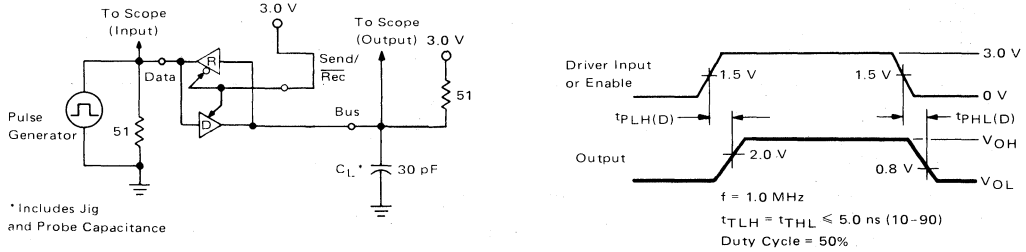


FIGURE 3 — SEND/$\overline{\text{RECEIVE}}$ INPUT TO BUS OUTPUT (DRIVER)

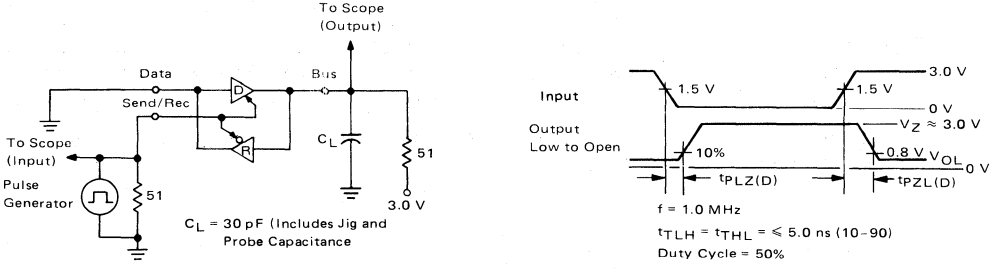


FIGURE 4 – SEND/$\overline{\text{RECEIVE}}$ INPUT TO DATA OUTPUT (RECEIVER)

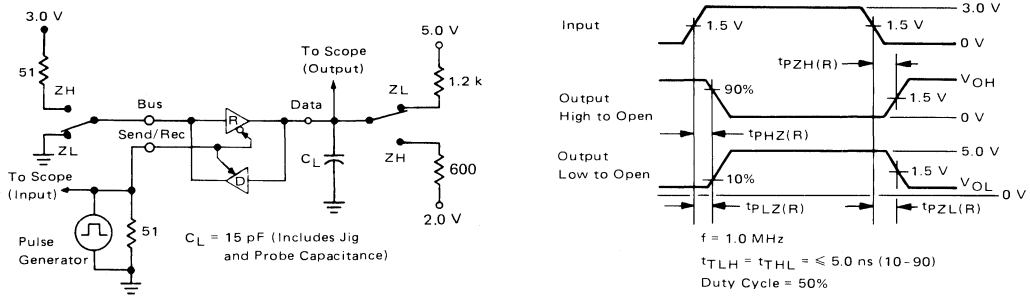


FIGURE 5 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

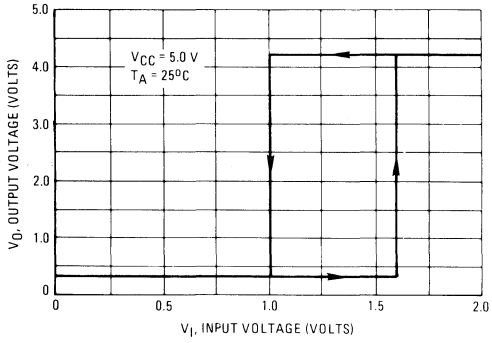


FIGURE 6 – TYPICAL BUS LOAD LINE

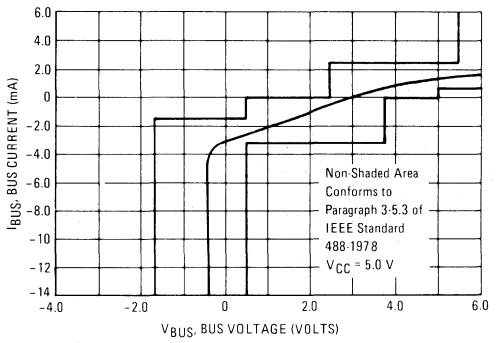


FIGURE 7 – SUGGESTED PRINTED CIRCUIT BOARD LAYOUT USING MC3447s AND MC68488

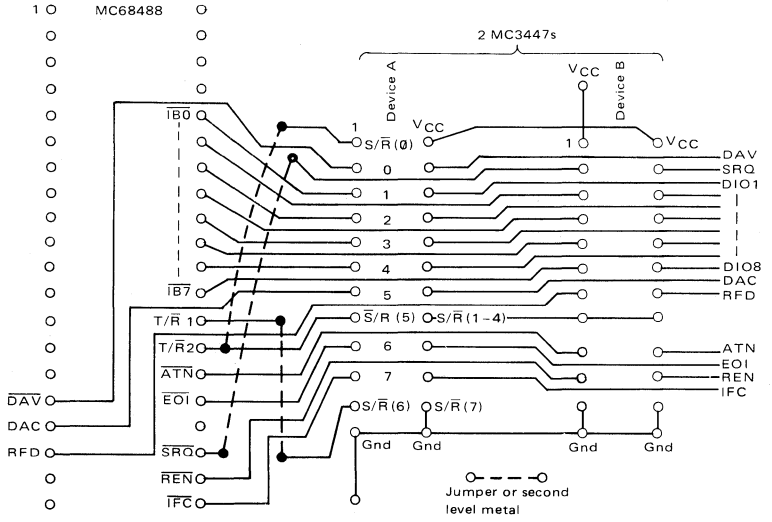
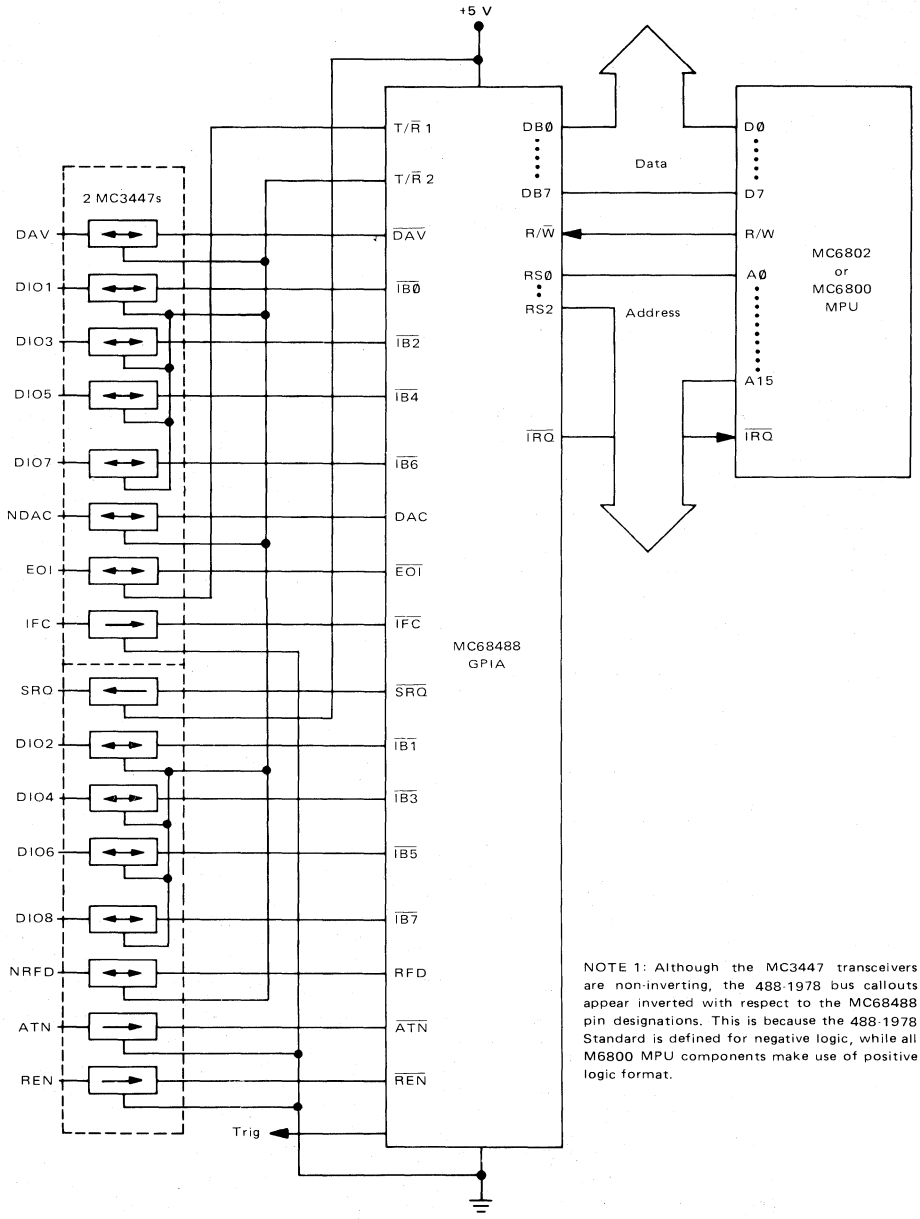


FIGURE 8 – SIMPLE SYSTEM CONFIGURATION

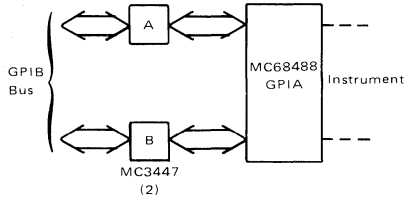


NOTE 1: Although the MC3447 transceivers are non-inverting, the 488-1978 bus callouts appear inverted with respect to the MC68488 pin designations. This is because the 488-1978 Standard is defined for negative logic, while all M6800 MPU components make use of positive logic format.

FIGURE 9 – SUGGESTED PIN DESIGNATIONS FOR USE WITH MC68488

MC68488 Connections		MC3447 Pin Designations				MC68488 Connections	
A	B					A	B
T/$\overline{R}$ 2	V_{CC}	S/$\overline{R}$ (0)	1	24	V_{CC}	V_{CC}	V_{CC}
$\overline{DAV}$	$\overline{SRQ}$	Data 0 0	2	23	Bus 0	DAV	SRQ
$\overline{IB0}$	$\overline{IB1}$	Data 1	3	22	Bus 1	DIO 1	DIO 2
$\overline{IB2}$	$\overline{IB3}$	Data 2	4	21	Bus 2	DIO 3	DIO 4
$\overline{IB4}$	$\overline{IB5}$	Data 3	5	20	Bus 3	DIO 5	DIO 6
$\overline{IB6}$	$\overline{IB7}$	Data 4	6	19	Bus 4	DIO 7	DIO 8
DAC	RFD	Data 5	7	18	Bus 5	NDAC	NRFD
T/$\overline{R}$ 2	T/$\overline{R}$ 2	S/$\overline{R}$ (5)	8	17	S/$\overline{R}$ (1–4)	T/$\overline{R}$ 2	T/$\overline{R}$ 2
$\overline{EOI}$	$\overline{ATN}$	Data 6	9	16	Bus 6	EOI	ATN
$\overline{IFC}$	$\overline{REN}$	Data 7	10	15	Bus 7	IFC	REN
T/$\overline{R}$ 1	Gnd	S/$\overline{R}$ (6)	11	14	S/$\overline{R}$ (7)	Gnd	Gnd
Gnd	Gnd	Logic Gnd	12	13	Bus Gnd	Gnd	Gnd

Octal GPIB Transceiver



MOTOROLA

MC3448A

BIDIRECTIONAL INSTRUMENTATION BUS (GPIB) TRANSCEIVER

This bidirectional bus transceiver is intended as the interface between TTL or MOS logic and the IEEE Standard Instrumentation Bus (488-1978, often referred to as GPIB). The required bus termination is internally provided.

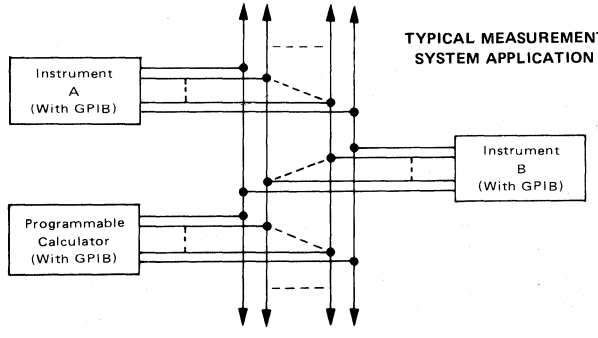
Each driver/receiver pair forms the complete interface between the bus and an instrument. Either the driver or the receiver of each channel is enabled by its corresponding Send/Receive input with the disabled output of the pair forced to a high impedance state. An additional option allows the driver outputs to be operated in an open collector(1) or active pull-up configuration. The receivers have input hysteresis to improve noise margin, and their input loading follows the bus standard specifications.

- Four Independent Driver/Receiver Pairs
- Three-State Outputs
- High Impedance Inputs
- Receiver Hysteresis – 600 mV (Typ)
- Fast Propagation Times – 15–20 ns (Typ)
- TTL Compatible Receiver Outputs
- Single +5 Volt Supply
- Open Collector Driver Output Option(1)
- Power Up/Power Down Protection
  (No Invalid Information Transmitted to Bus)
- No Bus Loading When Power Is Removed From Device
- Required Termination Characteristics Provided

(1) Selection of the "Open Collector" configuration, in fact, selects an open collector device with a passive pull-up load/termination which conforms to Figure 7, IEEE 488-1978 Bus Standard.

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

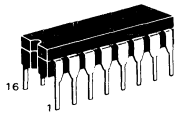
Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C



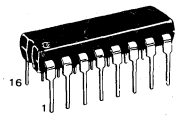
TYPICAL MEASUREMENT SYSTEM APPLICATION

QUAD THREE-STATE BUS TRANSCEIVER WITH TERMINATION NETWORKS

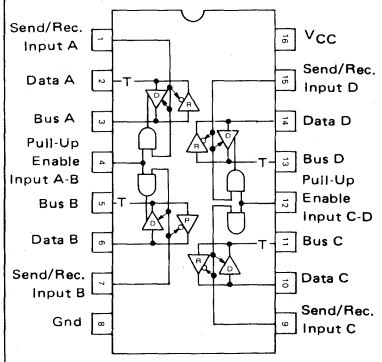
SILICON MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620-02



P SUFFIX
PLASTIC PACKAGE
CASE 648-05



TRUTH TABLE

Send/Rec.	Enable	Info. Flow	Comments
0	X	Bus → Data	—
1	1	Data → Bus	Active Pull-Up
1	0	Data → Bus	Open Col.

X = Don't Care

ELECTRICAL CHARACTERISTICS

(Unless otherwise noted 4.75 V ≤ V_{CC} ≤ 5.25 V and 0 ≤ T_A ≤ 70°C; typical values are at T_A = 25°C, V_{CC} = 5.0 V)

Characteristic	Symbol	Min	Typ	Max	Unit
Bus Voltage					V
(Bus Pin Open) ($V_{I(S/R)}$ = 0.8 V)	$V_{(BUS)}$	2.75	–	3.7	
($I_{(BUS)}$ = -12 mA)	$V_{IC(BUS)}$	–	–	-1.5	
Bus Current	$I_{(BUS)}$				mA
(5.0 V ≤ $V_{(BUS)}$ ≤ 5.5 V)		0.7	–	2.5	
($V_{(BUS)}$ = 0.5 V)		-1.3	–	-3.2	
(V_{CC} = 0 V, 0 V ≤ $V_{(BUS)}$ ≤ 2.75 V)		–	–	+0.04	
Receiver Input Hysteresis ($V_{I(S/R)}$ = 0.8 V)	–	400	600	–	mV
Receiver Input Threshold					V
($V_{I(S/R)}$ = 0.8 V, Low to High)	$V_{ILH(R)}$	–	1.6	1.8	
($V_{I(S/R)}$ = 0.8 V, High to Low)	$V_{IHL(R)}$	0.8	1.0	–	
Receiver Output Voltage — High Logic State ($V_{I(S/R)}$ = 0.8 V, $I_{OH(R)}$ = -800 μA, $V_{(BUS)}$ = 2.0 V)	$V_{OH(R)}$	2.7	–	–	V
Receiver Output Voltage — Low Logic State ($V_{I(S/R)}$ = 0.8 V, $I_{OL(R)}$ = 16 mA, $V_{(BUS)}$ = 0.8 V)	$V_{OL(R)}$	–	–	0.5	V
Receiver Output Short Circuit Current ($V_{I(S/R)}$ = 0.8 V, $V_{(BUS)}$ = 2.0 V)	$I_{OS(R)}$	-15	–	-75	mA
Driver Input Voltage — High Logic State ($V_{I(S/R)}$ = 2.0 V)	$V_{IH(D)}$	2.0	–	–	V
Driver Input Voltage — Low Logic State ($V_{I(S/R)}$ = 2.0 V)	$V_{IL(D)}$	–	–	0.8	V
Driver Input Current — Data Pins					μA
($V_{I(S/R)}$ = $V_{I(E)}$ = 2.0 V)					
(0.5 ≤ $V_{I(D)}$ ≤ 2.7 V)	$I_{I(D)}$	-200	–	40	
($V_{I(D)}$ = 5.5 V)	$I_{IB(D)}$	–	–	200	
Input Current — Send/Receive					μA
(0.5 ≤ $V_{I(S/R)}$ ≤ 2.7 V)	$I_{I(S/R)}$	-100	–	20	
($V_{I(S/R)}$ = 5.5 V)	$I_{IB(S/R)}$	–	–	100	
Input Current — Enable					μA
(0.5 ≤ $V_{I(E)}$ ≤ 2.7 V)	$I_{I(E)}$	-200	–	20	
($V_{I(E)}$ = 5.5 V)	$I_{IB(E)}$	–	–	100	
Driver Input Clamp Voltage ($V_{I(S/R)}$ = 2.0 V, $I_{IC(D)}$ = -18 mA)	$V_{IC(D)}$	–	–	-1.5	V
Driver Output Voltage — High Logic State ($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V, I_{OH} = -5.2 mA)	$V_{OH(D)}$	2.5	–	–	V
Driver Output Voltage — Low Logic State (Note 1) ($V_{I(S/R)}$ = 2.0 V, $I_{OL(D)}$ = 48 mA)	$V_{OL(D)}$	–	–	0.5	V
Output Short Circuit Current ($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V)	$I_{OS(D)}$	-30	–	-120	mA
Power Supply Current					mA
(Listening Mode — All Receivers On)	I_{CCL}	–	63	85	
(Talking Mode — All Drivers On)	I_{CCH}	–	106	125	

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted)

Propagation Delay of Driver					ns
(Output Low to High)	$t_{PLH(D)}$	–	–	15	
(Output High to Low)	$t_{PHL(D)}$	–	–	17	
Propagation Delay of Receiver					ns
(Output Low to High)	$t_{PLH(R)}$	–	–	25	
(Output High to Low)	$t_{PHL(R)}$	–	–	23	

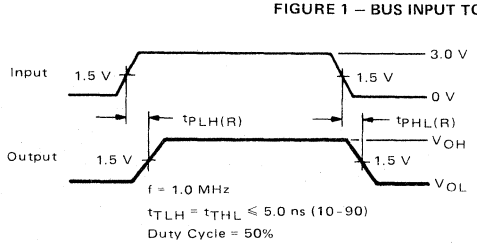
NOTE 1. A modification of the IEEE 488-1978 Bus Standard changes $V_{OL(D)}$ from 0.4 to 0.5 V maximum to permit the use of Schottky technology.

SWITCHING CHARACTERISTICS (continued) (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Propagation Delay Time — Send/Receive to Data					ns
Logic High to Third State	$t_{PHZ(R)}$	—	—	30	
Third State to Logic High	$t_{PZH(R)}$	—	—	30	
Logic Low to Third State	$t_{PLZ(R)}$	—	—	30	
Third State to Logic Low	$t_{PZL(R)}$	—	—	30	
Propagation Delay Time — Send/Receive to Bus					ns
Logic High to Third State	$t_{PHZ(D)}$	—	—	30	
Third State to Logic High	$t_{PZH(D)}$	—	—	30	
Logic Low to Third State	$t_{PLZ(D)}$	—	—	30	
Third State to Logic Low	$t_{PZL(D)}$	—	—	30	
Turn-On Time — Enable to Bus					ns
Pull-Up Enable to Open Collector	$t_{POFF(E)}$	—	—	30	
Open Collector to Pull-Up Enable	$t_{PON(E)}$	—	—	20	

PROPAGATION DELAY TEST CIRCUITS AND WAVEFORMS

FIGURE 1 — BUS INPUT TO DATA OUTPUT (RECEIVER)



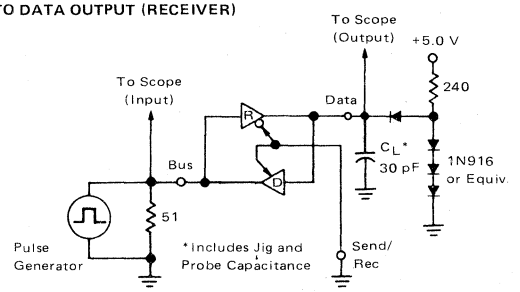
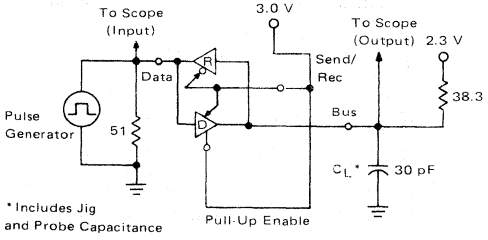


FIGURE 2 — DATA INPUT TO BUS OUTPUT (DRIVER)



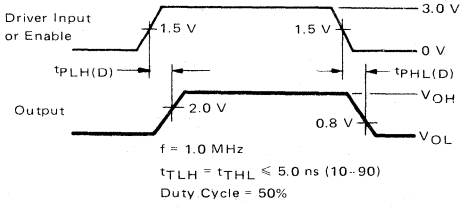
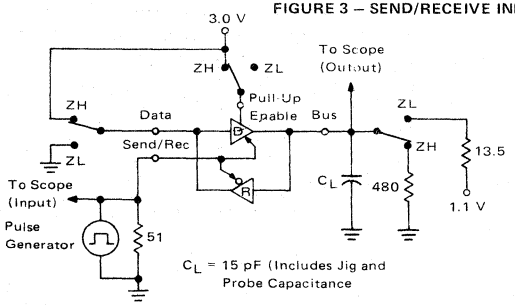


FIGURE 3 — SEND/RECEIVE INPUT TO BUS OUTPUT (DRIVER)



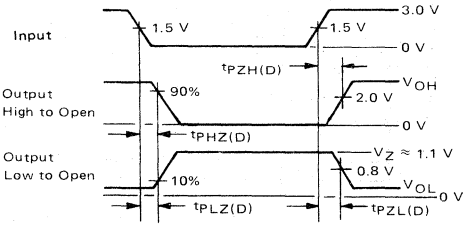
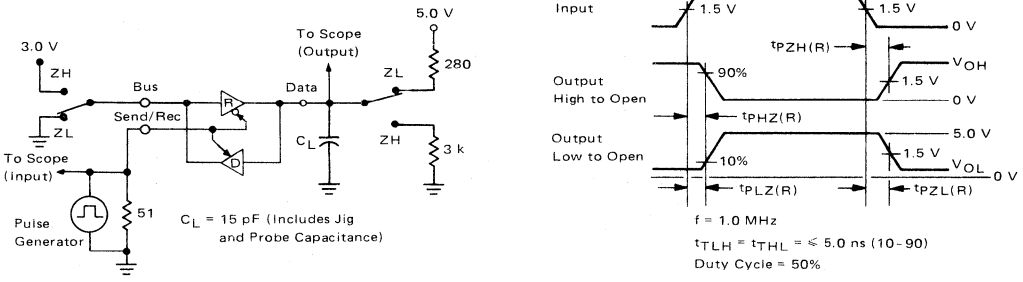


FIGURE 4 – SEND/RECEIVE INPUT TO DATA OUTPUT (RECEIVER)



3

FIGURE 5 – ENABLE INPUT TO BUS OUTPUT (DRIVER)

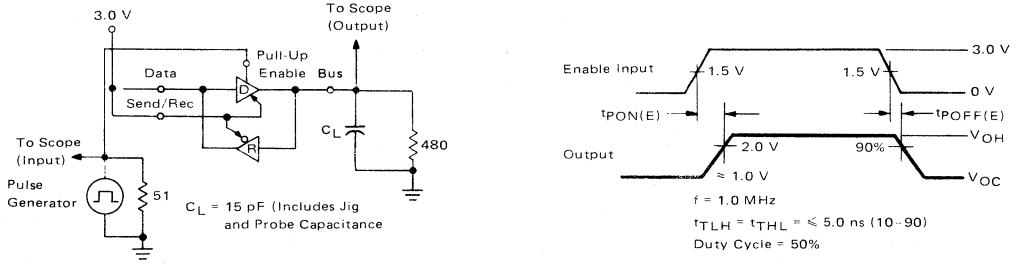


FIGURE 6 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

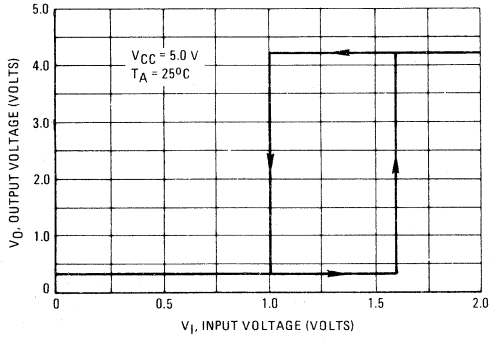


FIGURE 7 – TYPICAL BUS LOAD LINE

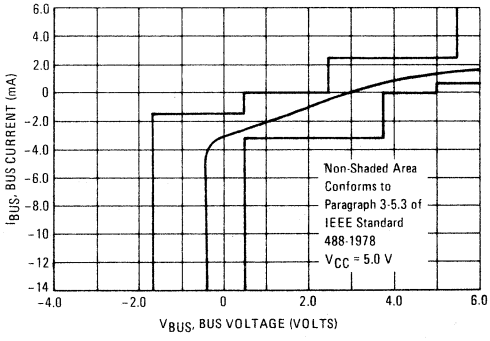
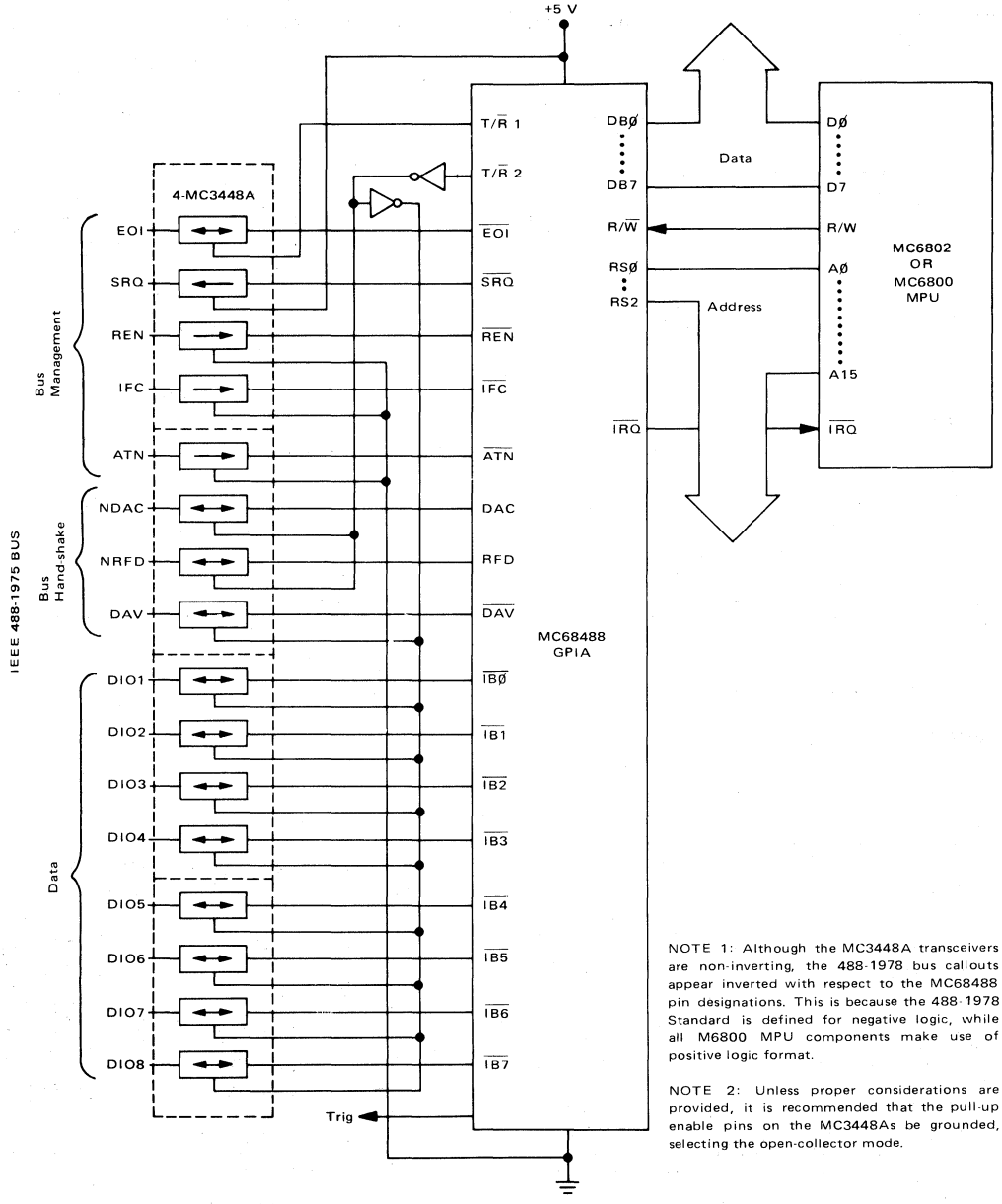


FIGURE 8 – SIMPLE SYSTEM CONFIGURATION



NOTE 1: Although the MC3448A transceivers are non-inverting, the 488-1978 bus callouts appear inverted with respect to the MC68488 pin designations. This is because the 488-1978 Standard is defined for negative logic, while all M6800 MPU components make use of positive logic format.

NOTE 2: Unless proper considerations are provided, it is recommended that the pull-up enable pins on the MC3448As be grounded, selecting the open-collector mode.

MOTOROLA

MC3870

Advance Information

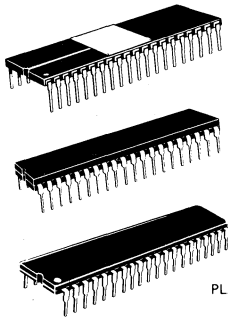
MOS
(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

SINGLE-CHIP MICROCONTROLLER

SINGLE-CHIP MICROCONTROLLER

The MC3870 is a monolithic 8-bit microcomputer utilizing ion-implanted, N-channel silicon-gate technology and advanced circuit design techniques. The single-chip MC3870 offers maximum cost effectiveness in a wide range of control and logic replacement applications.

- Software Compatible with F8 Family
- 2048 Byte Mask Programmable ROM
- 64 Byte Scratchpad RAM
- 32 Bits (4 Ports) TTL-Compatible I/O
- Programmable Binary Timer
  - Interval Timer Mode
  - Pulse Width Measurement Mode
  - Event Counter Mode
- External Interrupt
- Crystal, LC, RC, External
- Low Power (275 mW Typ.)
- Single +5 Volt ±10% Power Supply



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

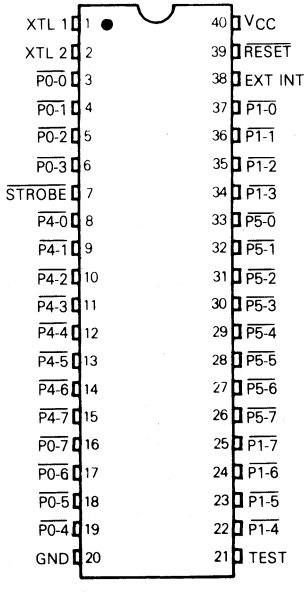
3

ABSOLUTE MAXIMUM RATINGS*

	Operating Temperature	
	0 to 70°C	−40 to +85°C
Temperature Under Bias	−20°C to +85°C	−50°C to +100°C
Storage Temperature	−65°C to +150°C	−65°C to +150°C
Voltage on any Pin with Respect to Ground (Except open-drain pins and TEST)	−1.0 V to +7 V	−1.0 V to +7 V
Voltage on TEST with Respect to Ground	−1.0 V to +9 V	−1.0 V to +9 V
Voltage to Open-Drain Pins with Respect to Ground	−1.0 V to +13.5 V	−1.0 V to +13.5 V
Power Dissipation	1.5 W	1.5 W
Power Dissipation by any One I/O Pin	60 mW	60 mW
Power Dissipation by All I/O Pins	600 mW	600 mW

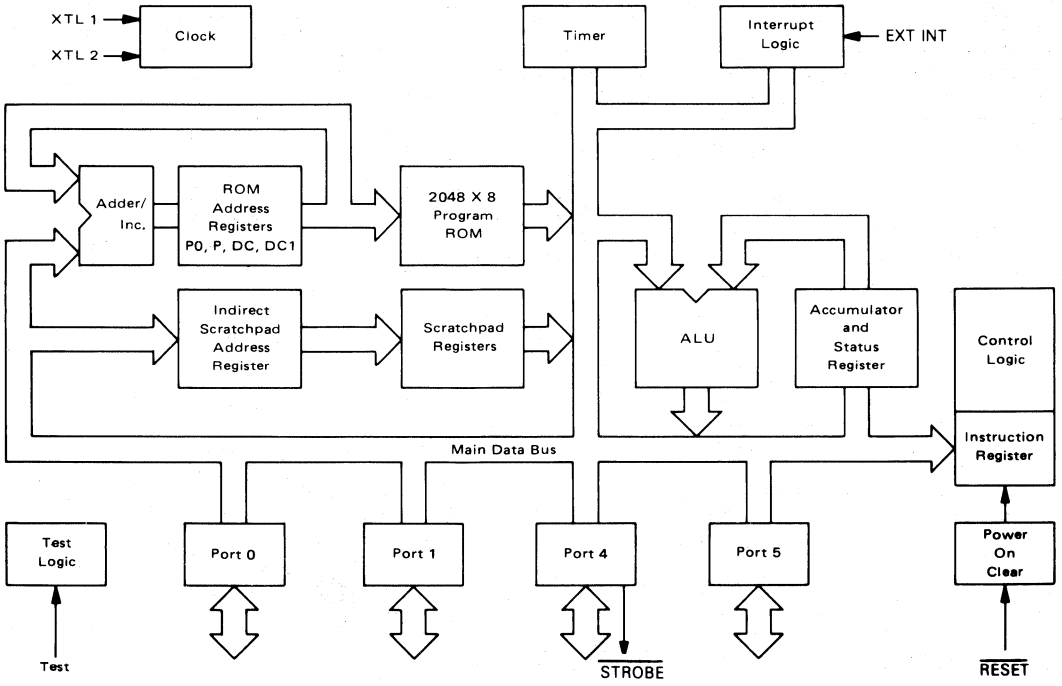
*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

PIN ASSIGNMENT



This is advance information and specifications are subject to change without notice.
Published by Motorola Inc. with permission of Mostek, Inc.

FIGURE 1 — BLOCK DIAGRAM



3

AC CHARACTERISTICS

Signal	Symbol	Parameter	0 to 70°C		−40° to +85°C		Unit	Notes
			Min	Max	Min	Max		
XTL1 XTL2	to	Time Base Period, all clock modes	250	1000	250	500	ns	
	tex(H)	External clock pulse width high	90	700	100	390	ns	
	tex(L)	External clock pulse width low	100	700	110	390	ns	
ϕ	tϕ	Internal ϕ clock	2t$_0$			2t$_0$		
WRITE	tw	Internal WRITE Clock period	4tϕ 6tϕ			4tϕ 6tϕ		Short Cycle Long Cycle
I/O	tdI/O	Output delay from internal WRITE clock	0	1000	0	1200	ns	50 pF plus one TTL load
	tsI/O	Input setup time to internal WRITE clock	1000		1200		ns	
$\overline{\text{STROBE}}$	t$_{I/O-s}$	Output valid to $\overline{\text{STROBE}}$ delay	3tϕ − 1000	3tϕ + 250	3tϕ − 1200	3tϕ + 300	ns	I/O load = 50 pF + 1 TTL load
	t$_{sL}$	$\overline{\text{STROBE}}$ low time	8tϕ − 250	12tϕ + 250	8tϕ − 300	12tϕ + 300	ns	$\overline{\text{STROBE}}$ load = 50 pF + 3 TTL loads
$\overline{\text{RESET}}$	t$_{RH}$	$\overline{\text{RESET}}$ hold time, low	6tϕ + 750		6tϕ + 1000		ns	
	t$_{RPOC}$	$\overline{\text{RESET}}$ hold time, low for power clear	power supply rise time + 0.1		power supply rise time + 0.15		ms	
EXT INT	t$_{EH}$	EXT INT hold time in active and inactive state	6tϕ + 750		6tϕ + 1000		ns	To trigger interrupt
			2tϕ		2tϕ		ns	To trigger timer

DC CHARACTERISTICS (I/O Power Dissipation ≤ 100 mW) (Note 2)

Symbol	Parameter	0 to 70 + C		−40 to +85°C		Unit	Conditions
		Min	Max	Min	Max		
V_{CC}	Power Supply Voltage	4.5	5.5	4.75	5.25	V	
V_{IHEX}	External Clock Input High Level	2.4	V_{CC}	2.4	V_{CC}	V	
V_{ILEX}	External Clock Input Low Level	−0.3	0.6	−0.3	0.6	V	
I_{IHEX}	External Clock Input High Current	—	100	—	130	μA	$V_{IHEX} = 2.40$
I_{ILEX}	External Clock Input Low Current	—	−100	—	−130	μA	$V_{ILEX} = 0.60$
$V_{IHI/O}$	Input High Level, I/O Pins	2.0	V_{CC}	2.2	V_{CC}	V	Standard Pullup
		2.0	13.2	2.2	13.2	V	Open Drain (1)
V_{IHR}	Input High Level, $\overline{RESET}$	2.0	V_{CC}	2.2	V_{CC}	V	
V_{IHEI}	Input High Level, EXT INT	2.0	V_{CC}	2.2	V_{CC}		
V_{IL}	Input Low Level	−0.3	0.8	−0.3	0.7	V	(1)
I_{IL}	Input Low Current, All Pins with Standard Pullup Resistor	—	−1.6	—	−1.9	mA	$V_{IN} = 0.4$ V
I_L	Input Leakage Current, Open Drain Pins,	—	+10	—	+18	μA	$V_{IN} = 13.2$ V
	and Inputs with No Pullup Resistor	—	−5	—	−8		$V_{IN} = 0.2$ V
I_{OH}	Output High Current Pins with Standard Pullup Resistor	−100	—	−90	—	μA	$V_{OH} = 2.4$ V
I_{OHDD}	Output High Current Direct Drive Pins	−1.5	—	−1.3	—	mA	$V_{OH} = 1.5$ V
		—	−8.5	—	−11		$V_{OH} = 0.7$ V
I_{OHS}	STROBE Output High Current	−300	—	−270	—	μA	$V_{OH} = 2.4$ V
I_{OL}	Output Low Current	1.8	—	1.65	—	mA	$V_{OL} = 0.4$ V
I_{OLS}	STROBE Output Low Current	5.0	—	4.5	—	mA	$V_{OL} = 0.4$ V
I_{CC}	Power Supply Current	—	85	—	110	mA	Outputs Open
P_D	Power Dissipation	—	400	—	525	mW	Outputs Open

1. $\overline{RESET}$ and EXT INT have internal Schmitt triggers giving minimum 0.2 V hysteresis.
2. Power dissipation for I/O pins is calculated by $\Sigma(V_{CC} - V_{IL})(|I_{IL}|) = \Sigma(V_{CC} - V_{OH})(|I_{OH}|) = \Sigma(V_{OL})(I_{OL})$

TIMER AC CHARACTERISTICS

Definitions:

Error = Indicated time value − actual time value

$t_{psc} = t\phi \times$ Prescale Value

Interval Timer Mode:

Single interval error, free running (Note 3) ..... $\pm 6t\phi$
Cumulative interval error free running (Note 3) 0
Error between two Timer reads (Note 2) ..... $\pm (t_{psc} + t\phi)$
Start Timer to stop Timer error (Notes 1, 4) ..... $+t\phi$ to $-(t_{psc} + t\phi)$
Start Timer to read Timer error (Notes 1, 2) ..... $-5t\phi$ to $-(t_{psc} + 7t\phi)$
Start Timer to interrupt request error (Notes 1, 3) ..... $-2t\phi$ to $-8t\phi$
Load Timer to Stop Timer error (Note 1) ..... $+t\phi$ to $-(t_{pcs} + 2t\phi)$
Load Timer to read Timer error (Notes 1, 2) ..... $-5t\phi \pm$ to $-(t_{psc} + 8t\phi)$
Load Timer to interrupt request error (Notes 1, 3) ..... $-2t\phi$ to $-9t\phi$

Pulse Width Measurement Mode:

Measurement accuracy (Note 4) ..... $+t\phi$ to $-(t_{psc} + 2t\phi)$
Minimum pulse width of EXT INT pin ..... $2t\phi$

Event Counter Mode:

Minimum active time of EXT INT pin ..... $2t\phi$
Minimum inactive time of EXT INT pin ..... $2t\phi$

NOTES:

1. All times which entail loading, starting, or stopping the Timer are referenced from the end of the last machine cycle of the OUT or OUTS instruction.
2. All times which entail reading the Timer are referenced from the end of the last machine cycle of the IN or INS instruction.
3. All times which entail the generation of an interrupt request are referenced from the start of the machine cycle in which the appropriate interrupt request latch is set. Additional time may elapse if the interrupt request occurs during a privileged or multicycle instruction.
4. Error may be cumulative if operation is repetitively performed.

FIGURE 2 — $\overline{\text{STROBE}}$ SOURCE CAPABILITY
(TYPCIAL AT V_{CC} = 5 V, T_A = 25°C)

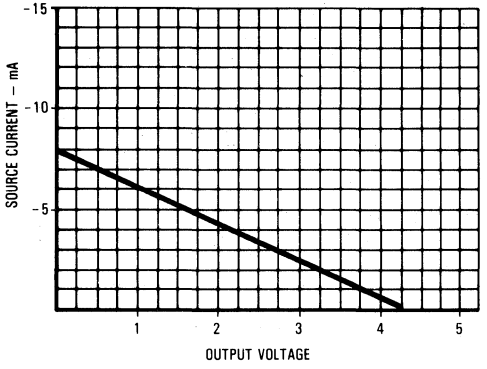
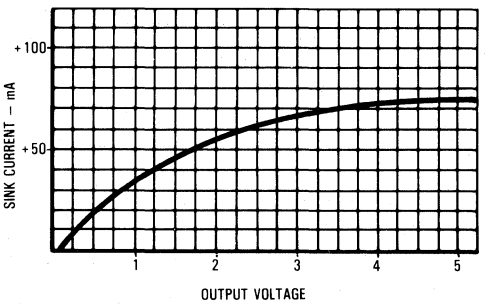


FIGURE 3 — $\overline{\text{STROBE}}$ SINK CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)



3

FIGURE 4 — STANDARD I/O PORT SOURCE CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

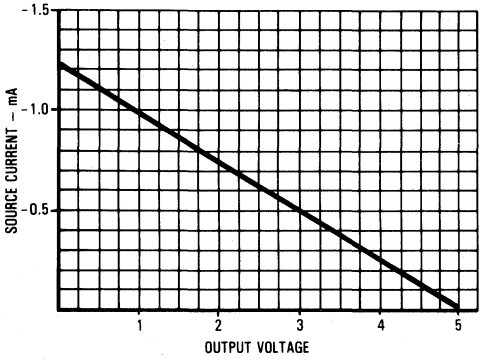


FIGURE 5 — DIRECT DRIVE I/O PORT SOURCE CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

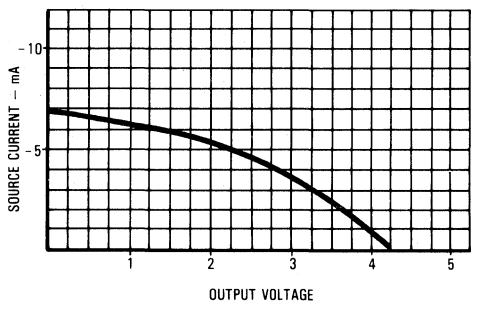


FIGURE 6 — I/O PORT SINK CAPABILITY
(TYPICAL AT V_{CC} = 5 V, T_A = 25°C)

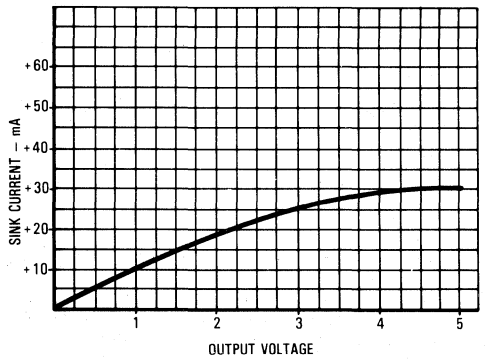


FIGURE 7 — MAXIMUM OPERATING TEMPERATURE
vs I/O POWER DISSIPATION

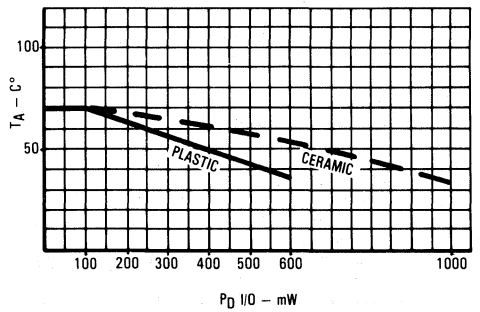


FIGURE 8 — MC3870 I_{DD} vs TEMPERATURE (V_{CC} = 5 V)

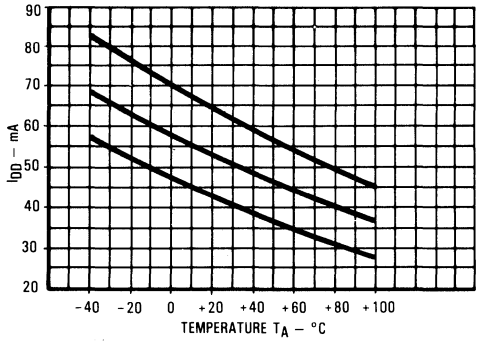
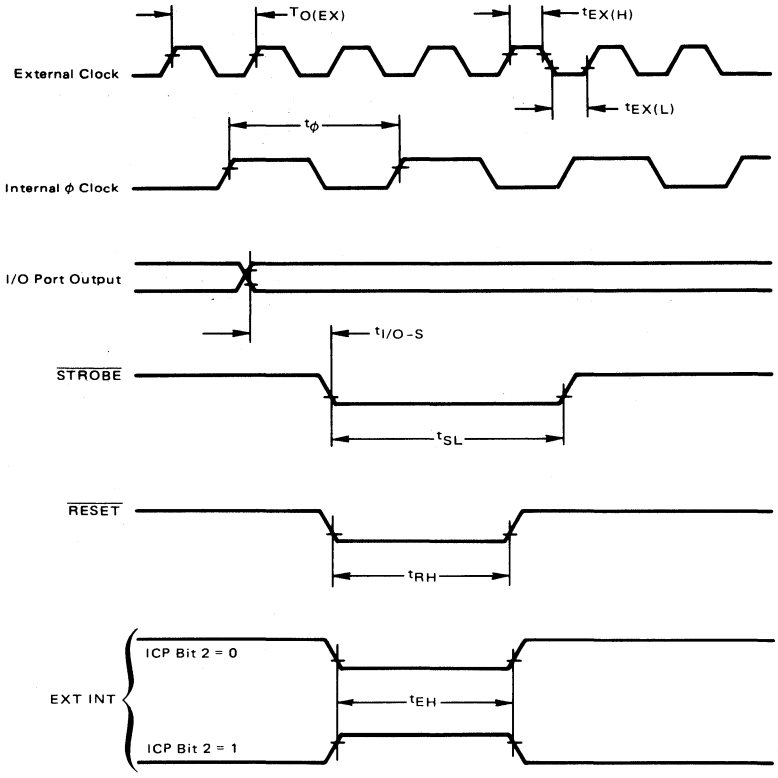
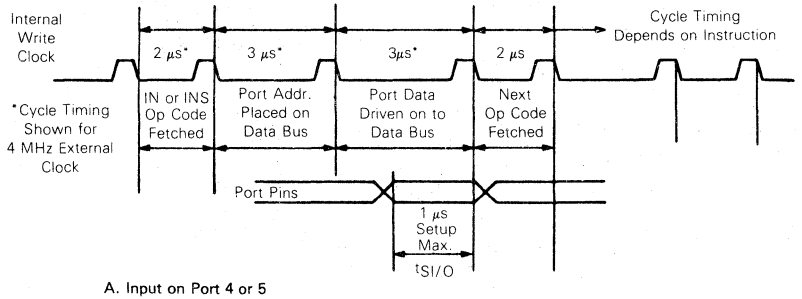


FIGURE 9 — ac TIMING DIAGRAM

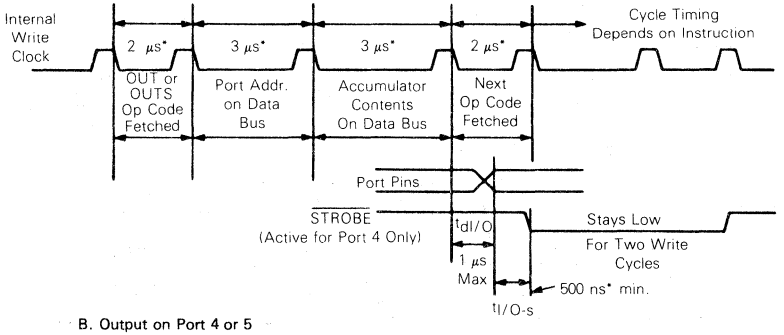


NOTE: All measurements are referenced to V_{IL} max., V_{IH} min., V_{OL} max., or V_{OH} min.

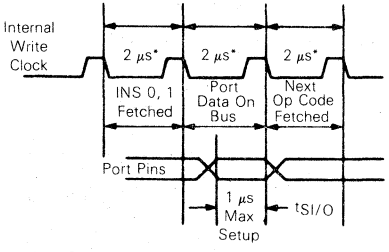
FIGURE 10 — INPUT/OUTPUT ac TIMING



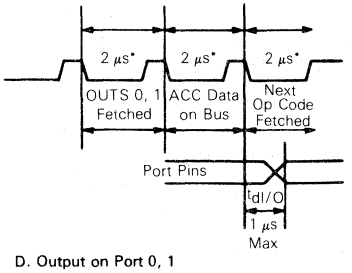
A. Input on Port 4 or 5



B. Output on Port 4 or 5



C. Input on Port 0 or 1



D. Output on Port 0, 1

MC3870 CLOCKS

The time base for the MC3870 may originate from one of four sources. The four configurations are shown in Figure 12. There is an internal 26 pF capacitor between XTL 1 and GND and an internal 26 pF capacitor between XTL 2 and GND. Thus, external capacitors are not necessarily required. In all external clock modes the external time-base frequency is divided by two to form the internal ϕ clock. The external clock frequency is divided by eight during short instruction cycles and is divided by twelve during long instruction cycles as given per instruction in the instruction set towards the end of this data sheet. To get the total instruction cycle time, divide the external clock frequency by eight, invert the number, then multiply by the short number of cycles. Then divide the external clock frequency by twelve and invert the number (Yx) then multiply by the number of long cycles. Add these two numbers to get the number of microseconds per instruction for a given clock frequency.

CRYSTAL SELECTION

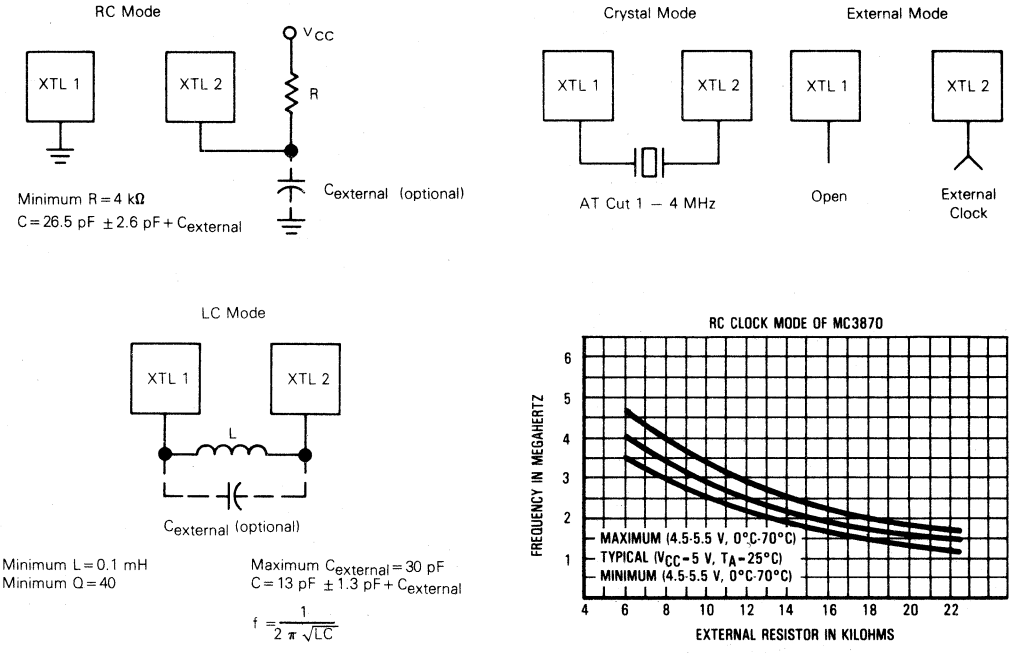
The use of a crystal as the time base is highly recommended as the frequency stability and reproducibility from system-to-system is unsurpassed. The 3870 has an internal divide-by-two to allow the use of inexpensive and widely available TV Color Burst Crystals (3.58 MHz). The following crystal parameters are suggested for 3870 applications:

a) Parallel Resonance, Fundamental Mode AT-Cut, HC-33/μ holder
b) Frequency Tolerance measured with 18 pF load (0.1% accuracy) — drive level 10 mW
c) Shunt capacitance (Co) = 7 pF max
d) Series resistance (Rs)

f = 1 MHz	Rs = 550 ohms max
f = 2 MHz	Rs = 300 ohms max
f = 3 MHz	Rs = 100 ohms max
f = 3.58 MHz	Rs = 100 ohms max
f = 4 MHz	Rs = 100 ohms max

3

FIGURE 11 — CLOCK CONFIGURATION



FUNCTIONAL PIN DESCRIPTION

$\overline{P0\text{-}0}$ - $\overline{P0\text{-}7}$ AND $\overline{P1\text{-}0}$ - $\overline{P1\text{-}7}$

Ports 1 and 2 are 16 lines which can be individually used as standard TTL-type inputs or latched outputs.

$\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$ AND $\overline{P5\text{-}0}$ - $\overline{P5\text{-}7}$

Ports 4 and 5 are 16 lines which can be individually used as standard, open drain, or direct drive type latched outputs or inputs. Refer to Figure 14 for more information on port options.

$\overline{\text{STROBE}}$

This output, which is normally high, provides a single low pulse after valid data is present on port 4 ($\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$) during an output instruction.

$\overline{\text{RESET}}$

This active low input is used to reset the internal state of the microcomputer. When allowed to go high, program execution begins at $000.

EXT/INT

This input is an external interrupt. Its active state is software programmable. The input is also used in conjunction with the timer for pulse width measurement and event counting.

XTL 1 AND XTL 2

These two inputs interface a crystal (1 to 4 MHz), LC network, RC network, or an external single-phase clock to the microcomputer.

TEST

TEST is an input used only in testing the MC3870. For normal circuit functionality, this pin is left unconnected or may be grounded.

V_{CC}

This is the power supply input (+5 V ±10%).

Pin Name	Description	Type
$\overline{P0\text{-}0}$ - $\overline{P0\text{-}7}$	I/O Port 0	Bidirectional
$\overline{P1\text{-}0}$ - $\overline{P1\text{-}7}$	I/O Port 1	Bidirectional
$\overline{P4\text{-}0}$ - $\overline{P4\text{-}7}$	I/O Port 4	Bidirectional
$\overline{P5\text{-}0}$ - $\overline{P5\text{-}7}$	I/O Port 5	Bidirectional
$\overline{\text{STROBE}}$	Ready Strobe	Output
EXT INT	External Interrupt	Input
$\overline{\text{RESET}}$	External Reset	Input
TEST	Test Line	Input
XTL 1, XTL 2	Time Base	Input
V_{CC}, GND	Power Supply Lines	Input

MC3870 ARCHITECTURE

This section describes the basic functional elements of the MC3870 as shown in the block diagram of Figure 1. A programming model is shown in Figure 12.

MAIN CONTROL LOGIC

The Instruction Register (IR) receives the operation code (OP code) or the instruction to be executed from the program ROM via the data bus. During all OP code fetches eight bits are latched into the IR. Some instructions are completely specified by the upper four bits of the OP code. In those instructions the lower four bits are an immediate register address or an immediate 4-bit operand. Once latched into the IR the main control logic decodes the instruction and provides the necessary control gating signals to all circuit elements.

ROM ADDRESS REGISTERS

There are four 11-bit registers associated with the 2K × 8 ROM. These are the Program Counter (P0), the Stack Register (P), the Data Counter (DC), and the Auxiliary Data Counter (DC1). The Program Counter is used to address instructions or immediate operands. P is used to save the contents of P0 during an interrupt or subroutine call. Thus, P contains the return address at which processing is to resume upon completion of the subroutine or the interrupt routine.

The Data Counter (DC) is used to address data tables. This register is auto-incrementing. Of the two data counters only DC can access the ROM. However, the XDC instruction allows DC and DC1 to be exchanged.

Associated with the address registers is an 11-bit Adder/Incrementer. This logic element is used to increment P0 or DC when required and is also used to add displacements to P0 on relative branches or to add the data bus contents to DC in the ADC (Add Data Counter) instruction.

2048 × 8 ROM

The microcomputer program and data constants are stored in the program ROM. When a ROM access is required, the appropriate address register (P0 or DC) is gated onto the ROM address bus and the ROM output is gated onto the main data bus. The first byte in the ROM is location zero.

SCRATCHPAD AND IS

The scratchpad provides 64 8-bit registers which may be used as general purpose RAM memory. The Indirect Scratchpad Address Register (IS) is a 6-bit register used to address the 64 registers. All 64 registers may be accessed using IS. In addition, the lower order 12 registers may also be directly addressed.

IS can be visualized as holding two octal digits. This division of IS is important since a number of instructions increment or decrement only the least-significant three bits of IS when referencing scratchpad bytes via IS. This makes it easy to reference a buffer consisting of contiguous scratchpad bytes. For example, when the low order octal digit is incremented or decremented IS is incremented from octal 27 (0'27') to 0'20' or is decremented from 0'20' to 0'27'. This feature of the IS is very useful in many program sequences. All six bits of IS may be loaded at one time or either half may be loaded independently.

Scratchpad registers 9 through 15 (decimal) are given mnemonic names (J, H, K, and Q) because of special linkages between these registers and other registers such as

the Stack Register. These special linkages facilitate the implementation of multi-level interrupts and subroutine nesting. For example, the instruction LR K,P stores the lower eight bits of the Stack Register into register 13 (K lower or KL) and stores the upper three bits of P into register 12 (K upper or KU).

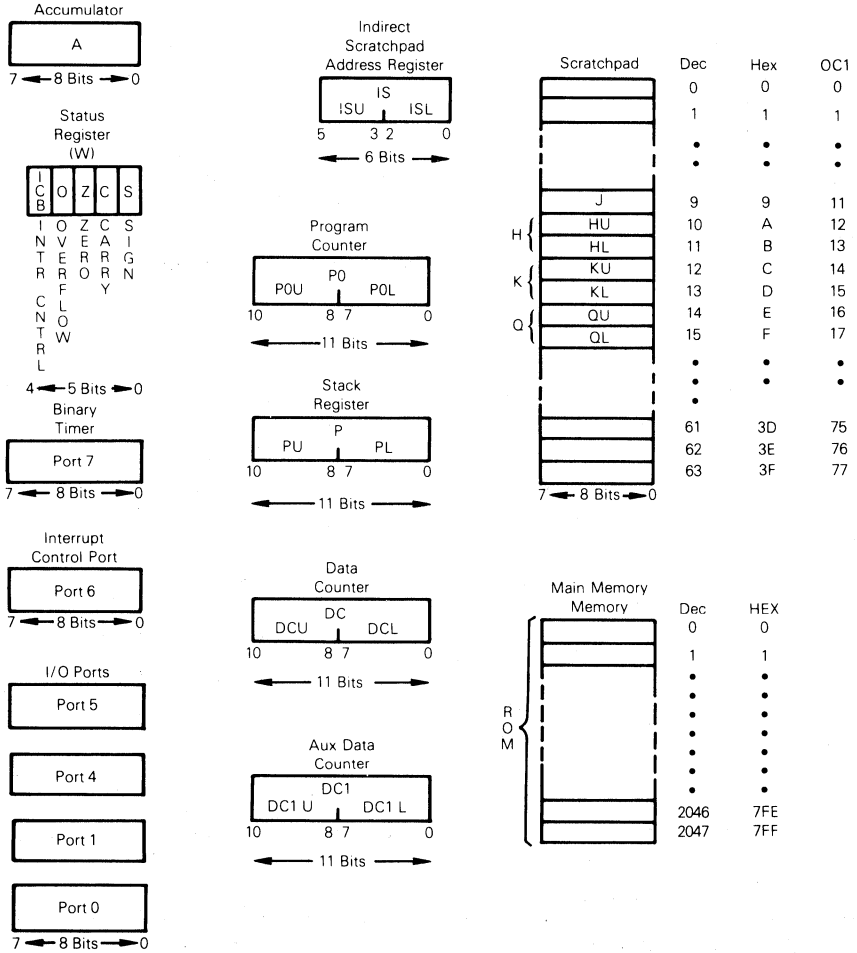
ARITHMETIC AND LOGIC UNIT (ALU)

After receiving commands from the main control logic, the ALU performs the required arithmetic or logic operations (using the data presented on the two input buses) and provides the result on the result bus. The arithmetic operations that can be performed in the ALU are binary add, decimal adjust, add with carry, decrement, and increment. The logic operations that can be performed are AND, OR, EXCLUSIVE OR, "1's" complement, shift right, and shift left. Besides providing the result on the result bus, the ALU also provides four signals representing the status of the result. These signals, stored in the Status Register (W), represent CARRY, OVERFLOW, SIGN, and ZERO condition of the result of the operation.

ACCUMULATOR (A)

The Accumulator (A) is the principal register for data manipulation within the 3870. The A serves as one input to the ALU for arithmetic or logical operations. The result of ALU operations are stored in the A.

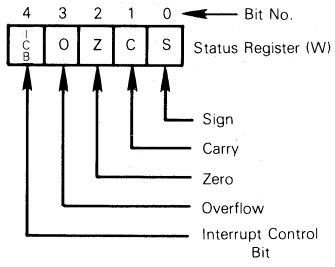
FIGURE 12 — MC3870 PROGRAMMABLE REGISTERS, PORTS, AND MEMORY MAP



THE STATUS REGISTER (W)

The Status Register (also called the W register) holds five status flags as shown in Figure 13.

FIGURE 13 — STATUS REGISTER (W)



Summary of Status Bits

OVERFLOW = $Carry_7 \oplus CARRY_6$

ZERO = $\overline{ALU_7} \wedge \overline{ALU_6} \wedge \overline{ALU_5} \wedge \overline{ALU_4} \wedge \overline{ALU_3} \wedge \overline{ALU_2} \wedge \overline{ALU_1} \wedge \overline{ALU_0}$

CARRY = $CARRY_7$

SIGN = $\overline{ALU_7}$

INTERRUPT CONTROL BIT (ICB)

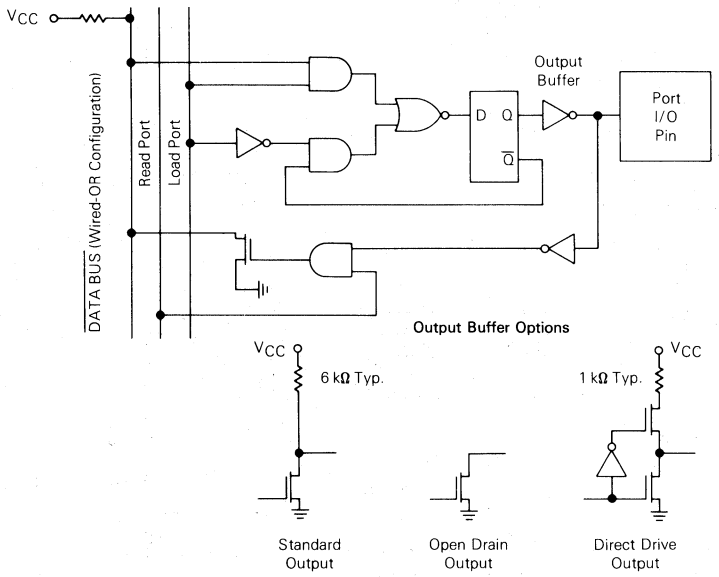
The ICB may be used to allow or disallow interrupts in the MC3870. This bit is not the same as the two interrupt enable bits in the Interrupt Control Port (ICP). If the ICB is set and the MC3870 interrupt logic communicates an interrupt request to the CPU section, the interrupt will be acknowledged and processed upon completion of the first non-privileged instruction. If the ICB is cleared an interrupt request will not be acknowledged or processed until the ICB is set.

I/O PORTS

The MC3870 provides four complete bidirectional Input/Output ports. These are ports 0, 1, 4, and 5. In addition, the Interrupt Control Port is addressed as port 6 and the binary timer is addressed as port 7. An output instruction (OUT or OUTS) causes the contents of A to be latched into the addressed port. An input instruction (IN or INS) transfers the contents of the port to A (port 6 is an exception which is described later). The schematic of an I/O pin and available output drive options are shown in Figure 14.

An output ready strobe is associated with port 4. This flag may be used to signal a peripheral device that the MC3870 has just completed an output of new data to port 4. The strobe provides a single low pulse shortly after the output operation is completely finished, so either edge may be used to signal the peripheral. $\overline{STROBE}$ may also be used as an input strobe simply by doing a dummy output of H '00' strobe to port 4 after completing the input operation.

FIGURE 14 — I/O PIN CONCEPTUAL DIAGRAM WITH OUTPUT BUFFER OPTIONS



Ports 0 and 1 are Standard Output type only.

Ports 4 and 5 may both be any of the three output options (programmable bit-by-bit).

The $\overline{STROBE}$ output is always configured similar to a Direct Drive Output except that it is capable of driving 3 TTL loads.

$\overline{RESET}$ and EXT INT may have standard 6 kΩ (typical) pullup or may have no pullup. These two inputs have Schmitt trigger inputs with a minimum of 0.2 volts of hysteresis.

TIMER AND INTERRUPT CONTROL PORT

The Timer is an 8-bit binary down counter which is software programmable to operate in one of three modes: the Interval Timer Mode, the Pulse Width Measurement Mode, or the Event Counter Mode. As shown in Figure 15, associated with the Timer are an 8-bit register called the interrupt control port, a programmable prescaler, and an 8-bit modulo-N register. A functional logic diagram is shown in Figure 16.

INTERRUPT CONTROL PORT (PORT 6)

The desired timer mode, prescale value, starting and stopping the timer, active level of the EXT INT pin, and local enabling or disabling of interrupts are selected by outputting the proper bit configuration from the Accumulator to the Interrupt Control Port (port 6) with an OUT or OUTS instruction. Bits within the Interrupt Control Port are defined as follows:

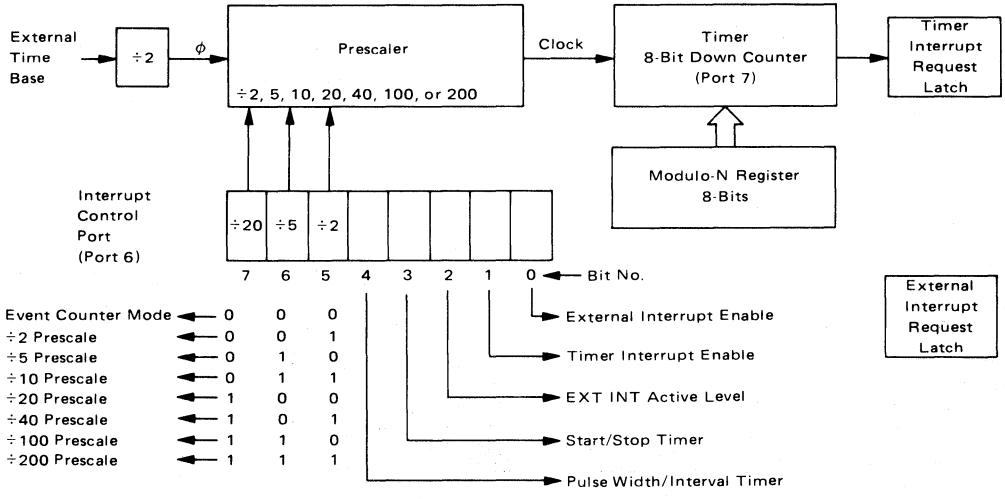
Bit 0 — External Interrupt Enable
Bit 1 — Timer Interrupt Enable
Bit 2 — EXT INT Active Level
Bit 3 — Start/Stop Timer
Bit 4 — Pulse Width/Interval Timer
Bit 5 — ÷2 Prescale
Bit 6 — ÷5 Prescale
Bit 7 — ÷20 Prescale

A special situation exists when reading the Interrupt Control Port (with IN or INS instruction). The Accumulator is *not* loaded with the content of the ICP; instead, Accumulator bits 0 through 6 are loaded with "0's" while bit 7 is loaded with the logic level being applied to the EXT INT pin, thus allowing the status of EXT INT to be determined without the necessity of servicing an external interrupt request. When reading the Interrupt Control Port (port 6) bit 7 of the Accumulator is loaded with the actual logic level being applied to the EXT INT pin, regardless of the status of ICP bit 2 (the EXT INT Active Level bit); that is, if EXT INT is a + *5 V bit 7* of the Accumulator is set to a logic "1", but if EXT INT is at GND then Accumulator bit 7 is reset to logic "0". This capability is useful in establishing a high speed polled handshake procedure or for using EXT INT as an extra input pin if external interrupts are not required and the Timer is used only in the Interval Timer Mode. However, if it is desirable to read the contents of the ICP then one of the 64 scratchpad registers or one byte of RAM may be used to save a copy of whatever is written to the ICP.

The rate at which the timer is clocked in the External Timer Mode is determined by the frequency of an internal ϕ clock and by the division value selected for the prescaler. (The internal ϕ clock operates at one-half the external time-base frequency). If ICP bit 5 is set and bits 6 and 7 are cleared, the prescaler divides ϕ by 2. Likewise, if bit 6 or 7 is individually set, the prescaler divides ϕ by 5 or 20 respectively. Combinations of bits 5, 6, and 7 may also be selected. For example, if bits 5 and 7 are set while 6 is cleared the prescaler will divide by 40. Thus, possible prescaler values are ÷2, ÷5, ÷10, ÷20, ÷40, ÷100, and ÷200.

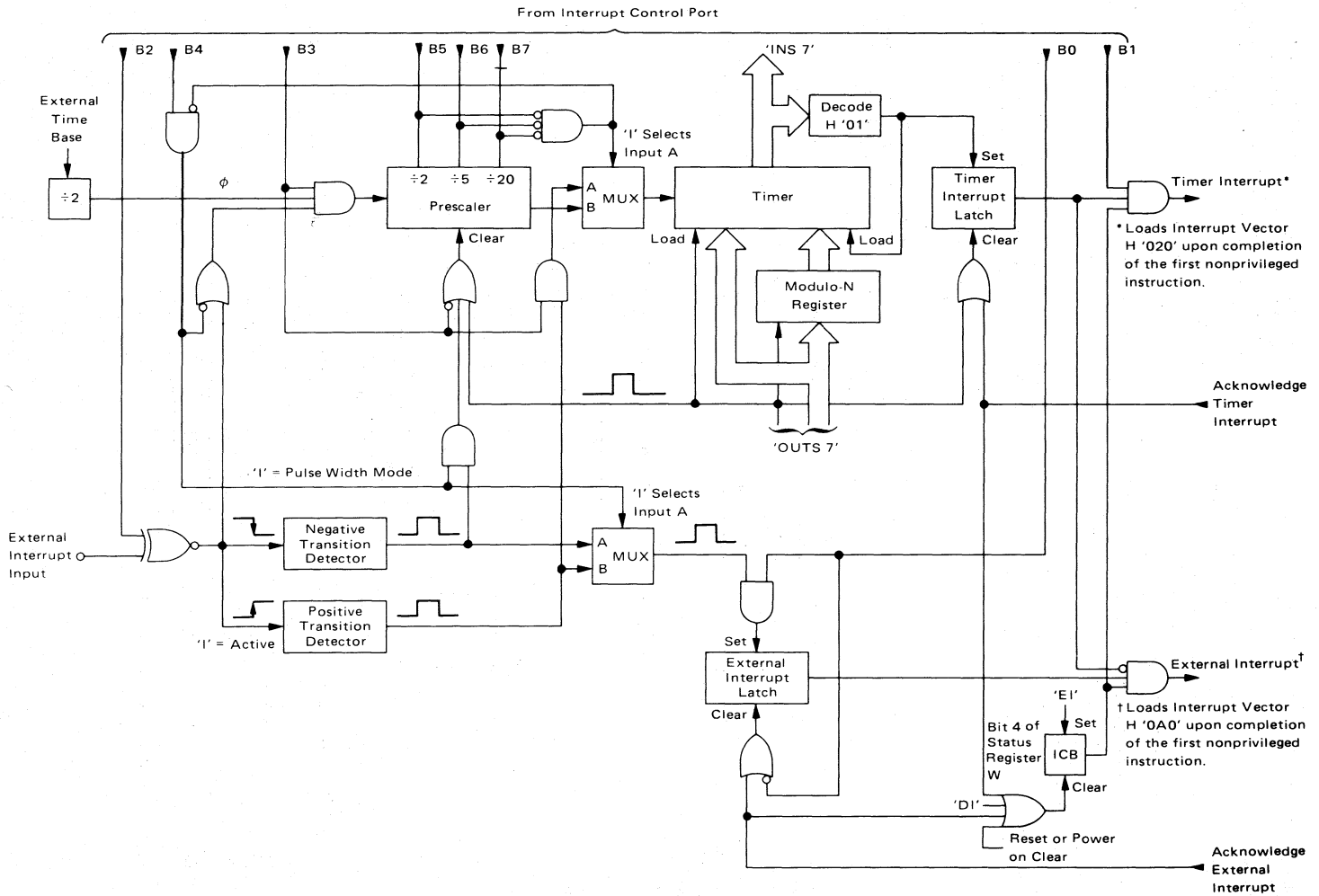
Any of three conditions will cause the prescaler to be reset: whenever the timer is stopped by clearing ICP bit 3, execution of an output instruction to Port 7, (the timer is assigned port address 7), or on the trailing edge transition of the EXT INT pin when in the Pulse Width Measurement Mode. These last two conditions are explained in more detail below.

FIGURE 15 — TIMER AND CONTROL PORT BLOCK DIAGRAM



NOTE: See Figure 17 for a more detailed functional diagram.

FIGURE 16 — MC3870 TIMER/INTERRUPT FUNCTIONAL DIAGRAM



An OUT or OUTS instruction to Port 7 will load the content of the Accumulator to both the Timer and the 8-bit modulo-N register, reset the prescaler, and clear any previously stored timer interrupt request. As previously noted, the Timer is an 8-bit down counter which is clocked by the prescaler in the Interval Timer mode and in the Pulse Width Measurement Mode. The prescaler is not used in the Event Counter Mode. The modulo-N register is a buffer whose function is to save the value which was most recently outputted to Port 7. The modulo-N register is used in all three timer modes.

Interval Timer Mode — When ICP bit 4 is cleared (logic 0) and at least one prescale bit is set, the Timer operates in the Interval Timer Mode. when bit 3 of the ICP is set the Timer will start counting down from the modulo-N value. After counting down to H '01', the Timer returns to the modulo-N value at the next count. On the transition from H '01' to H 'N' the Timer sets a timer interrut request latch. Note that the interrupt request latch is set by the transition to H 'N' and not be the presence of H 'N' in the timer, thus allowing a full 256 counts if the modulo-N register is preset to H '00'. If bit 1 of the ICP is set, the interrupt request is passed on to the CPU section of the MC3870. However, if bit 1 of the ICP is a logic 0 the interrupt request is not passed on to the CPU section but the interrupt request latch remains set. If ICP bit 1 is subsequently set, the interrupt request will then be passed on to the CPU section. (Recall from the discussion of the Status Register's Interrupt Control Bit that the interrupt request will be acknowledged by the CPU section only if ICB is set.) Only two events can reset the timer interrupt request latch: when the timer interrupt request latch is acknowledged by the CPU section, or when a new load of the modulo-N register is performed.

Consider an example in which the modulo-N register is loaded with H '64' (decimal 100). The timer interrupt request latch will be set at the 100th count following the timer start and the timer interrupt request latch will repeatedly be set on precise 100 counter intervals. If the prescaler is set at $\div 40$ the timer interrupt request latch will be set every 4000 ϕ clock periods. For a 2 MHz ϕ clock (4 MHz time-base frequency) this will produce 2 millisecond intervals.

The range of possible intervals is from 2 to 51,200 ϕ clock periods (1 μs to 25.6 ms for a 2 MHz clock). However, approximately 50 ϕ periods is a practical minimum because the time between setting the interrupt request latch and the execution of the first instruction of the interrupt service routine is at least 29 ϕ periods (the response time is dependent upon how many privileged instructions are encountered when the request occurs); 29 is based on the timer interrupt occuring at the beginning of a non-privileged short instruction. To establish time intervals greater than 51,200 ϕ clock periods is a simple matter of using the timer interrupt service routine to count the number of interrupts, saving the result in one or more of the scratchpad registers until the desired interval is achieved. With this technique virtually any time interval, or several time intervals, may be generated.

The Timer may be read at any time and in any mode using an input instruction (IN 7 or INS 7) and may take place "on the fly" without interfering with normal timer operation. Also, the Timer may be stopped at any time by clearing bit 3 of the ICP. The Timer will hold its current contents indefinitely and will resume counting when bit 3 is again set. Recall however that the prescaler is reset whenever the Timer is stopped; thus a series of starting and stopping will result in a cumulative truncation error.

A summary of other timer errors is given in the timing section of their specification. For a free running timer in the Interval Timer Mode the time interval between any two interrupt requests may be in error by $\pm 6\,\phi$ clock periods although the cumulative error over many intervals is zero. The prescaler and Timer generate precise intervals for setting the timer interrupt request latch but the time out may occur at any time within a machine cycle. (There are two types of machine cycles; short cycles which consist of 4 ϕ clock periods and long cycles which consist of 6 ϕ clock periods.) Interrupt requests are synchronized with the internal machine clock thus, giving rise to the possible $\pm 6\,\phi$ error. Additional errors may arise due to the interrupt request occuring while a privileged instruction or multicycle instruction is being executed. Nevertheless, for most applications all of the above errors are neglibible, especially if the desired time interval is greater than 1 ms.

Pulse Width Measurement Mode — When ICP bit 4 is set (logic 1) and at least one prescale bit is set, the Timer operates in the Pulse Width Measurement Mode. This mode is used for accurately measuring the duration of a pulse applied to the EXT INT pin. The Timer is stopped and the prescaler is reset whenever EXT INT is at its inactive level. The active level of EXT INT is defined by ICP bit 2; if cleared, EXT INT is active low; if set, EXT INT is active high. If ICP bit 3 is set, the prescaler and Timer will start counting when EXT INT transitions to the active level. When EXT INT returns to the inactive level the Timer then stops, the prescaler resets, and *if ICP bit 0* is set an external interrupt request latch is set. (Unlike timer interrupts, external interrupts are not latched if the ICP Interrupt Enable bit is not set.)

As in the Interval Timer Mode, the Timer may be read at any time, may be stopped at any time by clearing ICP bit 3, the prescaler and ICP bit 1 function as previously described, and the Timer still functions as an 8-bit binary down counter with the timer interrupt request latch being set on the Timer's transition from H '01' to H 'N'. Note that the EXT INT pin has nothing to do with loading the Timer; its action is that of automatically starting and stopping the Timer and of generating external interrupts. Pulse widths longer than the prescale value times the modulo-N value are easily measured by using the timer interrupt service routine to store the number of timer interrupts in one or more scratchpad registers.

As for accuracy, the actual pulse duration is typically slightly longer than the measured value because the status of the prescaler is not readable and is reset when the Timer is stopped. Thus, for maximum accuracy, it is advisable to use a small division setting for the prescaler.

Event Counter Mode — When ICP bit 4 is cleared and all prescale bits (ICP bits 5, 6, and 7) are cleared, the Timer operates in the Event Counter Mode. This mode is used for counting pulses applied to the EXT INT pin. If ICP bit 3 is set, the Timer will decrement on each transition from the inactive level to the active level of the EXT INT pin. The prescaler is not used in this mode; but as in the other two timer modes,

the Timer may be read at any time, may be stopped at any time by clearing ICP bit 3, ICP bit 1 functions previously described, and the timer interrupt request latch is set on the Timer's transition from H '01' to H 'N'.

Normally ICP bit 0 should be kept cleared in the Event Counter Mode; otherwise, external interrupts will be generated on the transition from the inactive level to the active level of the EXT INT pin.

For the Event Counter Mode, the minimum pulse width required on EXT INT is 2 ϕ clock periods and the minimum inactive time is 2 ϕ clock periods; therefore, the maximum repetition rate is 500 kHz.

External Interrupts — When the timer is in the Interval Timer Mode the EXT INT pin is available for non-timer related interrupts. If ICP bit 0 is set, an external interrupt request latch is set when there is a transition from the inactive level to the active level of EXT INT. (EXT INT is an edge-triggered input.) The interrupt request is latched until either acknowledged by the CPU section or until ICP bit 0 is cleared (unlike timer interrupt requests which remain latched even when ICP bit 1 is cleared). External interrupts are handled in the same fashion when the Timer is in the Pulse Width Measurement Mode or in the Event Counter Mode, except that only in the Pulse Width Measurement Mode the external interrupt request latch is set on the trailing edge of EXT INT; that is, on the transition from the active level to the inactive level.

INTERRUPT HANDLING

When either a timer or an external interrupt request is communicated to the CPU section of the MC3870, it will be acknowledged and processed at the completion of the first non-privileged instruction if the Interrupt Control Bit of the Status Register is set. If the Interrupt Control Bit is not set, the interrupt request will continue until either the Interrupt Control Bit is set and the CPU section acknowledges the interrupt or until the interrupt request is cleared as previously described.

If there is both a timer interrupt request and an external interrupt request when the CPU section starts to process the requests, the timer interrupt is handled first.

When an interrupt is allowed the CPU section will request that the interrupting element pass its interrupt vector address to the Program Counter via the data bus. The vector address for a timer interrupt is H '020'. The vector address for external interrupts is H '0A0'. After the vector address is passed to the Program Counter, the CPU section sends an acknowledge signal to the appropriate interrupt request latch which clears that latch. The execution of the interrupt service routine will then commence. The return address of the original program is automatically saved in the Stack Register, P.

The Interrupt Control Bit of W (Status Register) is automatically reset when an interrupt request is acknowledged. It is then the programmer's responsibility to determine when ICB will again be set (by executing an EI instruction). This action prevents an interrupt service routine from being interrupted unless the programmer so desires.

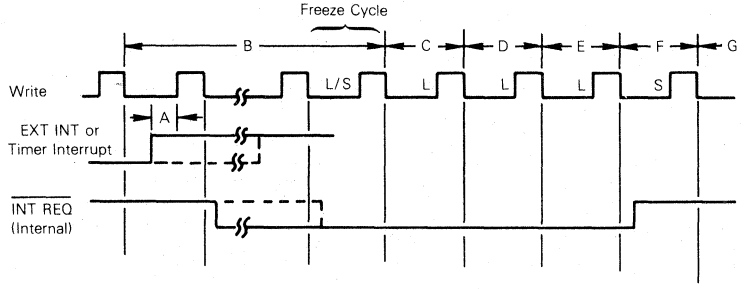
Figure 17 details the interrupt sequence which occurs whether the interrupt request is from an external source via EXT INT or from the MC3870's internal timer. Events are labeled with the letters A through G and are described below.

Event A — An interrupt request must satisfy a hold time requirement as specified in the AC Characteristics in order to guarantee that it is valid on the rising edge of the WRITE clock.

Event B — Event B represents the instruction being executed when the interrupt occurs. The last cycle of B is normally the instruction fetch for the next cycle. However, if B is not a privileged instruction and the CPU's Interrupt Control Bit is set, then the last cycle becomes a "freeze" cycle rather than a fetch. At the end of the freeze cycle the interrupt request latches are inhibited from altering the interrupt daisy-chain so that sufficient time will be allowed for the daisy-chain to settle. (If B is a privileged instruction, the instruction fetch is not replaced by a freeze cycle; instead, the fetch is performed and the next instruction is executed. Although unlikely to be encountered, a series of privileged instructions will be sequentially executed without interrupt. One more instruction, called a 'protected' instruction, will always be executed after the last privileged instruction. The last cycle of the protected instruction then performs the freeze.)

The dashed lines on EXT INT illustrate the last opportunity for EXT INT to cause the last cycle of a non-protected instruction to become a freeze cycle.

FIGURE 17 — INTERRUPT SEQUENCE



The freeze cycle is a short cycle (4 ϕ clock periods) in all cases except where B is the Decrement Scratchpad instruction, in which case the freeze cycle is a long cycle (6 ϕ clock periods).

INT REQ goes low on the next negative edge of WRITE if both PRI IN is low and the appropriate interrupt enable bit of the Interrupt Control Port is set. Both INT REQ and WRITE are internal signals.

Event C — A NO-OP long cycle to allow time for the internal priority chain to settle.

Event D — The Program Counter (PO) is pushed to the stack register (P) in order to save the return address. The interrupt circuitry places the lower 8 bits of the interrupt vector address onto the data bus. This is always a long cycle.

Event E — A long cycle in which the interrupt circuitry places the upper 8 bits of the interrupt vector address onto the data bus.

Event F — A short cycle in which the interrupting interrupt request latch is cleared. Also, the CPU's Interrupt Control Bit is cleared, thus disabling interrupts until an EI instruction is performed. The fetch of the next instruction from the interrupt address.

Event G — Begin execution of the first instruction of the interrupt service routine.

SUMMARY OF INTERRUPT SEQUENCE

For the MC3870 the interrupt response time is defined as the time elapsed between the occurrence of EXT INT going active (or the Timer transitioning to H 'N') and the beginning of execution of the first instruction of the interrupt service routine. The interrupt response time is a variable dependent upon what the microprocessor is doing when the interrupt request occurs. As shown in Figure 17, the minimum interrupt response time is 3 long cycles plus 2 short cycles plus one WRITE clock pulse width plus a setup time of EXT INT prior to the leading edge of the WRITE pulse — a total of 27 ϕ clock periods plus the setup time. At a 2 MHz ϕ this is 14.25 μs. Although the maximum could theoretically be infinite, a practical maximum is 35 μs (based on the interrupt request occurring near the beginning of a PI and LR K, P sequence).

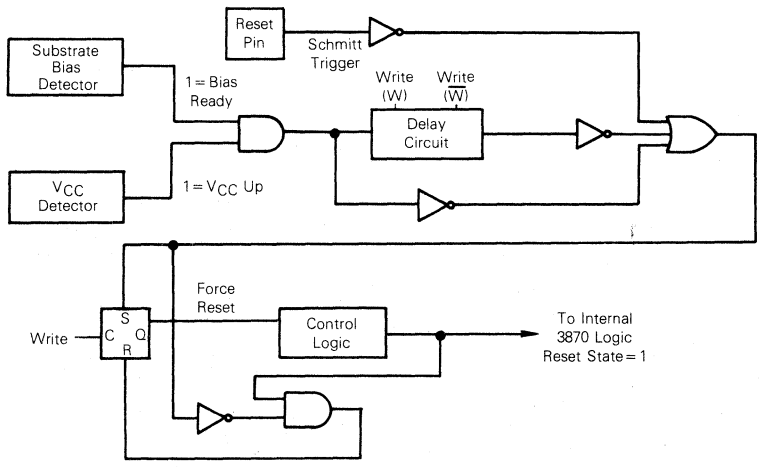
POWER-ON RESET

The intent of the Power-On Reset circuitry on the MC3870 is to automatically reset the device following a typical power-up situation, thus saving external reset circuitry in many applications. This circuitry is not guaranteed to sense a "Brown Out" (low voltage) condition nor is it guaranteed to operate under all possible power-on situations.

Three conditions are required before the MC3870 will leave the reset state and begin operation. Refer to Figure 18 as an aid to the following descriptions. The On-Chip V_{CC} detector senses a minimum value of V_{CC} before it will allow the MC3870 to operate. The threshold of this detector is set by analog circuitry because a stable voltage reference is not available with n-channel MOS processing. Processing variations will cause this threshold to vary from a low of 3.0 volts to a high of 4.3 volts with 3.5 volts being typical.

The MC3870 uses a substrate bias as a technique to provide improved performances versus power consumption relative to conventional grounded substrate approaches. This bias generator may start operating as low as $V_{CC}=3$ volts on some devices while others may require $V_{CC}=4$ volts in order to get adequate substrate bias. Until the substrate reaches the proper bias, the MC3870 will not be released from the reset state. The final condition required is that the clocks of the MC3870 must be functioning. Typically the clocks will start to function at V_{CC} equal to 3 to 3.5 volts but since the part is tested at 4.5 volts, Motorola can not guarantee any operation below 4.5 volts. The output of the delay circuit in Figure 18 will stay low until the clocks

FIGURE 18 — POWER ON RESET BLOCK DIAGRAM



start to function. If the input to the delay circuit is high, typically after 100 cycles of the WRITE clock (800 cycles of the external clock) the output of the delay circuit will go high allowing the MC3870 to begin execution.

If V_{CC} falls to ground for at least a few hundred nanoseconds the output of the delay circuit will go low immediately and the MC3870 will reset.

The internal logic may detect a valid V_{CC}, bias and clocks at $V_{CC} = 3.5$ volts and allow the MC3870 to start executing after the time delay. With a slowly rising power supply, the part may start running before V_{CC} is above 4.5 volts which is below the guaranteed voltage range. When power-on-clear is required with a slowly rising power supply, an external capacitor must be used on the $\overline{\text{RESET}}$ pin to hold it below 0.8 volts until V_{CC} is stable above 4.5 volts. (Note: The option to disconnect the internal pullup resistor on $\overline{\text{RESET}}$ is available which allows the use of a larger external pullup resistor and a small capacitor on $\overline{\text{RESET}}$.)

In many applications it is desirable if the unit does an automatic power-on-clear, but not mandatory. The unit will have a RESET push button and if the unit does not power-up correctly or malfunctions because of some disturbance on the V_{CC} line, the operator will simply press RESET and restore normal operation. It is for these applications that the internal power-on-clear circuitry was designed.

In some applications it is required that the microcomputer continue to run properly without operator intervention after brown-outs, power line disturbances, electrical noise, computer malfunction due to a programming bug, or any other disturbance except a catastrophic failure of some component.

One concept used to keep computers running is that of the "WATCHDOG TIMER". The computer is programmed to periodically reset the watchdog timer during the normal execution of its program (this is easily done in the MC3870 as its normal application is in some control function which is typically periodic). As long as the computer continues to execute its program the watchdog timer is continually reset and never times out. Should the computer stop executing its program for whatever reason, the watchdog timer will time out producing a RESET pulse to the CPU re-starting execution. This is a very positive way to assure that the computer is doing its job, i.e., executing the program. It is important that the software driving the watchdog timer test as many functional blocks (timer, ALU, scratchpad RAM, and ports) of the MC3870 as possible before resetting the watchdog timer. This is because operation of the MC3870, with an out of specification power supply, may allow some of the functions to operate correctly while other functions are not operable.

Motorola can guarantee correct operation of the MC3870 only while the V_{CC} voltage remains within its specified limits. If proper operation of the MC3870 must be guaranteed after a disturbance on the V_{CC} line, then an external circuit must be used to monitor the V_{CC} line and produce $\overline{\text{RESET}}$ to the MC3870 whenever V_{CC} is out of the specified limits.

A related characteristic to power-on-clear is the startup time of the basic timing element. The LC and RC oscillators begin to function almost immediately once V_{CC} is high enough to allow the on-board oscillator to operate ($V_{CC} = 3.5$ V). Operation with a crystal is partly mechanical and some start time is required to get the mass of the crystal into vibrational motion. This time is basically dependent on the frequency (mass) of the crystal. 4 MHz crystals typically require about 2-3 ms to start while 1 MHz crystals require 60-70 ms to start oscillating. Of course, this time may vary greatly from crystal to crystal and is also a function of the power supply rise time characteristic, however, the high-frequency crystals start faster and are definitely recommended (i.e., 3-4 MHz).

The condition of the port pins during the power-in-clear sequence is often asked. The port pins or the STROBE line cannot be specified until V_{CC} reaches 4.5 V and the MC3870 enters the RESET state. Before this, the port pins may stay at V_{SS}, may track V_{CC} as it rises, or they may track V_{CC} part way up then return to V_{SS} (ports 4 and 5 will go to V_{CC} once the clocks are running and the MC3870 has sufficient V_{CC} to properly operate the internal control logic and I/O ports).

EXTERNAL RESET

When $\overline{\text{RESET}}$ is taken low, the content of the Program Counter is pushed to the Stack Register and then the Program Counter and the ICB bit of the W Status Register are cleared. The original Stack Register content is lost. Ports 4, 5, 6, and 7 are loaded with H '00'. The contents of all other registers and ports are unchanged or undefined. When RESET is taken high, the first program instruction is fetched from ROM location H '000'. When an external reset of the MC3870 occurs, P0 is pushed into P and the old contents of P are lost. It must be noted that an external reset is recognized at the start of a machine cycle and not necessarily at the end of an instruction. Thus, if the MC3870 is executing a multi-cycle instruction, that instruction is not completed and the contents of P upon reset may not necessarily be the address of the instruction that would have been executed next. It may, for example, point to an immediate operand if the reset occurred during the second cycle of an LI or CI instruction. Additionally, several instructions (JMP, PI, PI, LR, P0, Q) as well as the interrupt acknowledge sequence modify P0 in parts. That is, they alter P0 by first loading one part then the other and the entire operation takes more than one cycle. Should reset occur during this modification process the value pushed into P will be part of the old P0 (the as yet unmodified part) and part of the new P0 (already modified part). Thus, care should be taken (perhaps by external gating) to insure that reset does not occur at an undesirable time if any significance is to be given to the contents of P after a reset occurs.

V_{CC} DECOUPLING

The MC3870 family devices have dynamic circuitry internally which requires a good high frequency decoupling capacitor to surpress noise on the V_{CC} line. A 0.01 μF or 0.1 μF ceramic capacitor should be placed between V_{CC} and ground, located physically close to the MC3870 device. This will reduce noise generated by the MC3870 to about 70-100 mV on the V_{CC} line.

TEST LOGIC

Special test logic is implemented to allow access to the internal main data bus for test purposes.

In normal operation, the TEST pin is unconnected or is connected to GND. When TEST is placed at a TTL level (2.0 V to 2.6 V) port 4 becomes an output of the internal data

bus. The data appearing on the port 4 pins is logically true whereas input data forced on port 5 must be logically false. When TEST is placed at a high level (6.0 V to 7.0 V), the ports act as above and additionally the 2K × 8 program ROM is prevented from driving the data bus. In this mode, operands and instructions may be forced externally through port 5 instead of being accessed from the program ROM. When TEST is in either the TTL state or the high state, STROBE ceases its normal function and becomes a machine cycle clock (identical to the F8 multi-chip system WRITE clock except inverted).

Timing complexities render the capabilities associated with the TEST pin impractical for use in a user's application, but these capabilities are thoroughly sufficient to provide a rapid method for thoroughly testing the MC3870.

SUPPLEMENTARY NOTES

The Interrupt Control Bit of the W Status Register is automatically reset when an interrupt request is acknowledged. It is then the programmer's responsibility to determine when ICB will again be set (by execution an EI instruction). This action prevents an interrupt service routine from being interrupted unless the programmer so desires.

When reading the Interrupt Control Port (port 6), bit 7 of the Accumulator is loaded with the actual logic level being applied to the EXT INT pin, regardless of the status of ICP bit 2 (the EXT INT Active Level bit). This is, if EXT INT is at +5 V, bit 7 of the Accumulator is set to a logic "1"; but, if EXT INT is at GND, then Accumulator bit 7 is reset to logic "0".

In the MC3870 (F8 COMPATIBLE) INSTRUCTION SET summary, the number of cycles shown are "nominal" machine cycles. A nominal machine cycle is defined as 4 ϕ clock periods, thus, requiring 2 μs for a 2 MHz ϕ clock frequency (4 MHz external time-base frequency).

Also, the summary uses an older nomenclature for register names. The translation is as follows:

PC0 = P0	Program Counter
PC1 = P	Stack Register
DC0 = DC	Data Counter
DC1 = DC1	Auxiliary Data Counter

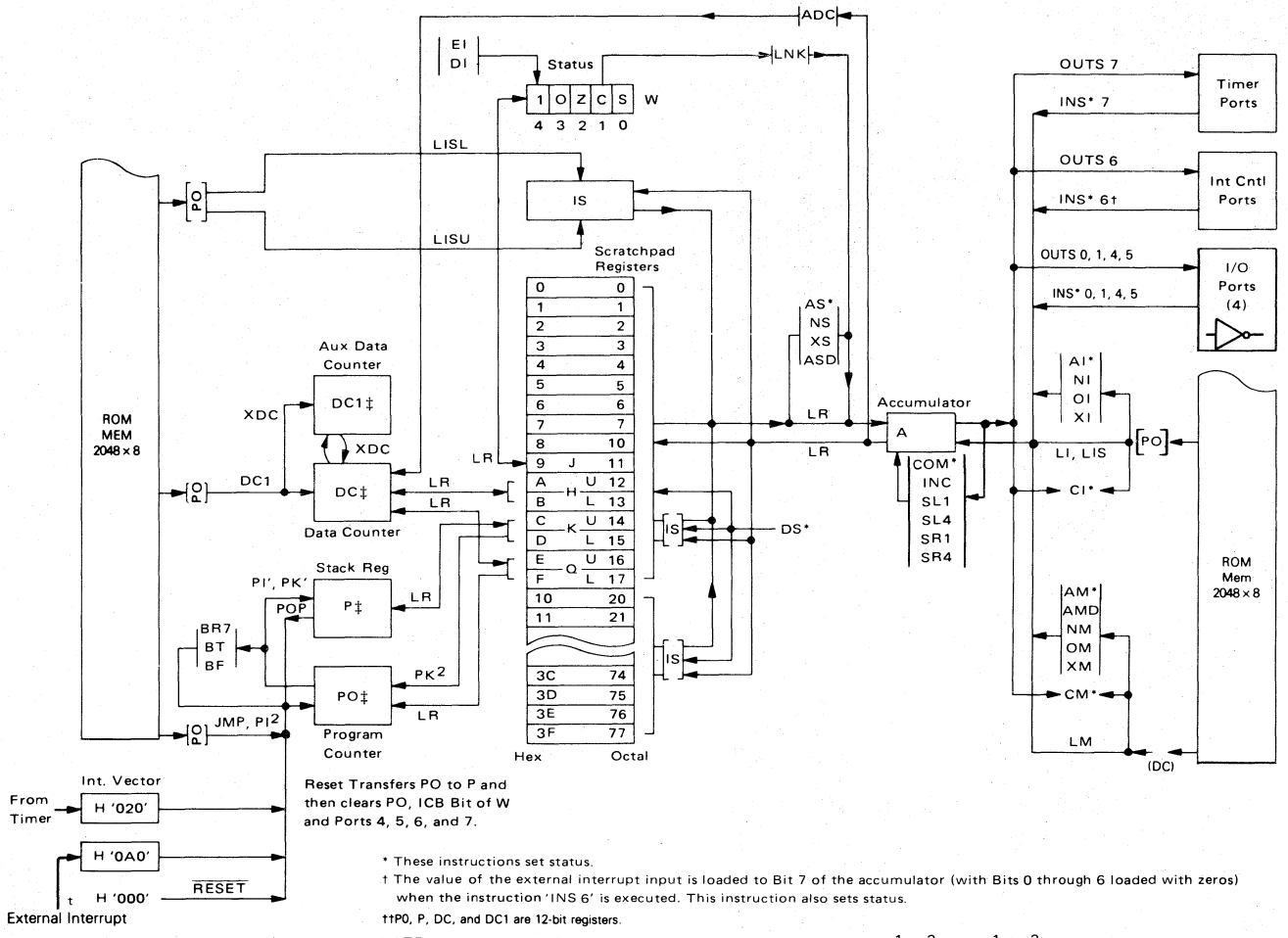
The nomenclature is used in order to be consistent with the assembly language mnemonics.

For the MC3870, execution of an INS or OUTS instruction requires 2 machine cycles for ports 0 and 1, whereas ports 4 and 5 require 4 machine cycles.

INSTRUCTION EXECUTION

This section details the timing and execution of the MC3870 instruction set. Refer to Figure 19 for a MC3870 Programming Model.

FIGURE 19 — MC3870 PROGRAMMING MODEL



* These instructions set status.

† The value of the external interrupt input is loaded to Bit 7 of the accumulator (with Bits 0 through 6 loaded with zeros) when the instruction 'INS 6' is executed. This instruction also sets status.

††P0, P, DC, and DC1 are 12-bit registers.

NOTE: The Instructions PI and PK are shown in two sequential parts. (PI^1, PI^2 and PK^1, PK^2).

MC3870 INSTRUCTION SET

ACCUMULATOR GROUP INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHzϕ)	Status Bits OVR	ZERO	CRY	SIGN
Add Carry	LNK		A←(A) + CRY	19	1	1		2	1/0	1/0	1/0	1/0
Add Immediate	AI	ii	A←(A) + H'ii'	24ii	2	1	1	5	1/0	1/0	1/0	1/0
And Immediate	NI	ii	A←(A)∧H'ii'	21ii	2	1	1	5	0	1/0	0	1/0
Clear	CLR		A←H'00'	70	1	1		2	–	–	–	–
Compare Immediate	CI	ii	H'ii' + (A) + 1	25ii	2	1	1	5	1/0	1/0	1/0	1/0
Complement	COM		A←(A) + H'FF'	18	1	1		2	0	1/0	0	1/0
Exclusive or Immediate	XI	ii	A←(A) + H'ii'	23ii	2	1	1	5	0	1/0	0	1/0
Increment	INC		A←(A) + 1	1F	1	1		2	1/0	1/0	1/0	1/0
Load Immediate	LI	ii	A←H'ii'	20ii	2	1	1	5	–	–	–	–
Load Immediate Short	LIS	i	A←H'0i'	7i	1	1		2				
OR Immediate	OI	ii	A←(A)∨H'ii'	22ii	2	1	1	5	0	1/0	0	1/0
Shift Left One	SL	1	Shift Left 1	13	1	1		2	0	1/0	0	1/0
Shift Left Four	SL	4	Shift Left 4	15	1	1		2	0	1/0	0	1/0
Shift Right One	SR	1	Shift Right 1	12	1	1		2	0	1/0	0	1
Shift Right Four	SR	4	Shift Right 4	14	1	1		2	0	1/0	0	1

BRANCH INSTRUCTIONS In all conditional branches P0←(P0) + 2 if the test condition is not met. Execution is complete in 3 short cycles.

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz ϕ)	Status Bits OVR	ZERO	CRY	SIGN
Branch on Carry	BC	aa	P0←(P0) + 1 + H'aa' if CRY = 1	82aa	2	2	1	7	–	–	–	–
Branch on Positive	BP	aa	P0←(P0) + 1 + H'aa' if SIGN = 1	81aa	2	2	1	7	–	–	–	–
Branch on Zero	BZ	aa	P0←(P0) + 1 + 'Haa' if Zero = 1	84aa	2	2	1	7	–	–	–	–
Branch on True	BT	taa	P0←(P0) + 1 + 'Haa' if any test is true	8taa	2	2	1	7	–	–	–	–
	TEST CONDITION											
	2^2 | 2 | 2^0											
	ZERO | CRY | SIGN											
Branch if Negative	BM	aa	P0←(P0) + 1 + H'aa' if SIGN = 0	91aa	2	2	1	7	–	–	–	–
Branch if No Carry	BNC	aa	P0←(P0) + 1 + H'aa' if CARRY = 0	92aa	2	2	1	7	–	–	–	–
Branch if No Overflow	BNO	aa	P0←(P0) + 1 + H'aa' if OVR = 0	98aa	2	2	1	7	–	–	–	–
Branch if Not Zero	BNZ	aa	P0←(P0) + 1 + H'aa' if ZERO = 0	94aa	2	2	1	7	–	–	–	–
Branch if False Test	BF	taa	P0←(P0) + 1 + H'aa' if all false test bits	9taa	2	2	1	7	–	–	–	–
	TEST CONDITION											
	2^7 | 2^6 | 2^5 | 2^4											
	OVF | ZERO | CRY | SIGN											
Branch if ISAR (Lower) ≠ 7	BR7	aa	P0←(P0) + 1 + H'aa' ISARL ≠ 7	8Faa	2	2	1	5	–	–	–	–
			P0←(P0) + 2 if ISARL = 7		2	2		4	–	–	–	–
Branch Relative	BR	aa	P0←(P0) + 1 + H'aa'	90aa	2	2	1	7	–	–	–	–
Jump*	JMP	aaaa	P0←H'aaaa'	29aaaa	3	1	3	11	–	–	–	–

*Privileged instruction, accumulator contents altered during execution JMP.

MEMORY REFERENCE INSTRUCTIONS In all Memory Reference Instructions, the Data Counter is incremented DC←(DC)+1.

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add Binary	AM		A←(A)+[(DC)]	88	1	1	1	5	1/0	1/0	1/0	1/0
Add Decimal	AMD		A←(A)+[(DC)]• BCD Adjust	89	1	1	1	5	1/0	1/0	1/0	1/0
AND	NM		A←(A)Λ[(DC)]	8A	1	1	1	5	0	1/0	0	1/0
Compare	CM		[(DC)]+(A)+1	8D	1	1	1	5	1/0	1/0	1/0	1/0
Exclusive OR	XM		A←(A)⊕[(DC)]	8C	1	1	1	5	0	1/0	0	1/0
Load	LM		A←[(DC)]	16	1	1	1	5	–	–	–	–
Logical OR	OM		A←(A)v[(DC)]	8B	1	1	1	5	0	1/0	0	1/0
Store	ST		A←[(DC)]	17	1	1	1	5	–	–	–	–

ADDRESS REGISTER GROUP INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add to Data Counter	ADC		DC←(DC)+(A)	8E	1	1	1	5	–	–	–	–
Call to Subroutine*	PK		P0U←(r12); P0L←(r13), P←(P0)	0C	1	1	2	8	–	–	–	–
Call to Subroutine Immediate*	PI	aaaa	P←(P0), P0←H'aaaa	28aaaa	3	2	3	13	–	–	–	–
Exchange DC	XDC		(DC)⇄(DC1)	2C	1	2		4	–	–	–	–
Load Data Counter	LR	DC, Q	DCU←(r14), DCL←(r15)	OF	1	1	2	8	–	–	–	–
Load Data Counter	LR	DC, H	DCU←(r10), DCL←(r11)	10	1	1	2	8	–	–	–	–
Load DC Immediate	DCI	aaaa	DC H'aaaa'	2Aaaaa	3	3	2	12	–	–	–	–
Load Program Counter	LR	P0, Q	P0U ←(r14), P0L ←(r15)	OD	1	1	2	8	–	–	–	–
Load Stack Register	LR	P, K	PU ←(r12), PL←r13)	09	1	1	2	8	–	–	–	–
Return from Subroutine*	P0P		P0 ←(P)	1C	1	2		4	–	–	–	–
Store Data Counter	LR	Q, DC	r14←(DCU), r15←(DCL)	OE	1	1	2	8	–	–	–	–
Store Data Counter	LR	H, DC	r10← DCU, r11←(DCL)	11	1	1	2	8	–	–	–	–
Store Stack Register	LR	K, P	r12←(PU), r13←(PL)	08	1	1	2	8	–	–	–	–

SCRATCHPAD REGISTER INSTRUCTIONS (Refer to Scratchpad Addressing Modes)

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz φ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Add Binary	AS	r	A←(A)+(r)	Cr	1	1		2	1/0	1/0	1/0	1/0
Add Decimal	ASD	r	A←(A)+(r)	Dr	1	2		4	1/0	1/0	1/0	1/0
Decrement	DS	r	r←(r)+H'FF'	3r	1		1	3	1/0	1/0	1/0	1/0
Load	LR	A, r	A←(r)	4r	1	1		2	–	–	–	–
Load	LR	A, KU	A←(r12)	00	1	1		2	–	–	–	–
Load	LR	A, KL	A←(r13)	01	1	1		2	–	–	–	–
Load	LR	A, QU	A←(r14)	02	1	1		2	–	–	–	–
Load	LR	A, QL	A←(r15)	03	1	1		2	–	–	–	–
Load	LR	r, A	r←(A)	5r	1	1		2	–	–	–	–
Load	LR	KU, A	r12←(A)	04	1	1		2	–	–	–	–
Load	LR	KL, A	r13←(A)	05	1	1		2	–	–	–	–
Load	LR	QU, A	r14←(A)	06	1	1		2	–	–	–	–
Load	LR	QL, A	r15←(A)	07	1	1		2	–	–	–	–
AND	NS	r	A←(A)Λ(r)	Fr	1	1		2	0	1/0	0	1/0
Exclusive OR	XS	r	A←(A)+(r)	Er	1	1		2	0	1/0	0	1/0

*Privileged instruction, accumulator contents altered during execution of PI instruction.

MISCELLANEOUS INSTRUCTIONS

Operation	Mnemonic Op Code	Operand	Function	Machine Code	Bytes	Cycles Short	Cycles Long	(2 MHz ϕ)	Status Bits OVR	Status Bits ZERO	Status Bits CRY	Status Bits SIGN
Disable Interrupt	DI		Reset ICB	1A	1	1		2	–	–	–	–
Enable Interrupt*	EI		Set ICB	1B	1	1		2	–	–	–	–
Input	IN	04,05,06,07	A ← (Input Port aa)	26aa	2	1	2	8	0	1/0	0	1/0
Input Short	INS	0, 1	A ← (Input Port 0 or 1)	A0,A1	1	2		4	0	1/0	0	1/0
Input Short	INS	4,5,6,7	A ← (Input Port a)	Aa	1	1	2	8	0	1/0	0	1/0
Load ISAR	LR	IS,A	IS ← (A)	0B	1	1		2	–	–	–	–
Load ISAR Lower	LISL	bbb	ISL ← bbb	6(1bbb)**	1	1		2	–	–	–	–
Load ISAR Upper	LISU	bbb	ISU ← bbb	6(0bbb)**	1	1		2	–	–	–	–
Load Status Register*	LR	W,J	W ← (r9)	1D	1	2		4	1/0	1/0	1/0	1/0
No Operation	NOP		P0 ← (P0) + 1	2B	1	1		2	–	–	–	–
Output*	OUT	04,05,06,07	Output Port aa ← (A)	27aa	2	1	2	8	–	–	–	–
Output Short	OUTS	0, 1	Output Port 0 or 1 ← (A)	B0,B1	1	2		4	–	–	–	–
Output Short	OUTS	4,5,6,7	Output Port a ← (A)	Ba	1	1	2	8	–	–	–	–
Store ISAR	LR	A,IS	A ← (IS)	0A	1	1		2	–	–	–	–
Store Status Reg	LR	J,W	r9 ← (W)	1E	1	1		2	–	–	–	–

*Privileged instruction
**b = 1-bit immediate operand

3

NOTES

Lower case denotes variables specified by programmer

Function	Definitions
←	is replaced by
( )	the contents of
( )	Binary "1s" complement of
+	Arithmetic Add (Binary or Decimal)
⊕	Logical "OR" exclusive
Λ	Logical "AND"
v	Logical "OR" inclusive
H"	Hexadecimal digit
[( )]	Contents of memory specified by ( )
a	Address Variable (four bits)
A	Accumulator
b	One bit immediate operand
DC	Data Counter (Indirect Address Register)
DC1	Data Counter 1 (Auxiliary Data Counter)
DCL	Least significant 8 bits of Data Counter Addressed
DCU	Most significant 8 bits of Data Counter Addressed
H	Scratchpad Register 10 and 11
i	Immediate operand (four bits)
ICB	Interrupt Control Bit
IS	Indirect Scratchpad Address Register
ISL	Least Significant 3 bits of ISAR
ISU	Most Significant 3 bits of ISAR
J	Scratchpad Register 9
K	Registers 12 and 13
KL	Register 13
KU	Register 12
P0	Program Counter
P0L	Least Significant 8 Bits of Program Counter
P0U	Most Significant 8 bits of Program Counter
P	Stack Register
PL	Least Significant 8 bits of Program Counter
PU	Most Significant 8 bits of Active Stack Register
Q	Registers 14 and 15
QL	Reigster 15
QU	Register 14
r	Scratchpad Register (any address 0 through B) (See Below)
W	Status Register

Scratchpad Addressing Modes Using IS. (r≠0 through B)

r = H'C'	Register Addressed by IS is (Unmodified)
r = H'D'	Register Addressed by IS is Incremented
r = H'E'	Register Addressed by IS is Decremented
r = H'F'	Illegal OP Code

Status Register

–	No change in condition
1/0	is set to "1" or "0" depending on conditions
CRY	Carry Flag
OVR	Overflow Flag
SIGN	Sign of Result Flag
ZERO	Zero Flag

ORDERING INFORMATION

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

PROM(s) MCM2716s or MCM2708s

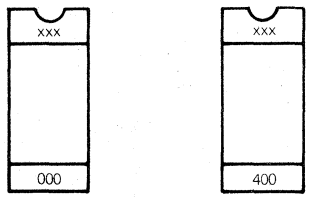
MDOS disk file

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.

PROMs — The MCM2708 or MCM2716 type PROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The PROMs must be clearly marked to indicate which PROM corresponds to which address space (000-3FF HEX), (400-7FF) or (000-7FF). See Figure 21 for recommended marking procedure.

After the PROM(s) are marked they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE 21 — PROM MARKING



xxx = Customer ID

VERIFICATION MEDIA

All original pattern media (PROMs 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 EPROM (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATION UNITS

Ten MC3870s 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 5 volts. These RVUs are included in the mask charge and are not production parts.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The floppies are not to be returned by Motorola as they are used for archival storage. The minimum MDOS system files must be on the disk as well as the absolute binary object file (filename .LO type of file) from the MC3870 cross assembler. An object file made from a memory dump using the ROLLOUT command is also admissable. Consider submitting a source listing as well as the following files: filename .LX (EXORciser® loadable format) and filename .SA (ASCII Source Code). These files will of course be kept confidential and are 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 representatives.

MDOS is Motorola's Disk Operating System available on development systems such as EXORcisers, or EXORsets, etc.

MC3870 ORDERING INFORMATION

Date _____ Customer PO Number _____

Customer Name _____

Address _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Contact _____

Customer Part Number _____

Options:

	Reset	Pullup ☐	No Pullup ☐
	External Interrupt	Pullup ☐	No Pullup ☐

Port Options:

	Standard TTL	Open Drain	Direct Drive
P4-0	☐	☐	☐
P4-1	☐	☐	☐
P4-2	☐	☐	☐
P4-3	☐	☐	☐
P4-4	☐	☐	☐
P4-5	☐	☐	☐
P4-6	☐	☐	☐
P4-7	☐	☐	☐
P5-0	☐	☐	☐
P5-1	☐	☐	☐
P5-2	☐	☐	☐
P5-3	☐	☐	☐
P5-4	☐	☐	☐
P5-5	☐	☐	☐
P5-6	☐	☐	☐
P5-7	☐	☐	☐

Pattern Media

☐ PROMs (MCM2716 or MCM2708)
(Customer can send in two extra PROMs, Motorola will program the customer's code on these PROMs for code verification.

☐ Floppy Disk

☐ Other _____

Clock Mode _____ ☐ XTAL ☐ RC ☐ LC ☐ External

Clock Freq _____

Temp Range _____ ☐ 0-70°C ☐ −40-+85°C

Marking Information (12 Characters Maximum)

_____

NOTE: All other media requires prior factory approval.

MOTOROLA

MC6801
MC6803

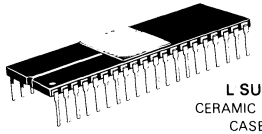
MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

The MC6801 is an 8-bit single-chip microcomputer unit (MCU) which significantly enhances the capabilities of the M6800 Family of parts. It includes an upgraded M6800 microprocessor unit (MPU) with upward-source and object-code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one +5-volt power supply. On-chip resources include 2048 bytes of ROM, 128 bytes of RAM, a serial communications interface (SCI), parallel I/O, and a three-function programmable timer. The MC6803 can be considered as an MC6801 operating in modes 2 or 3. An EPROM version of the MC6801, the MC68701 microcomputer, is available for systems development. The MC68701 is pin and code compatible with the MC6801/03 and can be used to emulate the MC6801/03. The MC68701 is described in a separate Advance Information publication.

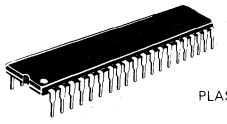
- Enhanced MC6800 Instruction Set
- 8 × 8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatibility with the M6800
- 16-Bit Three-Function Programmable Timer
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Bus Compatibility with the M6800 Family
- 2048 Bytes of ROM (MC6801 Only)
- 128 Bytes of RAM
- 64 Bytes of RAM Retainable During Powerdown
- 29 Parallel I/O and Two Handshake Control Lines
- Internal Clock Generator with Divide-by-Four Output
- −40 to 85°C Temperature Range
- −40 to 105°C Temperature Range

MOS
(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

MICROCOMPUTER MICROPROCESSOR

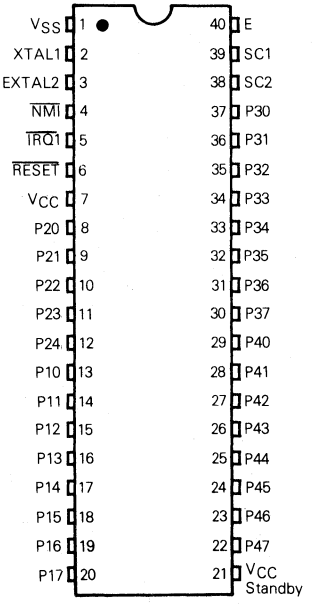


L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

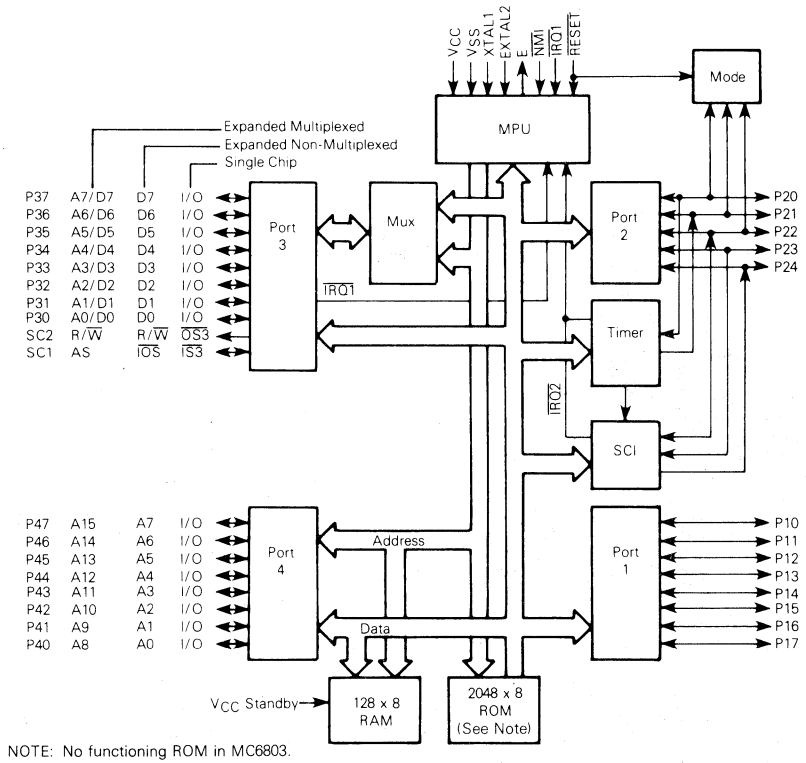
PIN ASSIGNMENT



GENERIC INFORMATION

Frequency MHz	Temperature	Generic Number: Ceramic Package L Suffix	Generic Number: Plastic Package P Suffix
1.0	0°C to 70°C	MC6801L1	MC6801P1
1.0	−40°C to 85°C	MC6801CL1	MC6801CP1
1.0	−40°C to 105°C	MC6801VL1	MC6801VP1
1.0	0°C to 70°C	MC6803L	MC6803P
1.0	−40°C to 85°C	MC6803CL	MC6803CP
1.0	−40°C to 105°C	MC6803VL	MC6803VP
1.25	0°C to 70°C	MC6801L1-1	MC6803P1-1
1.25	−40°C to 85°C	MC6801CL1-1	MC6803CP1-1
1.25	0°C to 70°C	MC6803L1	MC6803P1
1.25	−40°C to 85°C	MC6803CL-1	MC6803CP-1
1.5	0°C to 70°C	MC68A01L1	MC68A01P1
1.5	0°C to 70°C	MC68A03L	MC68A03P
2.0	0°C to 70°C	MC68B01L1	MC68B01P1
2.0	0°C to 70°C	MC68B03L	MC68B03P

FIGURE 1 — M6801 MICROCOMPUTER FAMILY BLOCK DIAGRAM



NOTE: No functioning ROM in MC6803.

POWER CONSIDERATIONS

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

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

Where:

T_A ≡ Ambient Temperature, °C

θ_{JA} ≡ Package Thermal Resistance, Junction-to-Ambient, °C/W

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

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

P_{PORT} ≡ Port Power Dissipation, Watts — User Determined

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

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

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

Solving equations 1 and 2 for K gives:

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

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

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6801, MC6803 MC6801C, MC6803C MC6801V, MC6803V	T_A	T_L to T_H 0 to 70 −40 to 85 −40 to 105	°C
Storage Temperature Range	T_{stg}	−55 to + 150	°C

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

THERMAL CHARACTERISTICS

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

3

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

Characteristic	Symbol	MC6801		MC6801-1		MC68A01		MC68B01		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	0.5	1.5	0.5	2.0	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
External Oscillator Frequency	$4f_o$	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	–	100	–	100	–	100	–	100	ms
Processor Control Setup Time	t_{PCS}	200	–	170	–	140	–	110	–	ns

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

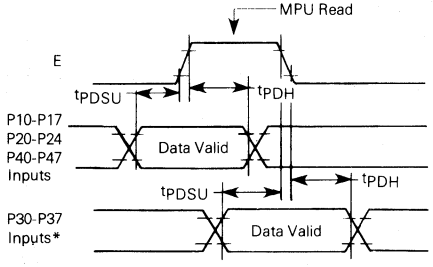
Characteristic		Symbol	MC6801 MC6803		MC6801C MC6803C		MC6801V MC6803V		Unit
			Min	Max	Min	Max	Min	Max	
Input High Voltage	$\overline{RESET}$	V_{IH}	V_{SS}+4.0	V_{CC}	V_{SS}+4.0	V_{CC}	V_{SS}+4.0	V_{CC}	V
	Other Inputs		V_{SS}+2.0	V_{CC}	V_{SS}+2.2	V_{CC}	V_{SS}+2.2	V_{CC}	
Input Low Voltage	All Inputs	V_{IL}	V_{SS}−0.3	V_{SS}+0.8	V_{SS}−0.3	V_{SS}+0.8	V_{SS}−0.3	V_{SS}+0.7	V
Input Load Current	Port 4	I_{in}	–	0.5	–	0.8	–	0.8	mA
(V_{in} = 0 to 2.4 V)	SCI		–	0.8	–	1.0	–	1.0	
Input Leakage Current (V_{in} = 0 to 5.25 V)	$\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	2.5	–	5.0	–	5.0	μA
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	Ports 1, 2, and 3	I_{TSI}	–	10	–	20	–	20	μA
Output High Voltage									
(I_{Load} = −65 μA, V_{CC} = Min)*	Port 4, SC1, SC2	V_{OH}	V_{SS}+2.4	–	V_{SS}+2.4	–	V_{SS}+2.4	–	V
(I_{Load} = −100 μA, V_{CC} = Min)	Other Outputs		V_{SS}+2.4	–	V_{SS}+2.4	–	V_{SS}+2.4	–	
Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = Min)	All Outputs	V_{OL}	–	V_{SS}+0.5	–	V_{SS}+0.6	–	V_{SS}+0.6	V
Darlington Drive Current (V_O = 1.5 V)	Port 1	I_{OH}	1.0	4.0	1.0	5.0	1.0	5.0	mA
Internal Power Dissipation (Measured at $T_A = T_L$ in Steady-State Operation)		P_{INT}	–	1200	–	1500	–	1500	mW
Input Capacitance	Port 3, Port 4, SCI	C_{in}	–	12.5	–	12.5	–	12.5	pF
(V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	Other Inputs		–	10	–	10	–	10	
V_{CC} Standby	Powerdown	V_{SBB}	4.0	5.25	4.0	5.25	4.0	5.25	V
	Powerup	V_{SB}	4.75	5.25	4.75	5.25	4.75	5.25	
Standby Current	Powerdown	I_{SBB}	–	6.0	–	8.0	–	8.0	mA

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

PERIPHERAL PORT TIMING (Refer to Figures 2-5)

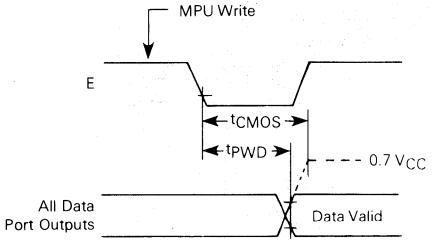
Characteristics	Symbol	MC6801 MC6803		MC6801-1 MC6803-1		MC68A01 MC68A03		MC68B01 MC68B03		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Peripheral Data Setup Time	t_{PDSU}	200	–	200	–	150	–	100	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	200	–	150	–	100	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	t_{PWD}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	–	2.0	–	2.0	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	200	–	150	–	100	–	ns
Input Data Hold Time	t_{IH}	50	–	50	–	40	–	30	–	ns
Input Data Setup Time	t_{IS}	20	–	20	–	20	–	20	–	ns

FIGURE 2 — DATA SETUP AND HOLD TIMES (MPU READ)



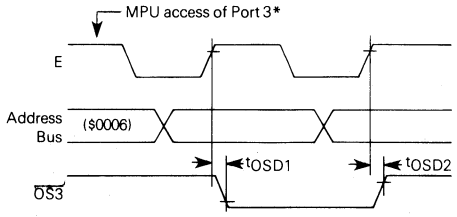
*Port 3 non-latched operation (LATCH ENABLE = 0)

FIGURE 3 — DATA SETUP AND HOLD TIMES (MPU WRITE)



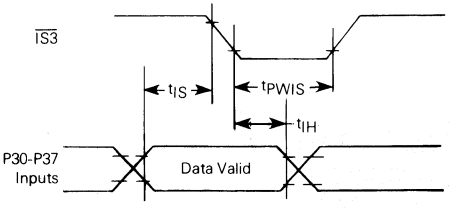
NOTES:
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}.
2. Not applicable to P21.
3. Port 4 cannot be pulled above V_{CC}.

FIGURE 4 — PORT 3 OUTPUT STROBE TIMING (MC6801 SINGLE-CHIP MODE)



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

FIGURE 5 — PORT 3 LATCH TIMING (MC6801 SINGLE-CHIP MODE)



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

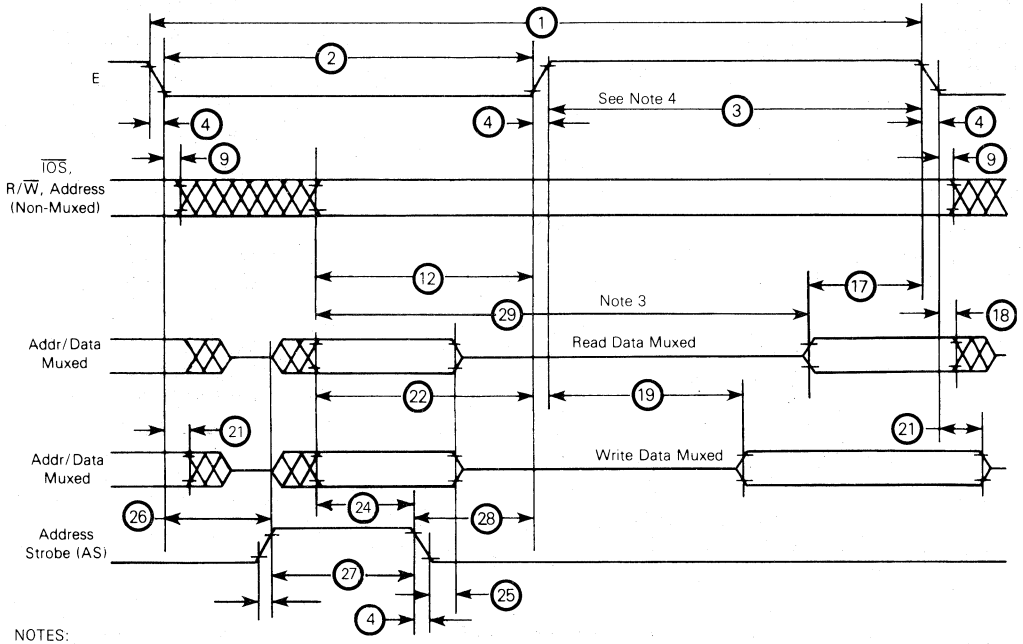
BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6801 MC6803		MC6801-1 MC6803-1		MC68A01 MC68A03		MC68B01 MC68B03		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	0.667	2.0	0.5	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	300	1000	210	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	300	1000	220	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	10	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	115	–	70	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	60	–	40	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	–	170	–	120	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	20	–	10	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	200	–	150	–	115	–	80	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	60	–	50	–	40	–	20	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	20	–	10	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	90**	–	70**	–	60**	–	45**	–	ns
27	Pulse Width, AS High*	PW_{ASH}	220	–	170	–	140	–	110	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	60	–	45	–	ns
29	Usable Access Time*	t_{ACC}	595	–	465	–	380	–	270	–	ns

*At specified cycle time.

**t_{ASD} parameters listed assume external TTL clock drive with 50% ±5% duty cycle. Devices driven by an external TTL clock with 50% ±1% duty cycle or which use a crystal have the following t_{ASD} specifications: 100 nanoseconds minimum (1.0 MHz devices), 80 nanoseconds minimum (1.25 MHz device), 65 nanoseconds minimum (1.5 MHz devices), 50 nanoseconds minimum (2.0 MHz devices).

FIGURE 6 — BUS TIMING



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

FIGURE 7 — CMOS LOAD

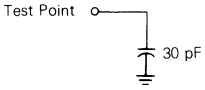
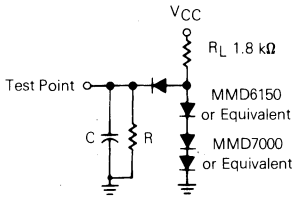


FIGURE 8 — TIMING TEST LOAD PORTS 1, 2, 3, 4



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

3

INTRODUCTION

The MC6801 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

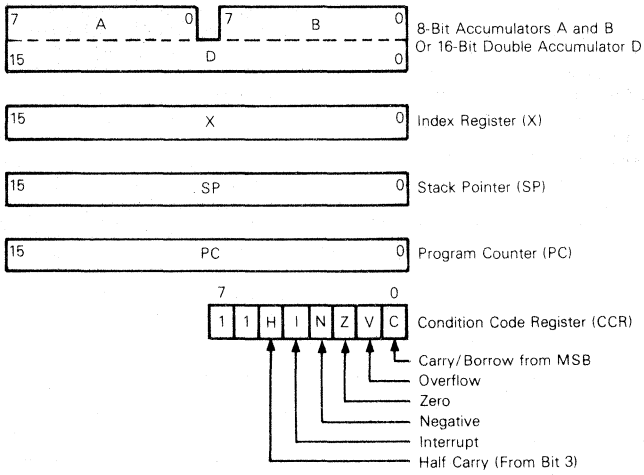
Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a data register and a write-only data direction register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port," by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

The microprocessor unit (MPU) is an enhanced MC6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800. The programming model is depicted in Figure 9, where accumulator D is a concatenation of accumulators A and B. A list of new operations added to the M6800 instruction set are shown in Table 1.

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

FIGURE 9 — PROGRAMMING MODEL



OPERATING MODES

The MC6801 provides eight different operating modes (0 through 7) and the MC6803 provides two operating modes (2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

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

MC6801 Single-Chip Modes (4, 7)

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

TABLE 1 — NEW INSTRUCTIONS

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

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

MC6801 Expanded Non-Multiplexed Mode (5)

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

Figure 12 illustrates a typical system configuration in the expanded non-multiplexed mode. The MCU interfaces directly with M6800 Family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as a memory-page select or chip-select line.

TABLE 2 — SUMMARY OF MC6801/03 OPERATING MODES

Common to all Modes: Reserved Register Area Port 1 Port 2 Programmable Timer Serial Communications Interface
Single Chip Mode 7 128 bytes of RAM; 2048 bytes of ROM Port 3 is a parallel I/O port with two control lines Port 4 is a parallel I/O port SC1 is Input Strobe 3 ($\overline{\text{IS3}}$) SC2 is Output Strobe 3 ($\overline{\text{OS3}}$)
Expanded Non-Multiplexed Mode 5 128 bytes of RAM; 2048 bytes of ROM 256 bytes of external memory space Port 3 is an 8-bit data bus Port 4 is an input port/address bus SC1 is Input/Output Select ($\overline{\text{IOS}}$) SC2 is Read/Write (R/$\overline{\text{W}}$)
Expanded Multiplexed Modes 1, 2, 3, 6* Four memory space options (64K address space): (1) No internal RAM or ROM (Mode 3) (2) Internal RAM, no ROM (Mode 2) (3) Internal RAM and ROM (Mode 1) (4) Internal RAM, ROM with partial address bus (Mode 6) Port 3 is a multiplexed address/data bus Port 4 is an address bus (inputs/address in Mode 6) SC1 is Address Strobe (AS) SC2 is Read/Write (R/$\overline{\text{W}}$)
Test Modes 0 and 4 Expanded Multiplexed Test Mode 0 May be used to test RAM and ROM Single Chip and Non-Multiplexed Test Mode 4 (1) May be changed to Mode 5 without going through Reset (2) May be used to test Ports 3 and 4 as I/O ports

*The MC6803 operates only in modes 2 and 3.

FIGURE 10 — SINGLE-CHIP MODE

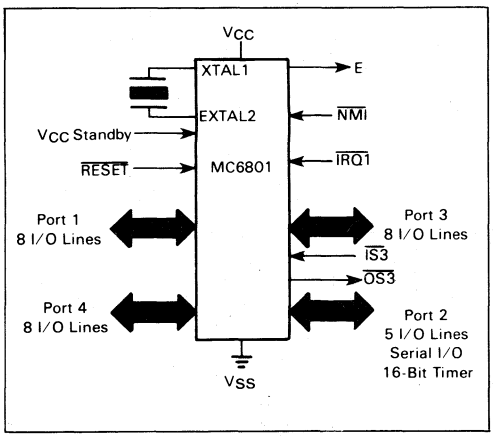


FIGURE 11 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION

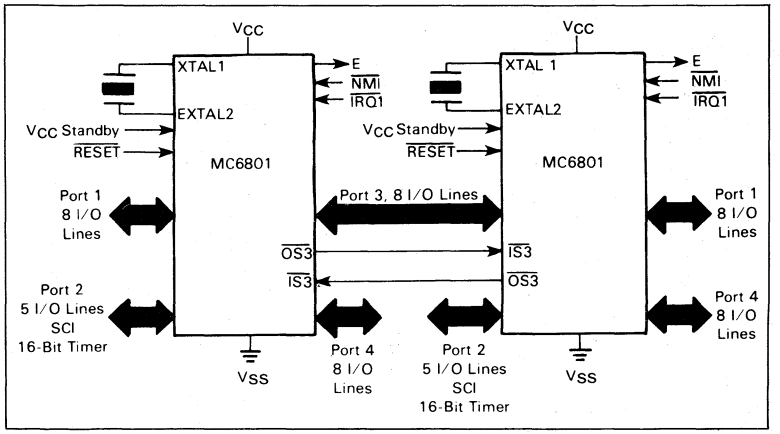
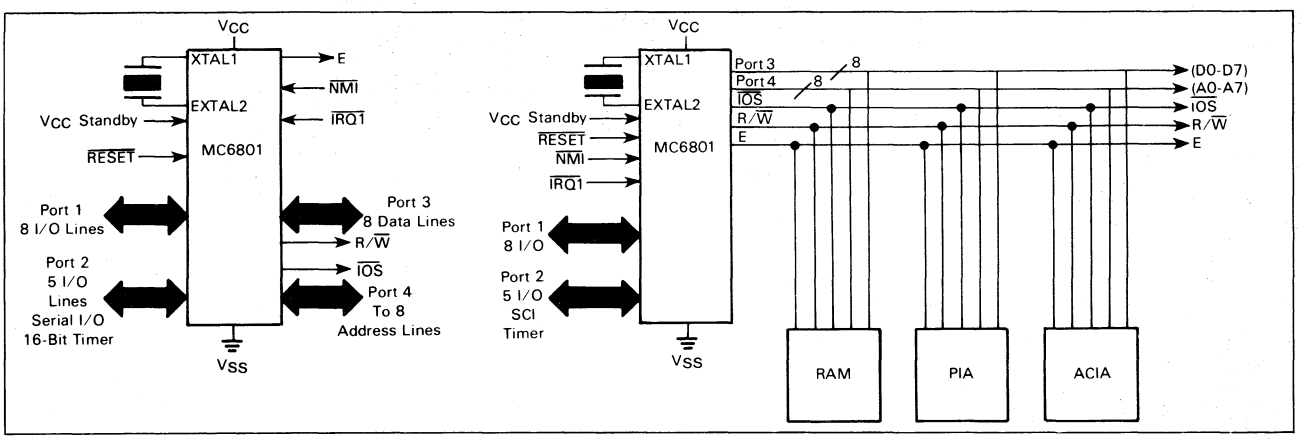


FIGURE 12 — EXPANDED NON-MULTIPLEXED CONFIGURATION



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

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

In mode 0, the reset vector is external for the first two E cycles after the positive edge of $\overline{\text{RESET}}$, and internal thereafter. In addition, the internal and external data buses are connected so there must be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

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

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

PROGRAMMING THE MODE

The operating mode is determined at $\overline{\text{RESET}}$ by the levels asserted on P22, P21, and P20. These levels are latched into PC2, PC1, and PC0 of the program control register on the positive edge of $\overline{\text{RESET}}$. The operating mode may be read from the port 2 data register as shown below, and programming levels and timing must be met as shown in Figure 15. A brief outline of the operating modes is shown in Table 3.

PORT 2 DATA REGISTER

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

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

TABLE 3 — MODE SELECTION SUMMARY

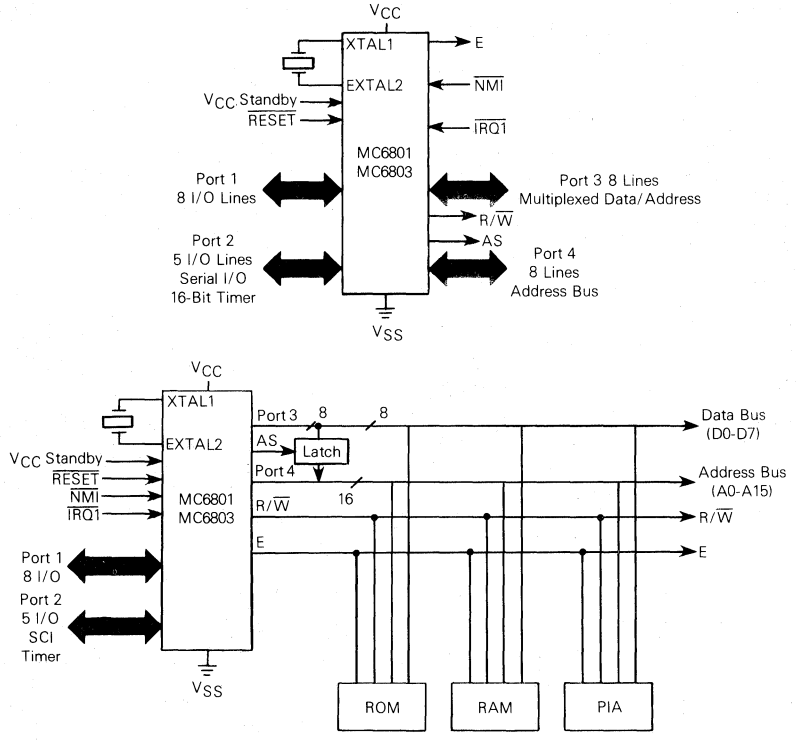
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	MUX(5, 6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single-Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed/No RAM or ROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and ROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed Test

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

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

* The MC6803 operates only in modes 2 and 3.

FIGURE 13 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 14 — TYPICAL LATCH ARRANGEMENT

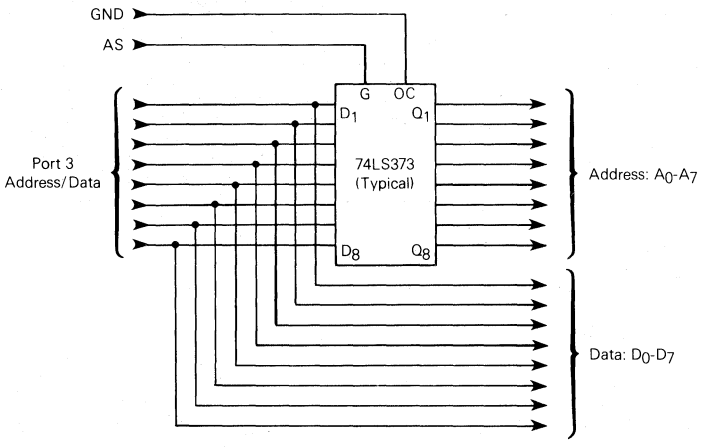
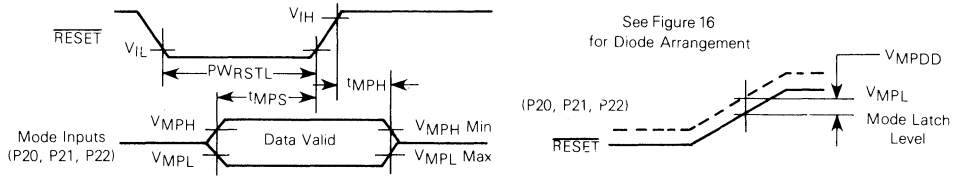


FIGURE 15 — MODE PROGRAMMING TIMING

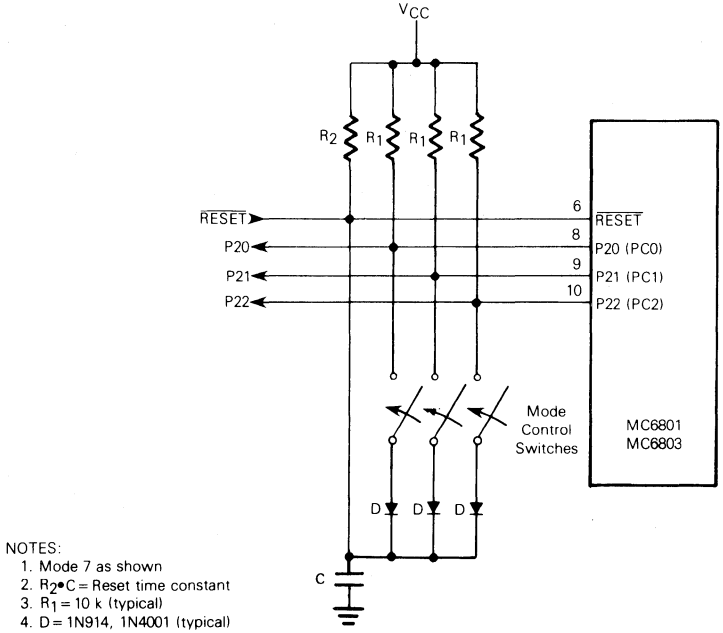


MODE PROGRAMMING (Refer to Figure 15)

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

*For $T_A = -40°C$ to 105°C, $V_{MPL} = 1.7$ V.

FIGURE 16 — TYPICAL MODE PROGRAMMING CIRCUIT

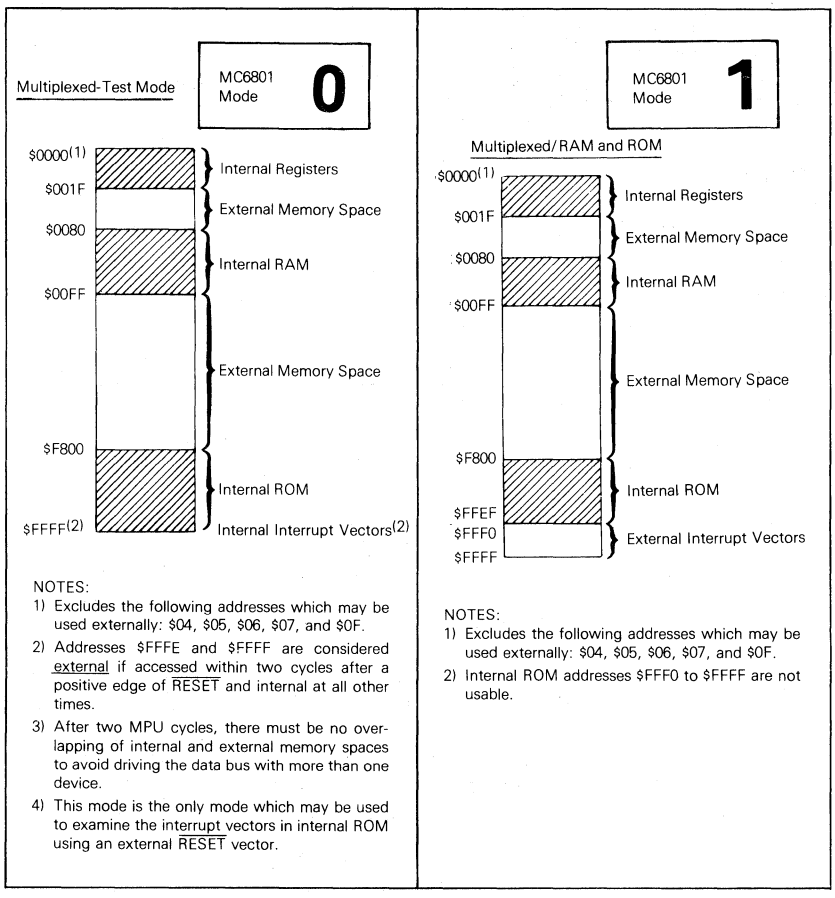


NOTES:
1. Mode 7 as shown
2. $R_2 \bullet C$ = Reset time constant
3. $R_1 = 10$ k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

MEMORY MAPS

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

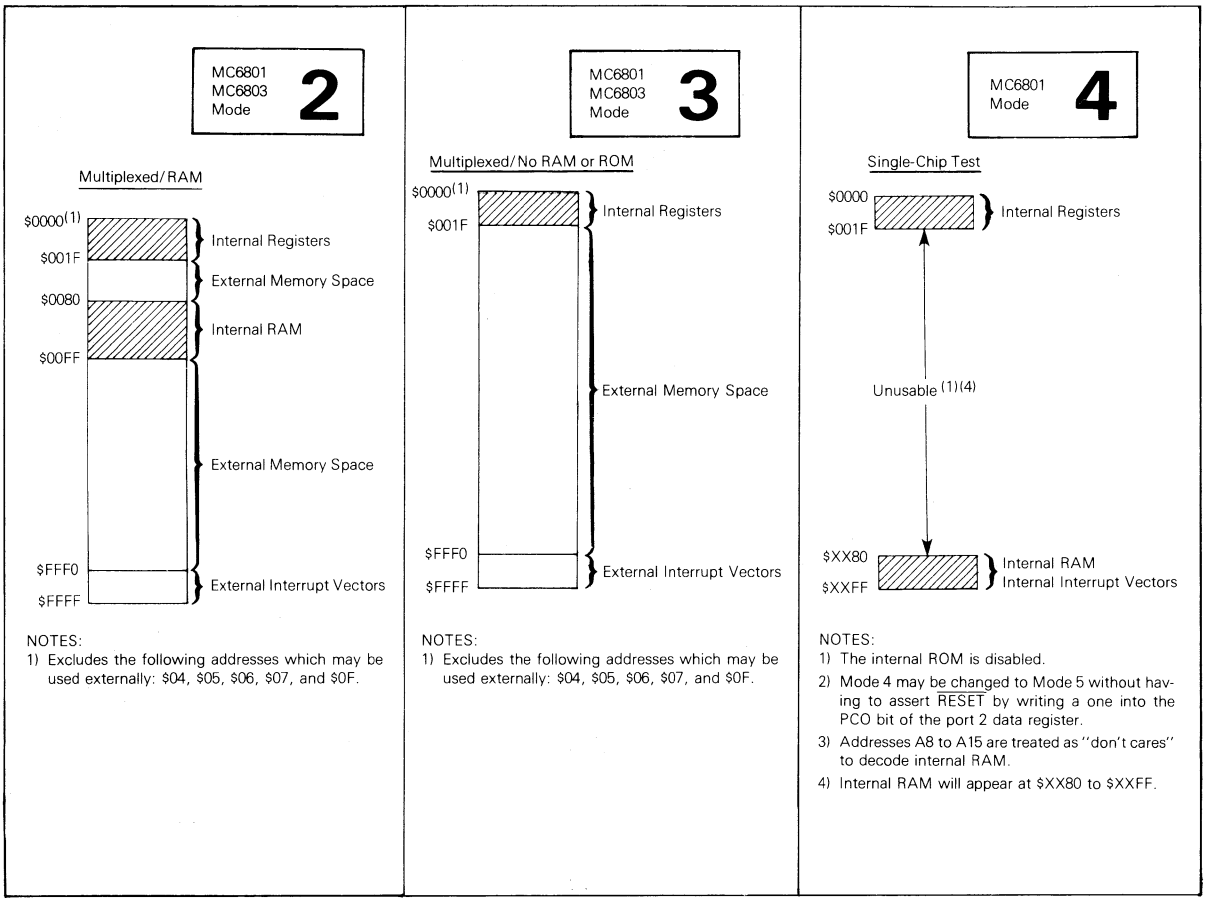
FIGURE 17 – MC6801/03 MEMORY MAPS (Sheet 1 of 3)



NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Addresses $FFFE and $FFFF are considered external if accessed within two cycles after a positive edge of $\overline{\text{RESET}}$ and internal at all other times.
3) After two MPU cycles, there must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.

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

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

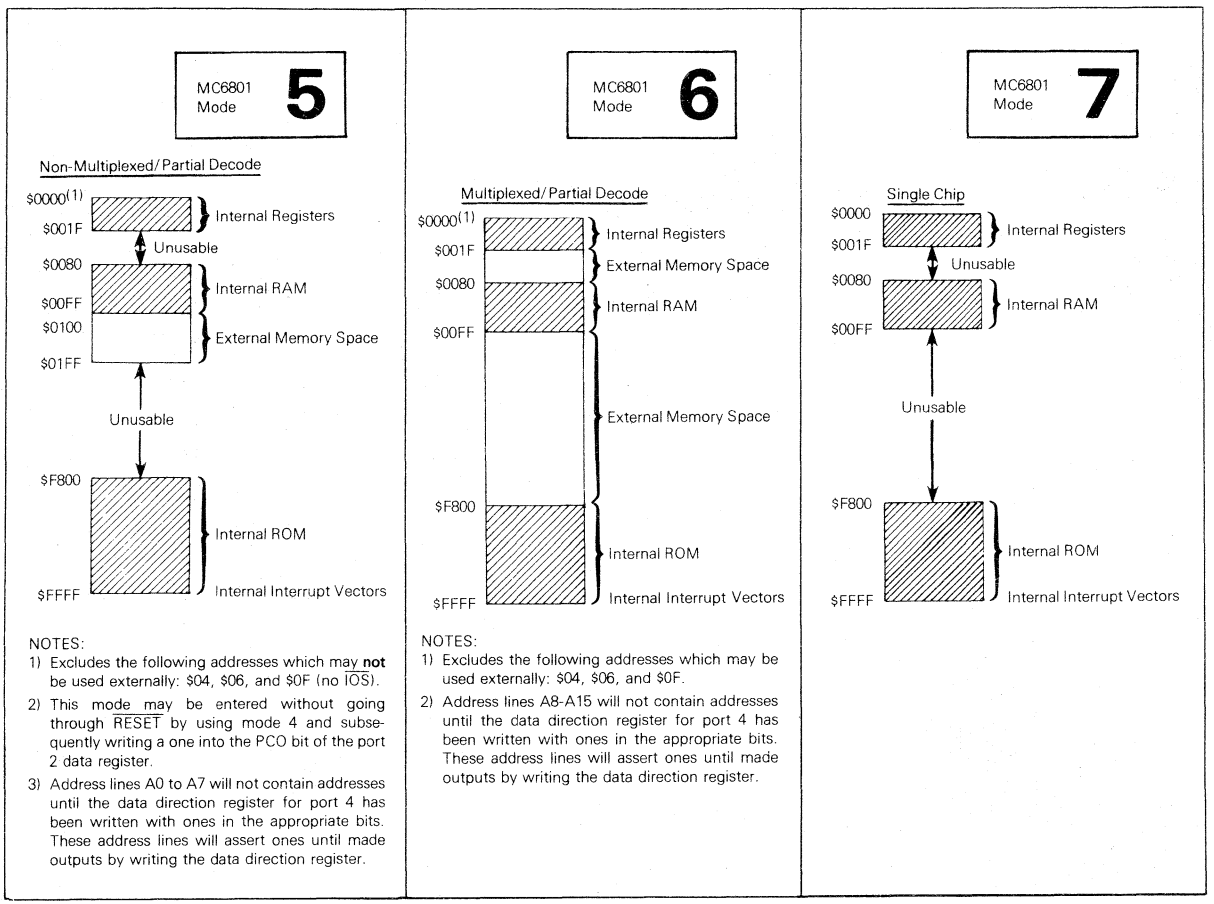


NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

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

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

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



MC6801/03 INTERRUPTS

The M6801 Family supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I bit and by individual enable bits. The I bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ1}$ and $\overline{IRQ2}$. The programmable timer and serial communications interface use an internal $\overline{IRQ2}$ interrupt line, as shown in Figure 1. External devices (and IS3) use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

All $\overline{IRQ2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in Table 5.

The interrupt flowchart is depicted in Figure 18 and is common to every interrupt excluding reset. During interrupt servicing the program counter, index register, A accumulator, B accumulator, and condition code register are pushed to the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 19 and 20.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

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

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

TABLE 4 — INTERNAL REGISTER AREA

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

*External addresses in modes 0, 1, 2, 3, 5, and 6; cannot be accessed in mode 5 (no $\overline{OS}$).
**External addresses in modes 0, 1, 2, and 3.
***1 = Output, 0 = Input.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

*$\overline{IRQ2}$ Interrupt

FIGURE 18 — INTERRUPT FLOWCHART

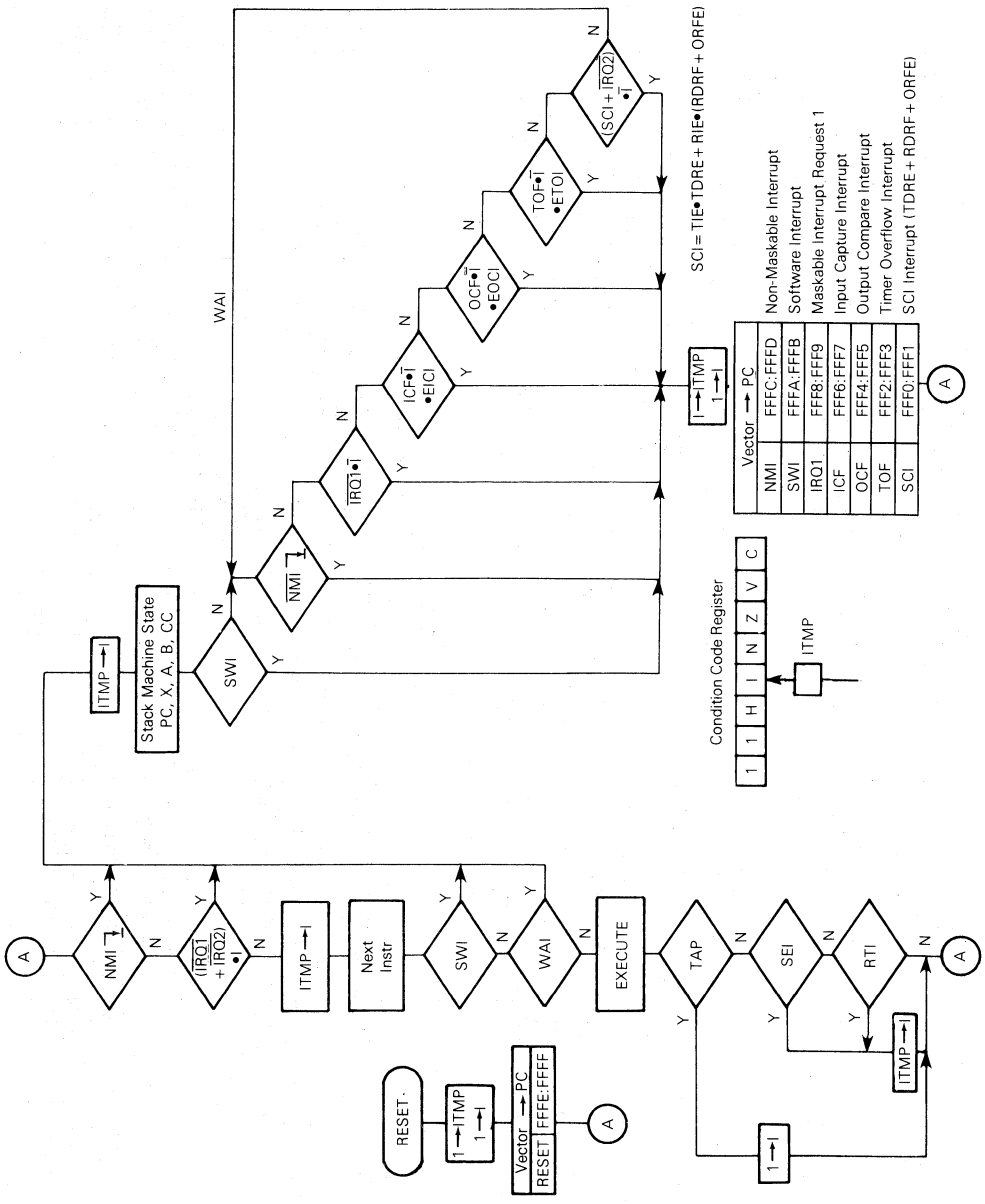


FIGURE 19 — INTERRUPT SEQUENCE

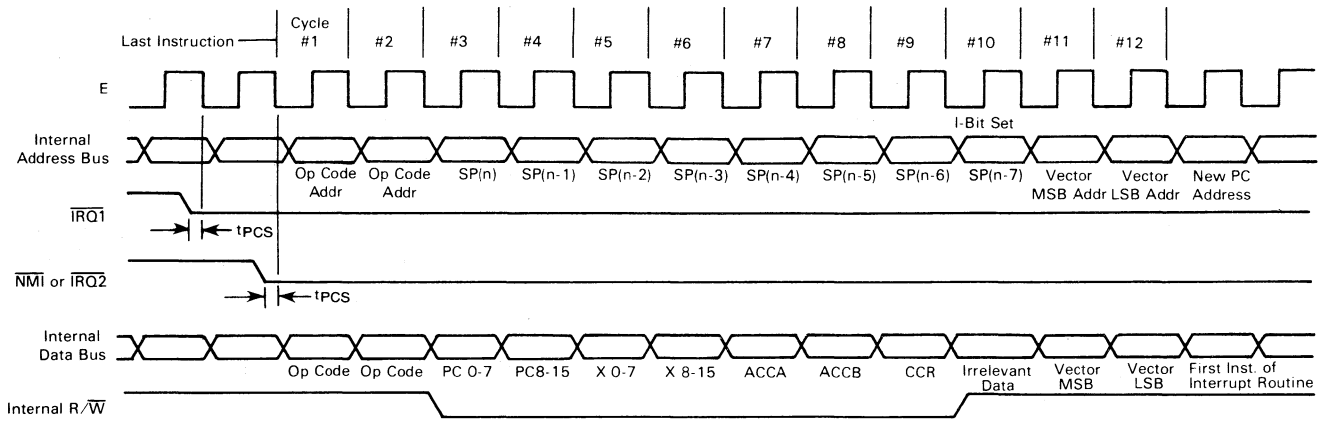
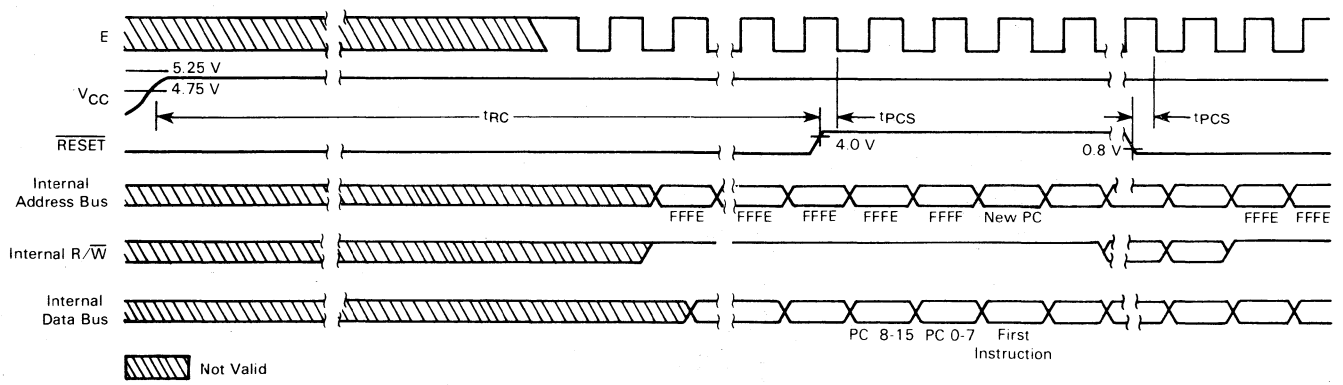


FIGURE 20 — RESET TIMING



XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz Color Burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL compatible clock at $4f_O$ with a duty cycle of 50% (±5%) with XTAL1 connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.* The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 21.

$\overline{\text{RESET}}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During powerup, $\overline{\text{RESET}}$ must be held below 0.8 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. $\overline{\text{RESET}}$ must be held low at least three E cycles if asserted during powerup operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

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

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

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

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

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

STROBE CONTROL 1 AND 2 (SC1 AND SC2)

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

SC1 and SC2 In Single-Chip Mode

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

SC2 is configured as $\overline{\text{OS3}}$ and can be used to strobe output data or acknowledge input data. It is controlled by output strobe select (OSS) in the port 3 control and status register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the port 3 data register. $\overline{\text{OS3}}$ timing is shown in Figure 4.

SC1 and SC2 In Expanded Non-Multiplexed Mode

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

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

SC1 and SC2 In Expanded-Multiplexed Mode

In the expanded-multiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as address strobe and can be used to demultiplex the eight least-significant addresses and the data bus. A latch controlled by address strobe captures address on the negative edge, as shown in Figure 14.

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

PORT 1 (P10-P17)

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

PORT 2 (P20-P24)

PORT 2 DATA REGISTER

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

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

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

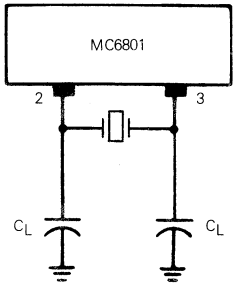
FIGURE 21 — M6801 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

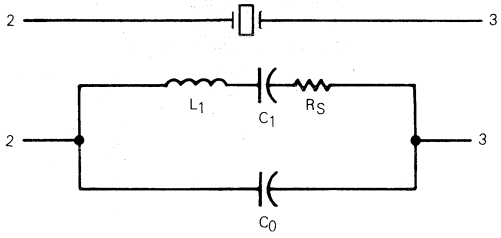
Nominal Crystal Parameters*

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

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



C_L = 20 pF (typical)



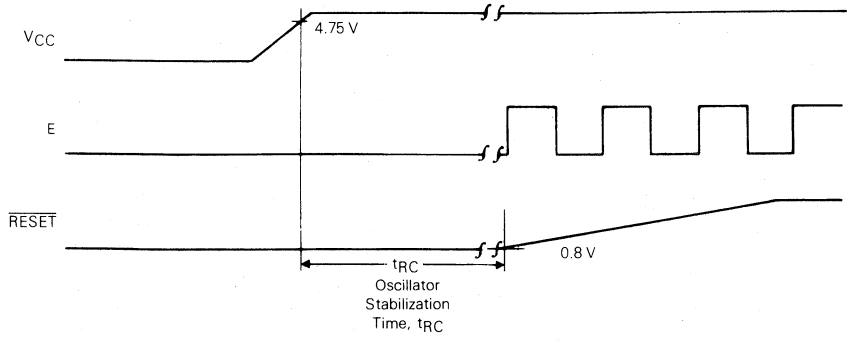
Equivalent Circuit

NOTE

TTL-compatible oscillators may be obtained from:

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

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



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

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

PORT 3 (P30-P37)

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

Port 3 In Single-Chip Mode

Port 3 is an 8-bit I/O port in the single-chip mode, with each line configured by the port 3 data direction register. There are also two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control port 3 data transfers.

Three port 3 options are controlled by the port 3 control and status register and are available only in single-chip mode: (1) port 3 input data can be latched using $\overline{IS3}$ as a control signal, (2) $\overline{OS3}$ can be generated by either an MPU read or write to the port 3 data register, and (3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 5.

PORT 3 CONTROL AND STATUS REGISTER

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

Bit 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of the port 3 data register. LATCH ENABLE is cleared during reset.

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

Bit 5 Not used.

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

Bit 7 IS3 FLAG. This read-only status bit is set by an $\overline{IS3}$ negative edge. It is cleared by a read of the port 3 control and status register (with IS3 FLAG set) followed by a read or write to the port 3 data register or during reset.

Port 3 In Expanded Non-Multiplexed Mode

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

Port 3 In Expanded-Multiplexed Mode

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

PORT 4 (P40-P47)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

Port 4 In Single-Chip Mode

In single-chip mode, port 4 functions as an 8-bit I/O port with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Port 4 In Expanded Non-Multiplexed Mode

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

Port 4 In Expanded-Multiplexed Mode

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

RESIDENT MEMORY

The MC6801 provides 2048 bytes of on-chip ROM and 128 bytes of on-chip RAM.

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

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM control register.

RAM CONTROL REGISTER ($14)

The RAM control register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM CONTROL REGISTER

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

Bit 0-5 Not used.

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

Bit 7 STBY PWR Standby Power. This bit is a read/write status bit which, when once set, remains set as long as V_{CC} standby remains above V_{SBB} (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that V_{CC} standby had fallen to a level sufficiently below V_{SBB} (minimum) to suspect that data in the standby RAM is not valid. This bit can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

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

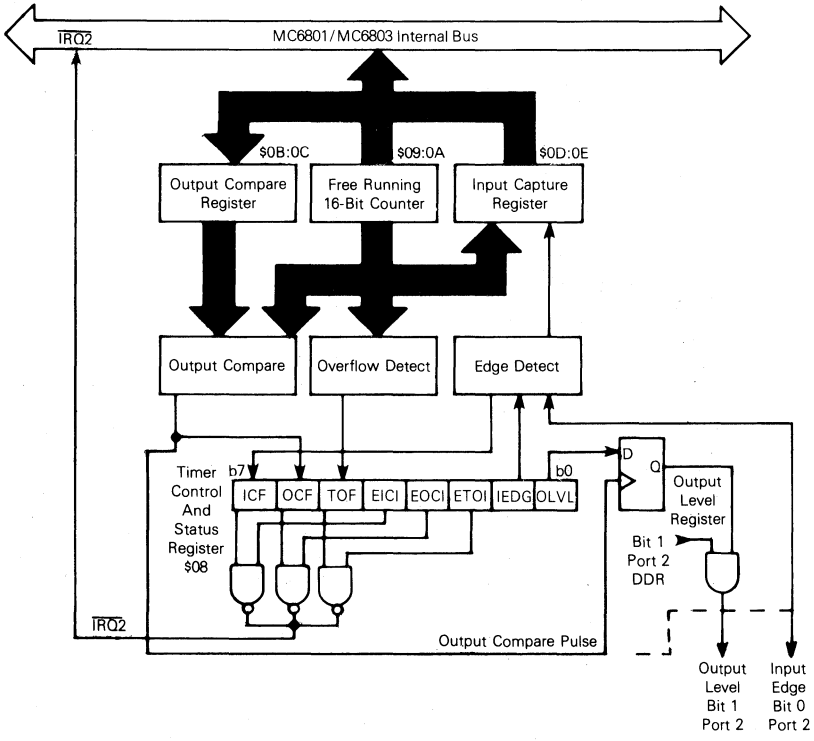
COUNTER ($09:0A)

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

OUTPUT COMPARE REGISTER ($0B:0C)

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

FIGURE 22 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The output compare register is set to $FFFF at RESET.

INPUT CAPTURE REGISTER ($0D:0E)

The input capture register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF: the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTER ($08)

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

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

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

TIMER CONTROL AND STATUS REGISTER (TCSR)

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

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

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

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

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

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

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

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

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

SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

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

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

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

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in Figure 23. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and

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

Rate and Mode Control Registers (RMCR) ($10)

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

RATE AND MODE CONTROL REGISTER (RMCR)

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

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

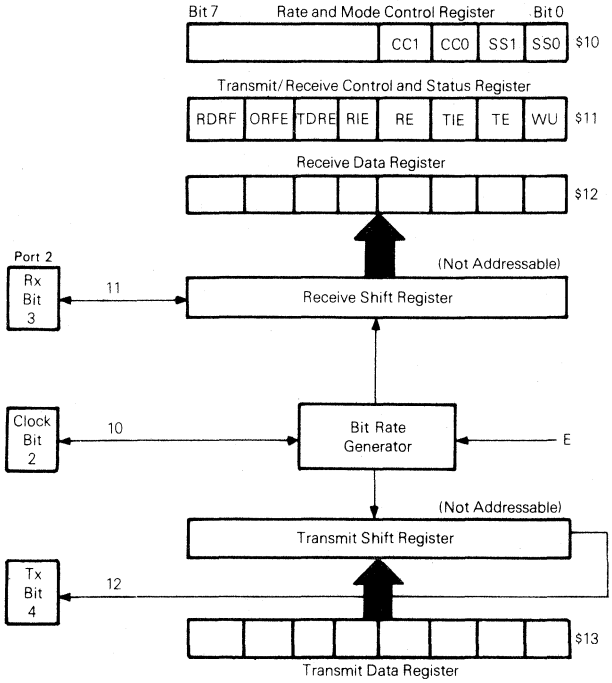
Bit 3:Bit 2 — CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE: The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter can disturb serial operations.

3

FIGURE 23 — SCI REGISTERS



Transmit/Receive Control And Status Register (TRCSR) ($11)

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

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

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

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

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

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

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

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

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

Bit 6 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared.* ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

Bit 7 RDRF — Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the receive data register. It is cleared by reading the TRCSR (with RDRF set), and then the receive data register, or during reset.

TABLE 6 — SCI BIT TIMES AND RATES

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

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

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

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

SERIAL OPERATIONS

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

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

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, ones will be sent until more data is provided. In Bi-phase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 24.

INSTRUCTION SET

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

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

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

PROGRAMMING MODEL

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

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

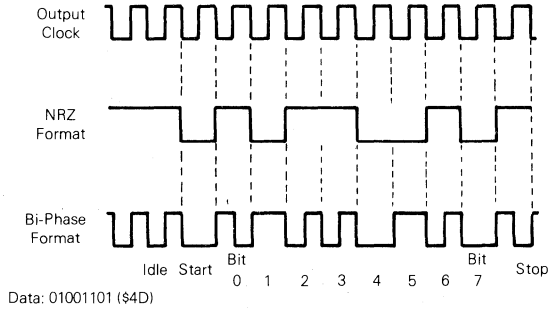
Stack Pointer — The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

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

Accumulators — The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

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

FIGURE 24 — SCI DATA FORMATS



ADDRESSING MODES

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

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

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

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

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

Inherent Addressing — The operand(s) are registers and no memory reference is required. These are single byte instructions.

Relative Addressing — Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of −126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	•				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	•				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	•				75	•				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	•				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	•				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	•				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	•				47	ASRA		2	1	7B	•				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	•				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	•				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	•				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	•				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	•				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	•				51	•				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	•				52	•				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	•				8F	•				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	•				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	•				99	ADCA		3	2	CD	•								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

* UNDEFINED OP CODE

NOTES: 1. Addressing Modes
INHER ≡ Inherent INDXD ≡ Indexed IMMED ≡ Immediate
REL ≡ Relative EXTND ≡ Extended DIR ≡ Direct
2. Unassigned opcodes are indicated by "•" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

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

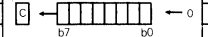
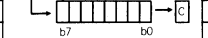
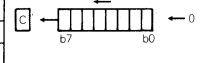
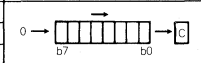
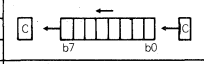
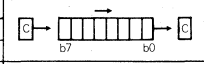
Accumulator and Memory Operations	MNEM	Immed Op	Immed ~	Immed #	Direct Op	Direct ~	Direct #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inher Op	Inher ~	Inher #	Boolean Expression	Condition Codes 5 H	4 I	3 N	2 Z	1 V	0 C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A•M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B•M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A•M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B•M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A – B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A – M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B – M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

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

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

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct Op	Direct ~	Direct #	Relative Op	Relative ~	Relative #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inherent Op	Inherent ~	Inherent #	Branch Test	Condition Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3				See Special Operations-Figure 25	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 25	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

SUMMARY OF CYCLE-BY-CYCLE OPERATION

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

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

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

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

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

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
EXTENDED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS LDX LDD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS STX STD	5	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST* INC	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address of Operand	0	New Operand Data
CPX SUBD ADDD	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Operand Address (High Order Byte)
		3	Opcode Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

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

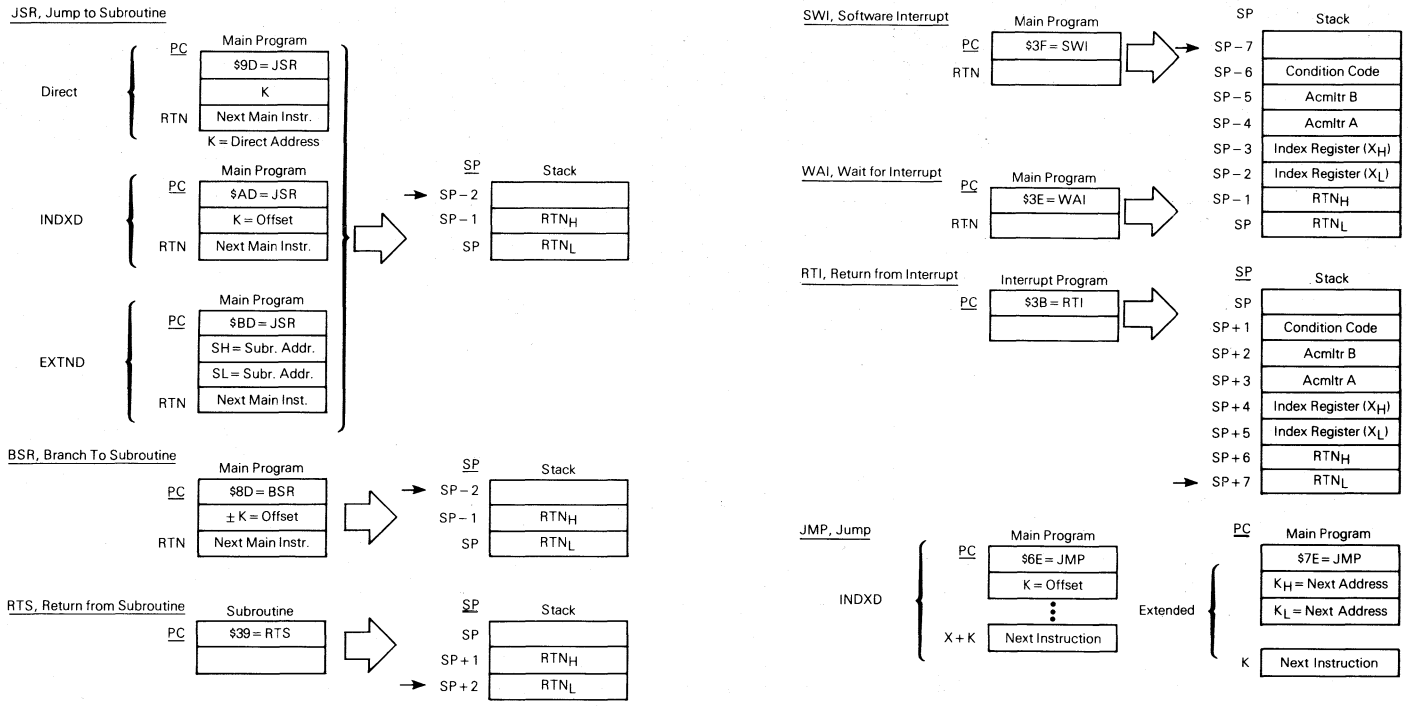
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC* SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer – 1 Stack Pointer – 2 Stack Pointer – 3 Stack Pointer – 4 Stack Pointer – 5 Stack Pointer – 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Buss FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

FIGURE 25 — SPECIAL OPERATIONS



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

APPENDIX
CUSTOM MC6801 ORDERING INFORMATION

A.0 CUSTOM MC6801 ORDERING INFORMATION

The custom MC6801 specifications may be transmitted to Motorola in any of the following media:

1) PROM(s)
2) MDOS diskette

The specification should be formatted and packed, as indicated in the appropriate paragraph below, and mailed prepaid and insured with a cover letter (see Figure A-2) to:

Motorola Inc.
MPU Marketing L10
3501 Ed Bluestein Blvd.
Austin, Texas 78721

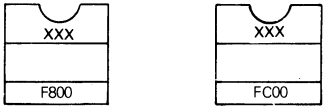
A copy of the cover letter should also be mailed separately.

A.1 PROMs

MCM2708 and MCM2716 type PROMs, programmed with the custom program (positive logic sense for address and data), may be submitted for pattern generation. The MC2708s must be clearly marked to indicate which PROM corresponds to which address space ($F800-$FFFF). See Figure A-1 for recommended marking procedure.

After the PROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE A-1



XXX = Customer ID

A.2 DISKETTE (MDOS)

The start/end location should be written on the label. EXORciser format.

FIGURE A-2

CUSTOMER NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE (____) _____ EXTENSION _____

CONTACT MS/MR _____

CUSTOMER PART # _____

TEMPERATURE RANGE
- ☐ 0° to 70°C
- ☐ −40 to 85°C
- ☐ −40 to 105°C

PACKAGE TYPE
- ☐ Ceramic
- ☐ Plastic

MARKING
- ☐ Standard
- ☐ Special

PATTERN MEDIA
- ☐ 2708 PROM
- ☐ 2716 PROM
- ☐ Diskette (MDOS) (Note 1) _____

NOTE: (1) Other Media Require Prior Factory Approval

SIGNATURE _____

TITLE _____

MC6801L1 — LILBug™ Monitor

An MC6801 may be purchased without specifying the ROM pattern. This standard part is labeled as MC6801L1 and contains a 2K monitor in the ROM. The monitor, LILbug, may be used to evaluate and debug a program under development. Details and a source listing are specified in the "LILbug Manual."

IMPORTANT NOTICE

Devices made with mask #T5P may generate multiple framing error flags in response to unframed data. These devices will eventually synchronize correctly after a framing error, but valid, framed data following an unframed data byte may generate false framing error flags.

MOTOROLA

MC6801U4
MC6803U4

Advance Information

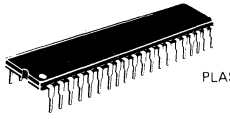
HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

MICROCOMPUTER/ MICROPROCESSOR

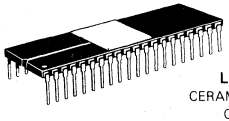
MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

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

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

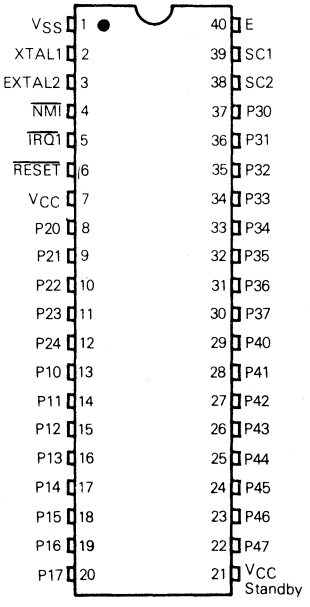


P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT

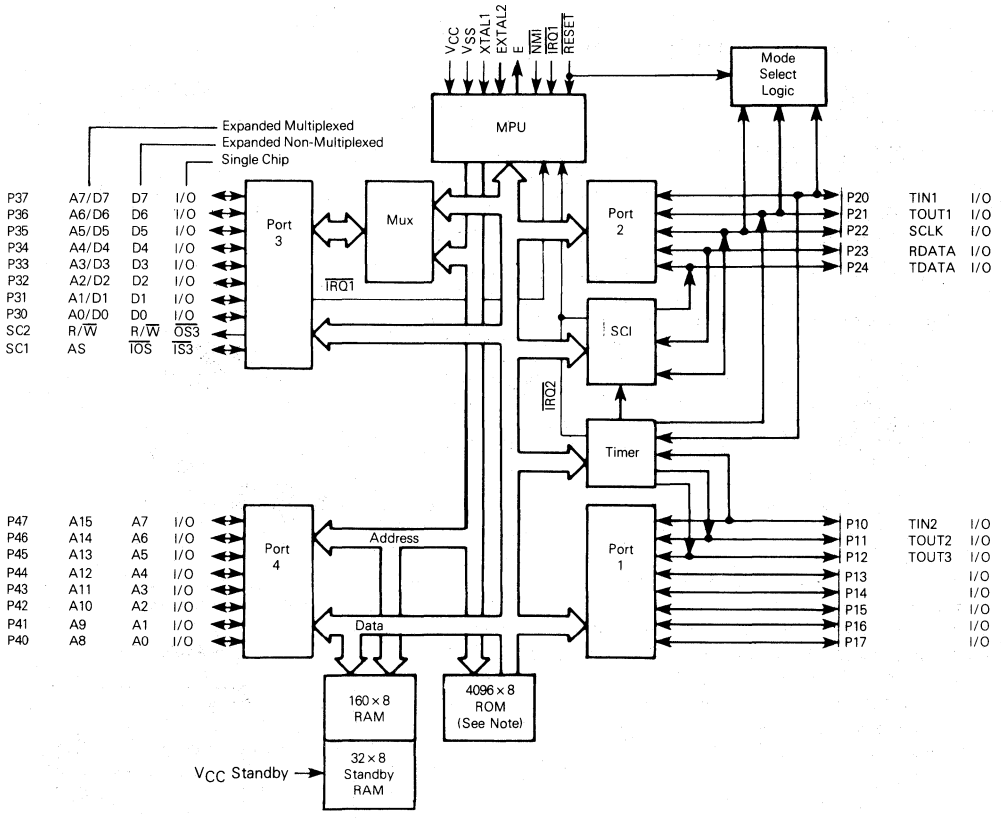


GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6801U4L1
	1.0	−40°C to 85°C	MC6801U4CL1
	1.0	0°C to 70°C	MC6803U4L
	1.0	−40°C to 85°C	MC6803U4CL1
	1.25	0°C to 70°C	MC6801U4L1-1
	1.25	0°C to 70°C	MC6803U4L-1
Plastic P Suffix	1.0	0°C to 70°C	MC6801U4P1
	1.0	−40°C to 85°C	MC6801U4CP1
	1.0	0°C to 70°C	MC6803U4P
	1.0	−40°C to 85°C	MC6803U4CP1
	1.25	0°C to 70°C	MC6801U4P1-1
	1.25	0°C to 70°C	MC6803U4P-1

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

MC6801U4 MICROCOMPUTER FAMILY BLOCK DIAGRAM



NOTE: No functioning ROM in MC6803U4.

MAXIMUM RATINGS

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

THERMAL CHARACTERISTICS

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

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

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic	Symbol	MC6801U4 MC6803U4		MC6801U4-1 MC6803U4-1		Unit
		Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	MHz
External Oscillator Frequency	4 f_o	2.0	4.0	2.0	5.0	MHz
Crystal Oscillator Startup Time	t_{rc}	–	100	–	100	ms
Processor Control Setup Time	t_{PCS}	200	–	170	–	ns

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

Characteristic		Symbol	MC6801U4, MC6803U4 Min	MC6801U4, MC6803U4 Max	MC6801U4C, MC6803U4C Min	MC6801U4C, MC6803U4C Max	Unit
Input High Voltage	$\overline{RESET}$	V_{IH}	V_{SS}+4.0	V_{CC}	V_{SS}+4.0	V_{CC}	V
	Other Inputs*		V_{SS}+2.0	V_{CC}	V_{SS}+2.2	V_{CC}	
Input Low Voltage	All Inputs*	V_{IL}	V_{SS}−0.3	V_{SS}+0.8	V_{SS}−0.3	V_{SS}+0.8	V
Input Load Current	Port 4	I_{in}	–	0.5	–	0.8	mA
	SCI		–	0.8	–	1.0	
Input Leakage Current (V_{in}=0 to 5.5 V)	$\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	2.5	–	5.0	μA
Hi-Z (Off-State) Input Current (V_{in}=0.5 to 2.4 V)	Port 1, Port 2, Port 3	I_{TSI}	–	10	–	20	μA
Output High Voltage (I_{Load}=−65 μA, V_{CC}=Min)	Port 4, SC1, SC2	V_{OH}	V_{SS}+2.4	–	V_{SS}+2.4	–	V
(I_{Load}=−100 μA, V_{CC}=Min)	Other Outputs		V_{SS}+2.4	–	V_{SS}+2.4	–	
Output Low Voltage (I_{Load}=2.0 mA, V_{CC}=Min)	All Outputs	V_{OL}	–	V_{SS}+0.5	–	V_{SS}+0.6	V
Darlington Drive Current (V_O=1.5 V)	Port 1	I_{OH}	1.0	4.0	1.0	5.0	mA
Internal Power Dissipation (Measured at T_A=T_L in Steady-State Operation)***		P_{INT}	–	1200	–	1500	mW
Input Capacitance (V_{in}=0, T_A=25°C, f_o=1.0 MHz)	Port 3, Port 4, SC1	C_{in}	–	12.5	–	12.5	pF
	Other Inputs		–	10.0	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}	4.0	5.25	4.0	5.25	V
	Powerup	V_{SB}	4.75	5.25	4.75	5.25	
Standby Current	Powerdown	I_{SBB}	–	3.0	–	3.5	mA

* Except mode programming levels; see Figure 16.

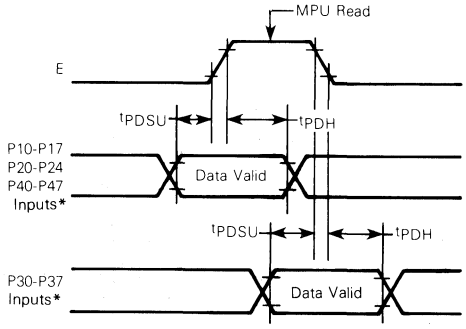
** Negotiable to −100 μA (for further information contact the factory).

*** For the MC6801U4/MC6803U4 T_L=0°C and for the MC6801U4C/MC6803U4C T_L=−40° C

PERIPHERAL PORT TIMING (Refer to Figures 1-4)

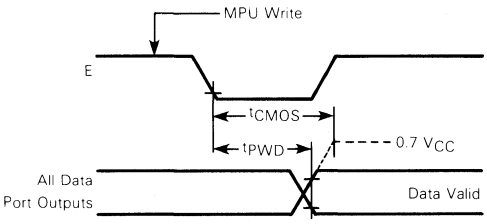
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid					
Port 1	t_{PWD}	–	–	350	ns
Port 2, 3, 4		–	–	350	
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	–	ns
Input Data Hold Time	t_{IH}	50	–	–	ns
Input Data Setup Time	t_{IS}	20	–	–	ns

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



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

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



NOTES:
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

3

FIGURE 3 — PORT 3 OUTPUT STROBE TIMING
(MC6801U4 SINGLE-CHIP MODE)

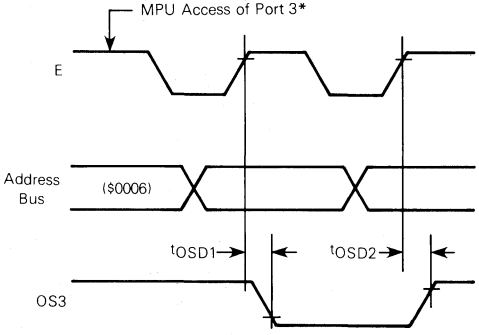
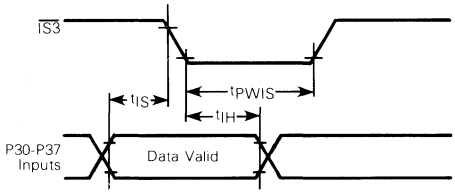


FIGURE 4 — PORT 3 LATCH TIMING
(MC6801U4 SINGLE-CHIP MODE)



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

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

FIGURE 5 — CMOS LOAD

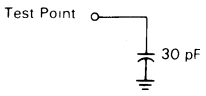
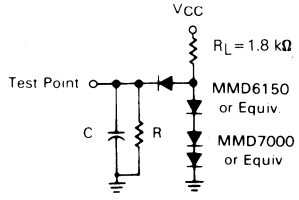


FIGURE 6 — TIMING TEST
LOAD PORTS 1, 2, 3, AND 4



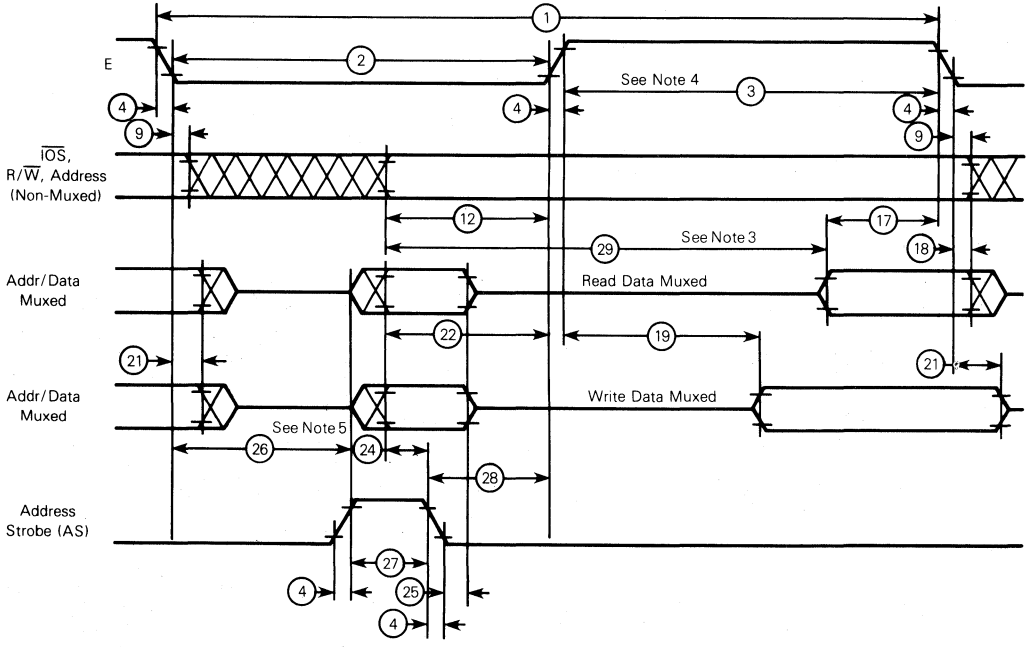
C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P10-P17, P20-P24
= 24 kΩ for P30-P37, E

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

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

*At specified cycle time.

FIGURE 7 — BUS TIMING



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

INTRODUCTION

The MC6801U4 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a data register and a write-only data direction register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port" by itself refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

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

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

3

FIGURE 8 — PROGRAMMING MODEL

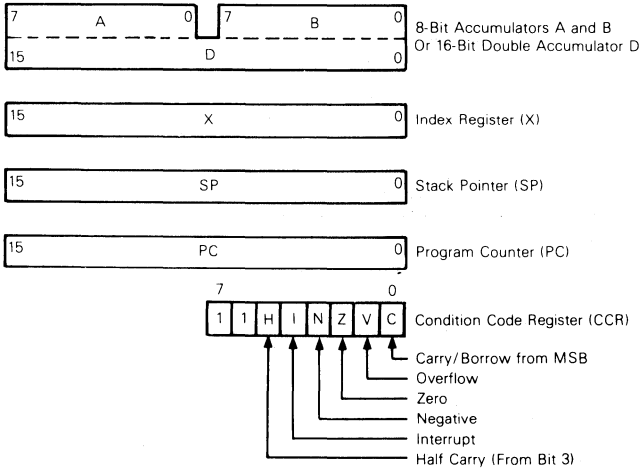


TABLE 1 — NEW INSTRUCTIONS

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

OPERATING MODES

The MC6801U4 provides seven different operating modes (modes 0 through 3 and 5 through 7) and the MC6803U4 provides two operating modes (modes 2 and 3). The operating modes are hardware selectable and determine the device memory map, the configuration of port 3, port 4, SC1, SC2, and the physical location of the interrupt vectors.

FUNDAMENTAL MODES

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

MC6801U4 SINGLE-CHIP MODE (7) – In the single-chip mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 9. The MCU functions as a monolithic microcomputer in this mode without external address or data buses. A maximum of 29 I/O lines and two port 3 control lines are provided. Peripherals or another MCU can be interfaced to port 3 in a loosely coupled dual-processor configuration, as shown in Figure 10.

3

TABLE 2 – SUMMARY OF MC6801U4/MC6803U4 OPERATING MODES

Single-Chip (Mode 7)
- 192 bytes of RAM, 4096 bytes of ROM
- Port 3 is a parallel I/O port with two control lines
- Port 4 is a parallel I/O port

Expanded Non-Multiplexed (Mode 5)
- 192 bytes of RAM, 4096 bytes of ROM
- 256 bytes of external memory space
- Port 3 is an 8-bit data bus
- Port 4 is an input port/address bus

Expanded Multiplexed (Modes 0, 1, 2, 3, 6*)
- Four memory space options (total 64K address space)
  - (1) Internal RAM and ROM with partial address bus (mode 1)
  - (2) Internal RAM, no ROM (mode 2)
  - (3) Extended addressing of internal I/O and RAM
  - (4) Internal RAM and ROM with partial address bus (mode 6)
- Port 3 is multiplexed address/data bus
- Port 4 is address bus (inputs/address in mode 6)
- Test mode (mode 0):
  - May be used to test internal RAM and ROM
  - May be used to test ports 3 and 4 as I/O ports by writing into mode 7
- Only modes 5, 6, and 7 can be irreversibly entered from mode 0

Resources Common to All Modes
- Reserved register area
- Port 1 input/output operation
- Port 2 input/output operation
- Timer operation
- Serial communications interface operation

*The MC6803U4 operates only in modes 2 and 3.

FIGURE 9 — SINGLE-CHIP MODE

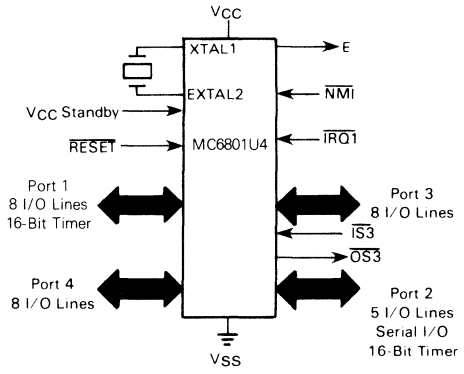
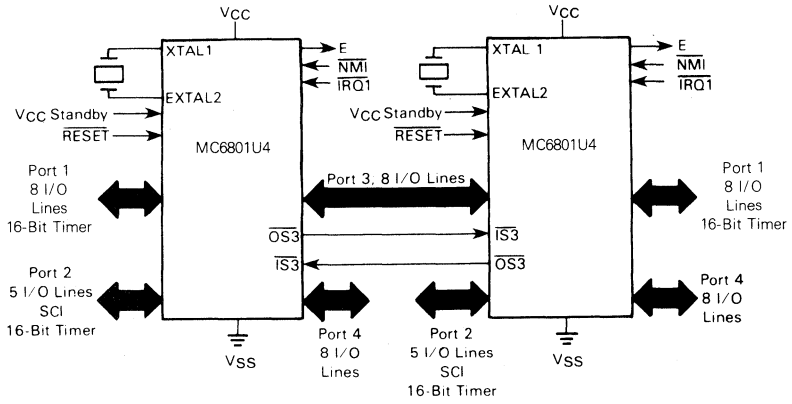


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION



MC6801U4 EXPANDED NON-MULTIPLEXED MODE (5) — A modest amount of external memory space is provided in the expanded non-multiplexed mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and port 4 is configured initially as an input data port. Any combination of the eight least significant address lines may be obtained by writing to the port 4 data direction register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the port 4 lines high until the port is configured.

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

EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) — A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E is high. In modes 0, 2, and 3, port 4 provides address lines A8 to A15. In modes 1 and 6, however, port 4 initially is configured at reset as an input data port. The port 4 data direction register can then be changed to provide any combination of address lines A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining port 4 lines as input data lines. Internal pullup resistors pull the port 4 lines high until software configures the port. In mode 1, the internal pullup resistors will hold the upper address lines high producing a value of $FFXX for a reset vector. A simple method of getting the desired address lines configured as outputs is to have an external EPROM not fully decoded so it appears at

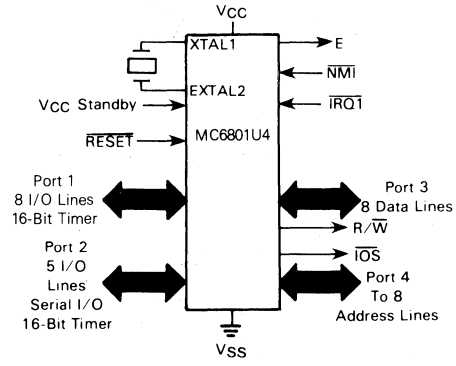
two address locations (i.e., $FXXX and $BXXX). Then, when the reset vector appears as $FFFE, the EPROM will be accessed and can point to an address in the top $100 bytes of the internal or external ROM/EPROM that will configure port 4 as desired.

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

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

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

FIGURE 11 — EXPANDED NON-MULTIPLEXED CONFIGURATION



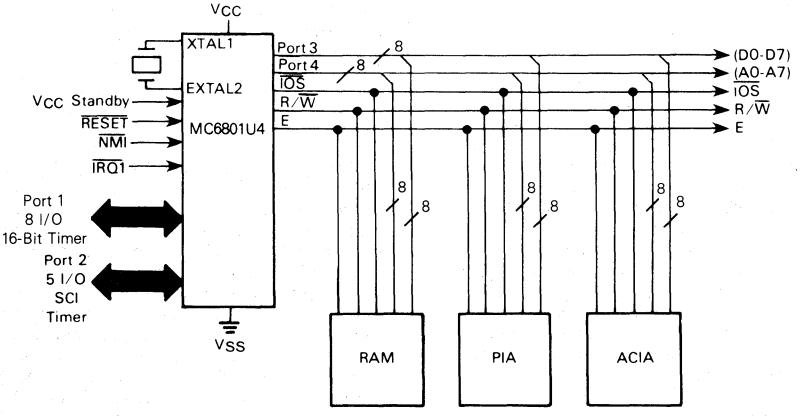
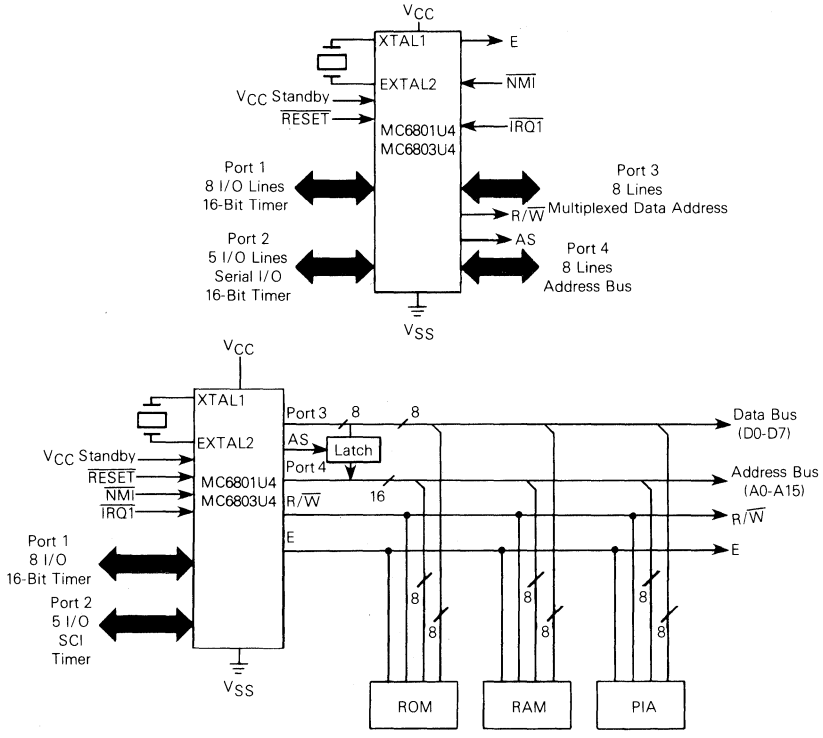
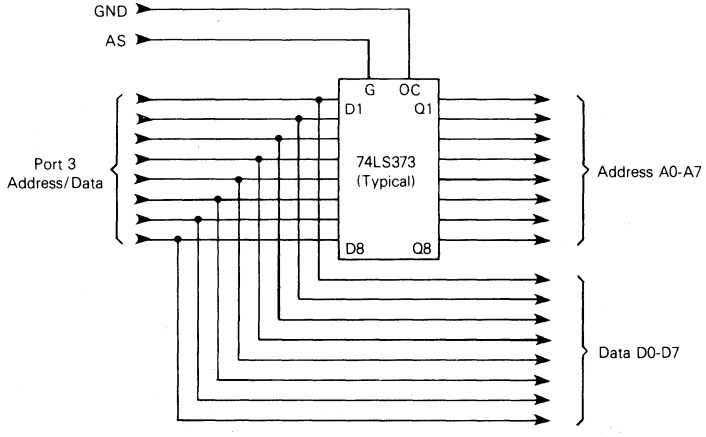


FIGURE 12 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 13 — TYPICAL LATCH ARRANGEMENT



PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

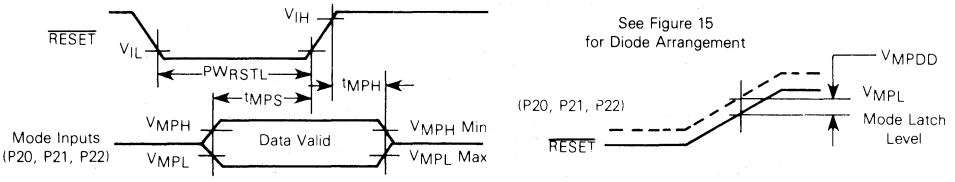
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

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

FIGURE 14 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 14)

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

TABLE 3 — MODE SELECTION SUMMARY

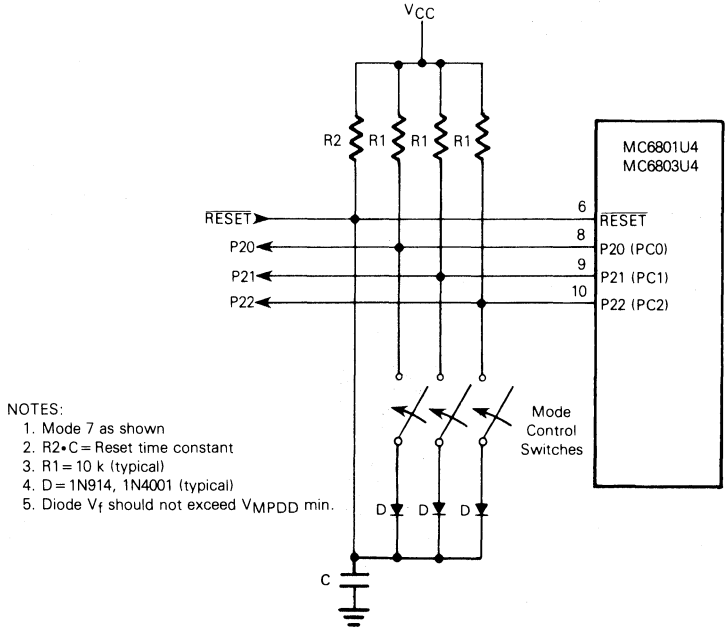
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(2, 3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(2, 3)	Non-Multiplexed/Partial Decode
4	H	L	L	—	—	—	—	Undefined(4)
3	L	H	H	E	I	E	MUX(1, 5)	Multiplexed/RAM
2	L	H	L	E	I	E	MUX(1)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

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

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

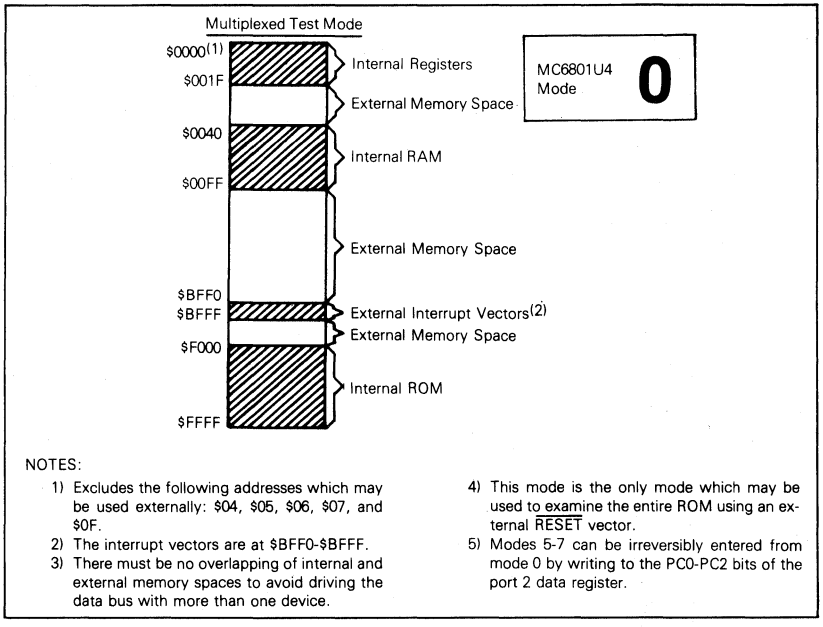
* The MC6803U4 operates only in modes 2 and 3.

FIGURE 15 — TYPICAL MODE PROGRAMMING CIRCUIT



NOTES:
1. Mode 7 as shown
2. R2•C = Reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

FIGURE 16 — MC6801U4/MC6803U4 MEMORY MAPS (Sheet 1 of 4)



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

FIGURE 16 — MC6801U4/MC6803U4 MEMORY MAPS (Sheet 2 of 4)

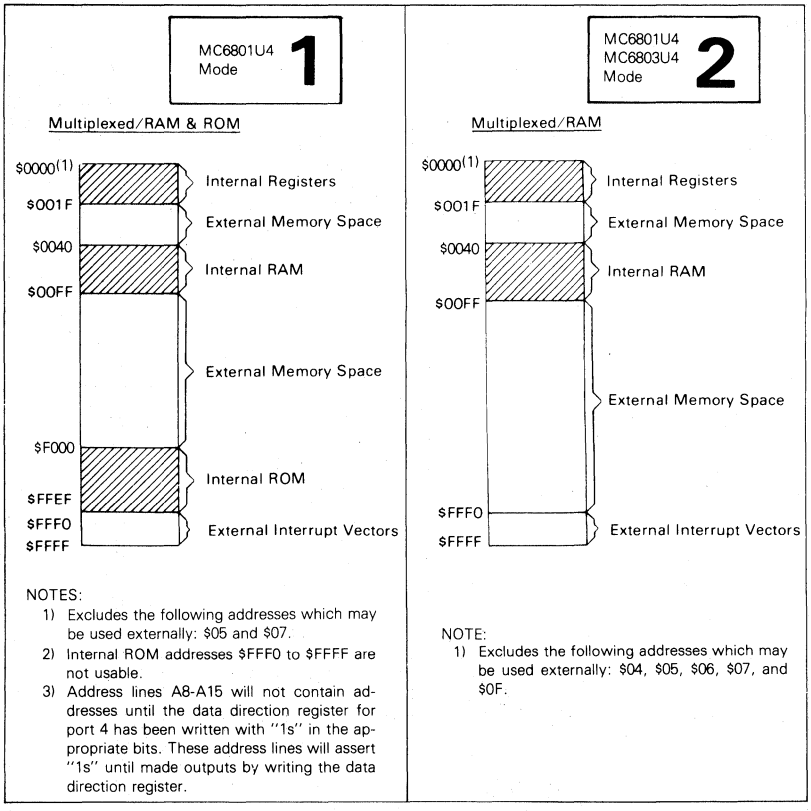
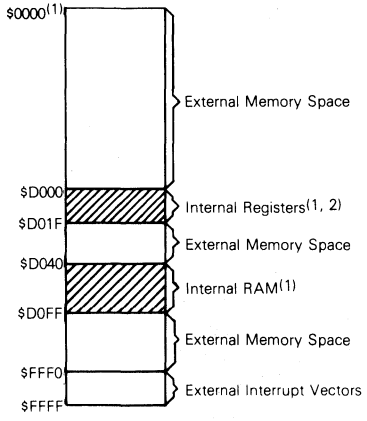


FIGURE 16 – MC6801U4/MC6803U4 MEMORY MAPS (Sheet 3 of 4)

MC6801U4
MC6803U4
Mode **3**

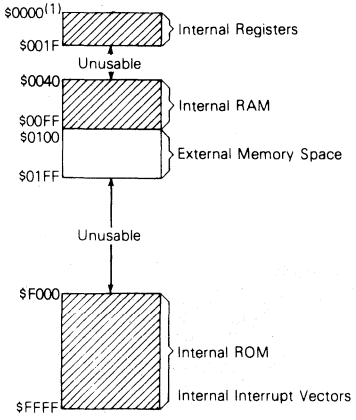
Multiplexed/RAM



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

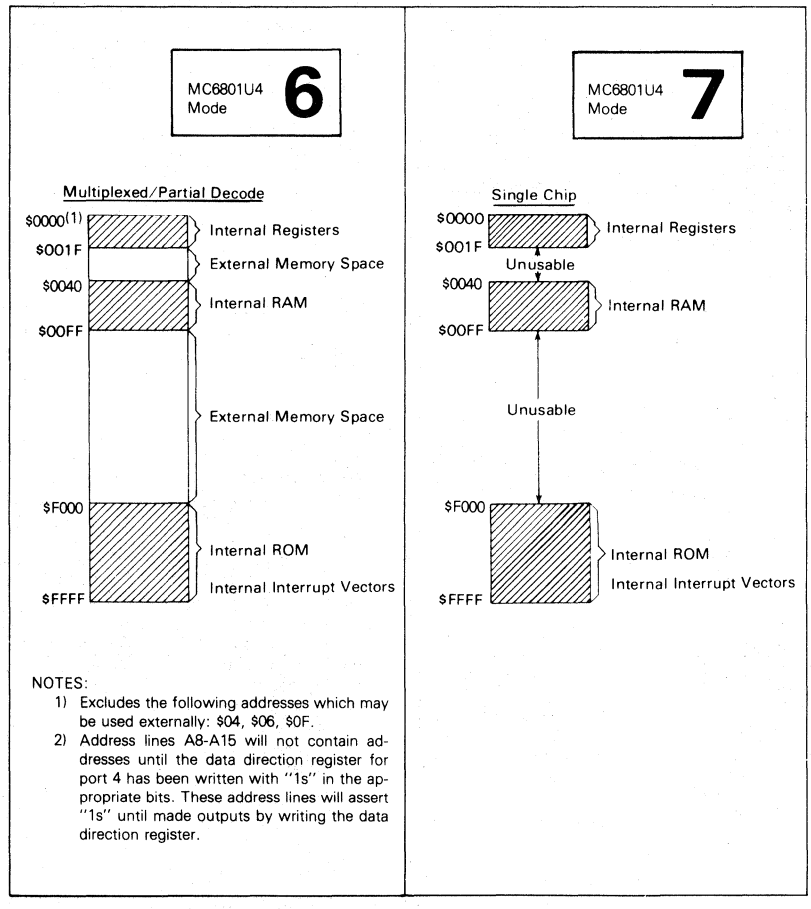
MC6801U4
Mode **5**

Non-Multiplexed/Partial Decode



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

FIGURE 16 — MC6801U4/MC6803U4 MEMORY MAPS (Sheet 4 of 4)



NOTES:

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

TABLE 4 — INTERNAL REGISTER AREA

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

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

MC6801U4/MC6803U4 INTERRUPTS

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

NOTE

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

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

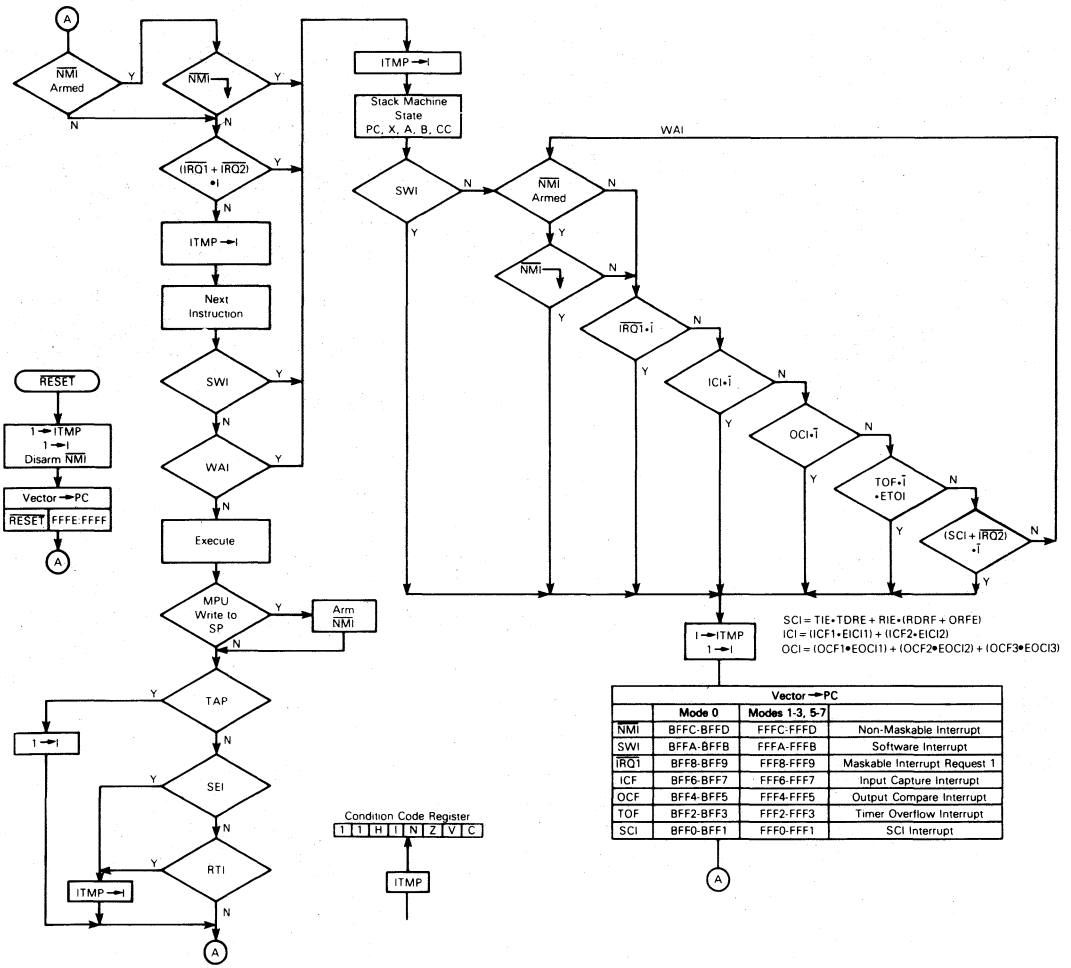
The interrupt flowchart is depicted in Figure 17 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, index register, A accumulator, B accumulator, and condition code register are pushed to the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 18 and 19.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

*$\overline{IRQ2}$ interrupt
**$\overline{NMI}$ must be armed (by accessng stack pointer) before an $\overline{NMI}$ is executed.
*** Mode 4 interrupt vectors are undefined.

FIGURE 17 — INTERRUPT FLOWCHART



Vector → PC			
	Mode 0	Modes 1-3, 5-7	
NMI	BFFC-BFFD	FFFC-FFFD	Non-Maskable Interrupt
SWI	BFFA-BFFB	FFFA-FFFB	Software Interrupt
IRQ1	BFF8-BFF9	FFF8-FFF9	Maskable Interrupt Request 1
ICF	BFF6-BFF7	FFF6-FFF7	Input Capture Interrupt
OCF	BFF4-BFF5	FFF4-FFF5	Output Compare Interrupt
TOF	BFF2-BFF3	FFF2-FFF3	Timer Overflow Interrupt
SCI	BFF0-BFF1	FFF0-FFF1	SCI Interrupt

FIGURE 18 — INTERRUPT SEQUENCE

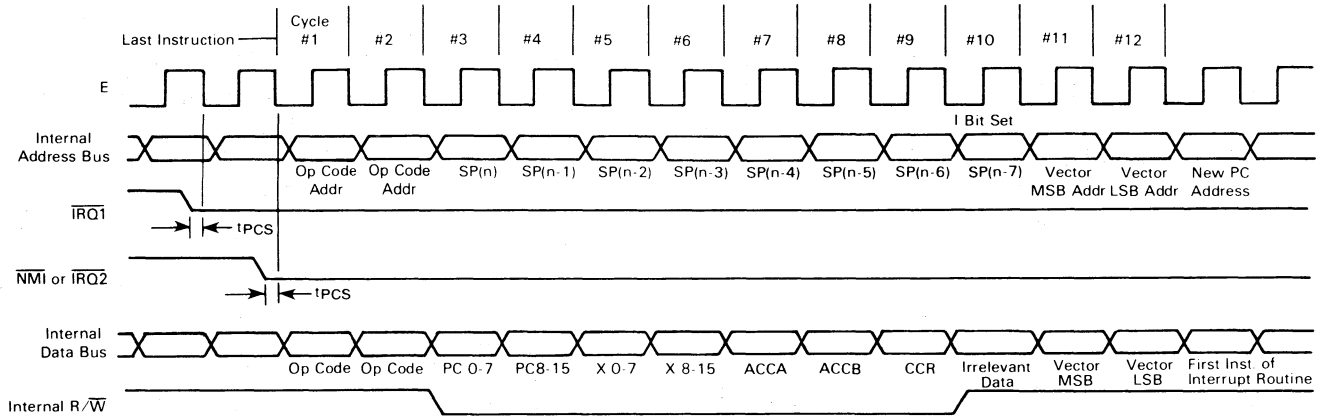
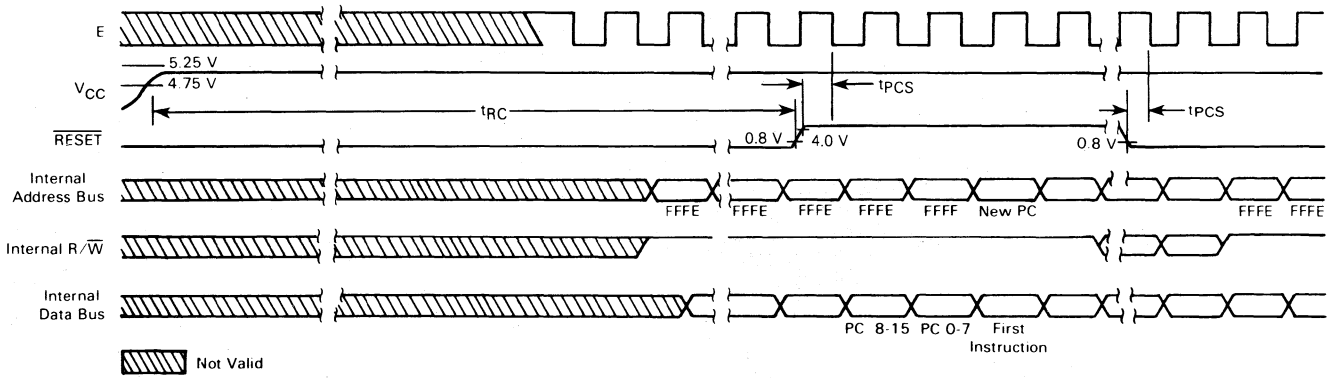


FIGURE 19 — RESET TIMING



FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

V_{CC} standby provides power to the standby portion ($40 through $5F in all modes except mode 3 which is $D040 through $D05F) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide +5 volts (±5%) and must reach V_{SB} volts before $\overline{RESET}$ reaches 4.0 volts. During power down, V_{CC} standby must remain above V_{SBB} (minimum) to sustain the standby RAM and STBY PWR bit. While in power-down operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during power-down operation.

XTAL1 AND EXTAL 2

These two input pins interface either a crystal or TTL-compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz color burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL-compatible clock at 4 f_O with a duty cycle of 50% (±5%) with XTAL1 connected ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 20.

$\overline{RESET}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{RESET}$ must be held below 0.8 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. $\overline{RESET}$ must be held low at least three E cycles if asserted during power-up operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

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

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

NOTE

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

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

$\overline{IRQ1}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9 ($BFF8 and $BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

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

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

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

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

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

SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE – In the expanded non-multiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

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

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

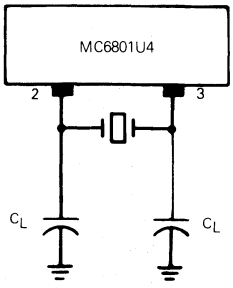
FIGURE 20 — MC6801U4/MC6803U4 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

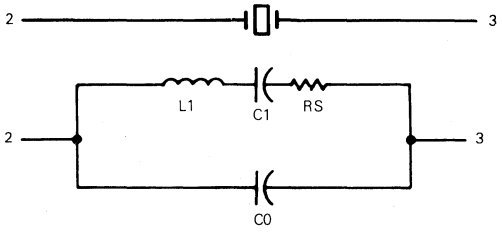
Nominal Crystal Parameters*

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

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



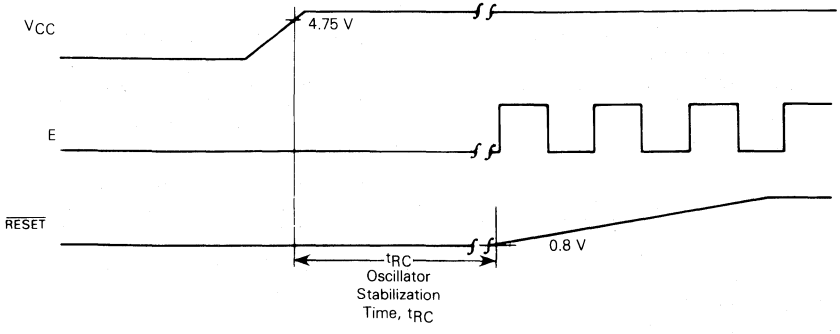
C_L = 20 pF (typical)



Equivalent Circuit

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

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



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

P10-P17 (PORT 1)

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

P20-P24 (PORT 2)

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

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

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

PORT 2 DATA REGISTER

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

P30-P37 (PORT 3)

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

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

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

PORT 3 CONTROL AND STATUS REGISTER

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

Bits 0-2 Not used.

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

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

Bit 5 Not used.

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

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

PORT 3 IN EXPANDED NON-MULTIPLEXED MODE — Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded non-multiplexed mode. The direction of data transfers is controlled by read/write (SC2). Data is clocked by E (enable).

PORT 3 IN EXPANDED MULTIPLEXED MODE — Port 3 is configured as a time multiplexed address (A7-A0) and data bus (D7-D0) in the expanded multiplexed mode where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF, and is the only port with internal pullup resistors. Unused lines can remain unconnected.

PORT 4 IN SINGLE-CHIP MODE — In single-chip mode, port 4 functions as an 8-bit I/O port with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5-volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

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

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

RESIDENT MEMORY

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

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

Power must be supplied to V_{CC} standby if the internal RAM is to be used regardless of whether standby power operation is anticipated.

The RAM is controlled by the RAM control register.

RAM CONTROL REGISTER ($14)

The RAM control register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down procedure.

RAM CONTROL REGISTER

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

Bits 0-5 Not used.

Bit 6 **RAM Enable** — This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of $\overline{\text{RESET}}$. If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

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

PROGRAMMABLE TIMER

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

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

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

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

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

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

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

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

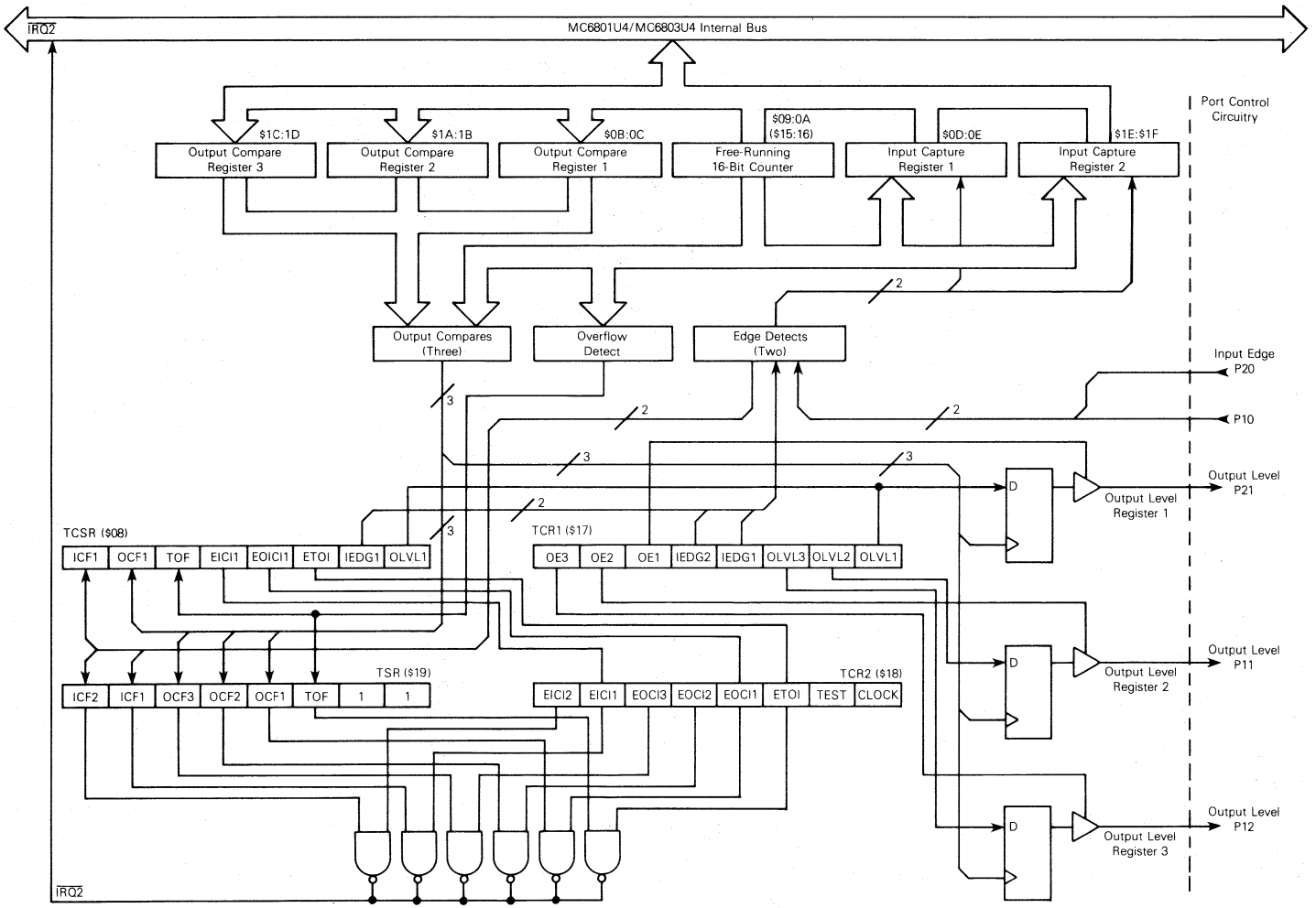
An input capture can occur independently of ICF; the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTERS

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

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

FIGURE 21 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



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

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

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

TIMER CONTROL AND STATUS REGISTER

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

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

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

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

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

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

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

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

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

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

TIMER CONTROL REGISTER 1

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

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

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

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

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

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

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

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

Bit 7 **Output Enable 3** — OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set
OE3 = 0 port 1 bit 2 data register output
OE3 = 1 output level register 3

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

TIMER CONTROL REGISTER 2
(Non-Test Modes)

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

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

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

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

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

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

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

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

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

TIMER CONTROL REGISTER 2
(Test Mode)

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

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

Bit 1 **TEST** — the TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.
TEST = 0 — Timer test mode enabled:
a) The timer LSB latch is transparent which allows the LSB to be read independently of the MSB.
b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.
TEST = 1 — Timer test mode disabled.

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

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

TIMER STATUS REGISTER

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

Bits 0-1 Not used.

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

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

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

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

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

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

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

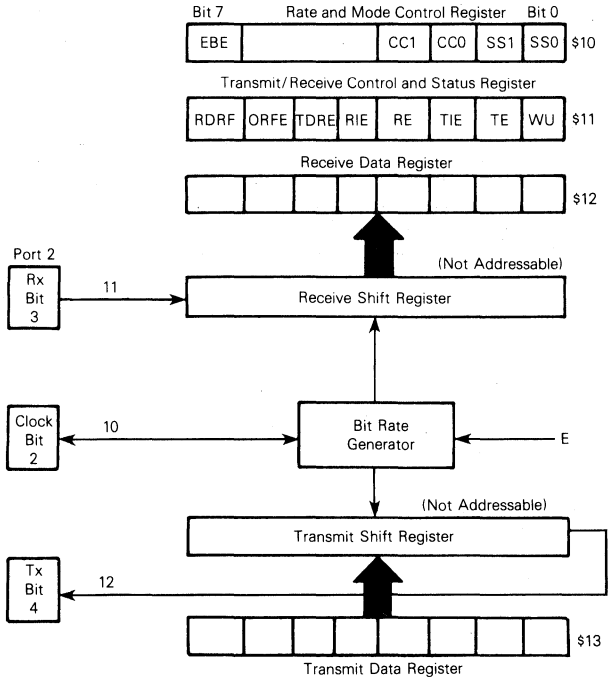
The following features of the SCI are programmable:

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

SERIAL COMMUNICATIONS REGISTERS

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

FIGURE 22 — SCI REGISTERS



RATE AND MODE CONTROL REGISTER (RMCR) ($10) — The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of five write-only bits which are cleared during reset. The two least significant bits in conjunction with bit 7 control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER

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

Bit 1:Bit 0 **SS1:SS0 Speed Select** — These two bits select the baud when using the internal clock. Eight rates may be selected (in conjunction with bit 7) which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3:Bit 2 **CC1:CC0 Clock Control and Format Select** — These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

Bits 4-6 Not used.

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

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

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (8×) the desired bit rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE

The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter in mode 0 can disturb serial operations.

TABLE 6 — SCI BIT TIMES AND RATES

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

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

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

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) — The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{\text{RESET}}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

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

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

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

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

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

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

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

Bit 6 **Overrun Framing Error** — If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared. ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

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

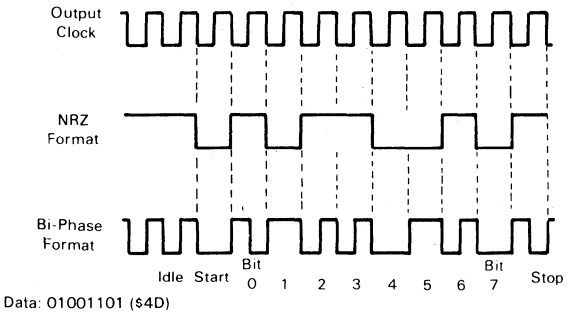
SERIAL OPERATIONS

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

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

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

FIGURE 23 — SCI DATA FORMATS



INSTRUCTION SET

The MC6801U4/MC6803U4 is directly source compatible with the MC6801 and upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced and several instructions have been added, including a hardware multiply. A list of new operations added to the MC6800 instruction set is shown in Table 1.

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

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

TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	•				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	•				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	•				75	•				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	•				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	•				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	•				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	•				47	ASRA		2	1	7B	•				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	•				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	•				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	•				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	•				4C	INCA		2		80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	•				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	•				51	•				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	•				52	•				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	•				8F	•				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	•				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	•				99	ADCA		3	2	CD	•								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

*UNDEFINED OP CODE

NOTES:

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

PROGRAMMING MODEL

A programming model for the MC6801U4/MC6803U4 is shown in Figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulators A and/or B. Other registers are defined as follows:

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

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

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

ACCUMULATORS – The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

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

ADDRESSING MODES

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

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

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

EXTENDED ADDRESSING – The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

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

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

RELATIVE ADDRESSING – Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of −126 to +129 bytes from the first byte of the instruction. These are two byte instructions.

SUMMARY OF CYCLE-BY-CYCLE OPERATION

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

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

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

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

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

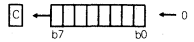
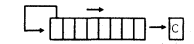
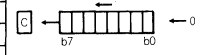
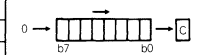
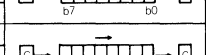
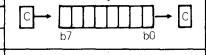
Accumulator and Memory Operations	MNEM	Immed Op	Immed ~	Immed #	Direct Op	Direct ~	Direct #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inher Op	Inher ~	Inher #	Boolean Expression	Condition Codes 5 H	4 I	3 N	2 Z	1 V	0 C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A • M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B • M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A • M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B • M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

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

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

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Condition Code Reg. 5	4	3	2	1	0
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3				See Special Operations-Figure 24	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 24	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

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

TABLE 13 – INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

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

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

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

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

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

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

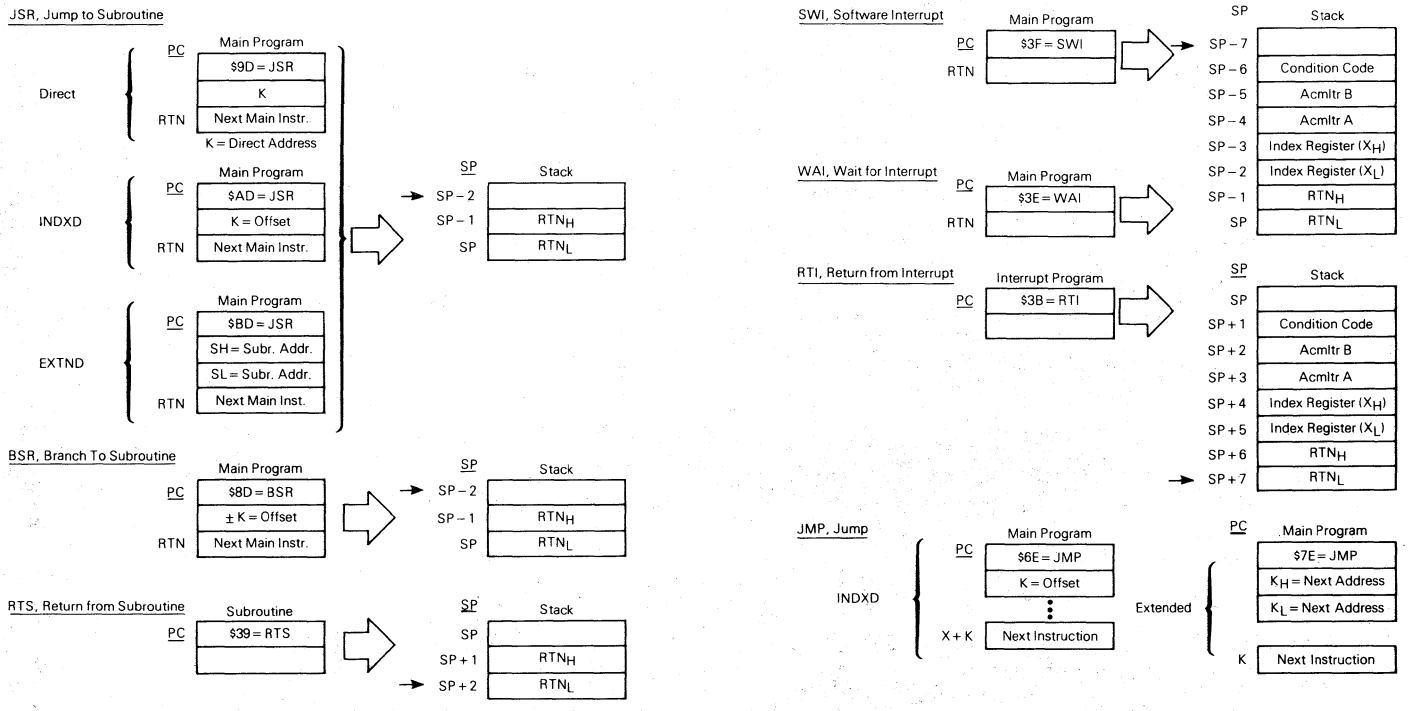
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Stack Pointer − 5 Stack Pointer − 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Buss FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

FIGURE 24 — SPECIAL OPERATIONS



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

APPENDIX
CUSTOM MC6801U4 ORDERING INFORMATION

A.1 CUSTOM MC6801U4 ORDERING INFORMATION

The custom MC6801U4 specifications may be transmitted to Motorola in any of the following media:

1) EPROMs
2) MDOS diskette

The specification should be formatted and packed, as indicated in the appropriate paragraph below, and mailed prepaid and insured with a cover letter (see Figure A-2) to:

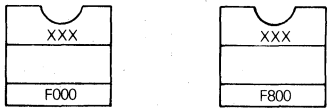
Motorola Inc.
3501 Ed Bluestein Blvd.
Austin, Texas 78721
Mail Drop L-13

A copy of the cover letter should also be mailed separately.

A.2 EPROMs

MCM2708 and MCM2716 type EPROMs, programmed with the custom program (positive logic sense for address and data), may be submitted for pattern generation. Both the MCM2708s and MCM2716s must be clearly marked to indicate which PROM corresponds to which address space ($F000-$FFFF). See Figure A-1 for recommended marking procedure.

FIGURE A-1



XXX = Customer ID

After the EPROMs are marked, they sould be placed in a conductive IC carrier and securely packed. Do not use styrofoam.

A.3 DISKETTE (MDOS)

The start/end location should be written on the label using EXORciser format.

MC6801U4L1 UNICORN Monitor

An MC6801U4 may be purchased without specifying the ROM pattern. This standard part is labeled as MC6801U4L1 and contains a 2K monitor (UNICORN) in the ROM. This monitor may be used to evaluate and debug a program under development. Details and a source listing are specified in the *UNICORN Monitor Reference Manual* M68UNICORN(D1).

EXORciser is a registered trademark of Motorola Inc.
UNICORN is a trademark of Motorola Inc.

FIGURE A-2

Customer Name _____

Address _____

State _____ City _____ Zip _____

Phone (____) _____ Extension _____

Contact Ms/Mr _____

Customer Part # _____

Package Type
☐ Ceramic
☐ Plastic

Marking
☐ Standard
☐ Special

Pattern Media
☐ 2708 EPROM
☐ 2716 EPROM
☐ Diskette (MDOS)

☐ OTHER (See Note) _____

NOTE: Other media require prior factory approval.

Signature _____

Title _____

MOTOROLA

Advance Information

MC6804J2

8-BIT MICROCOMPUTER

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

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC6804J2 microcomputer unit (MCU) is a member of the M6804 Family of very low-cost single-chip microcomputers. This 8-bit microcomputer 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.

1.2 FEATURES

The following are some of the hardware and software features of the MC6804J2 MCU.

HARDWARE FEATURES

- 5-Volt Single Supply
- 32 Bytes of RAM
- Memory Mapped I/O
- 1008 Bytes of Program ROM
- 64 Bytes of Data ROM
- 12 Bidirectional I/O Lines (Eight Lines with High Current Sink Capability)
- On-Chip Clock Generator
- Self-Test Mode
- Master Reset
- Complete Development System Support on EXORciser
- Software Programmable 8-Bit Timer Control Register and Timer Prescaler (7 Bits, 2^n)
- Timer Pin is Programmable as Input or Output
- On-Chip Circuit for ROM Verify

SOFTWARE FEATURES

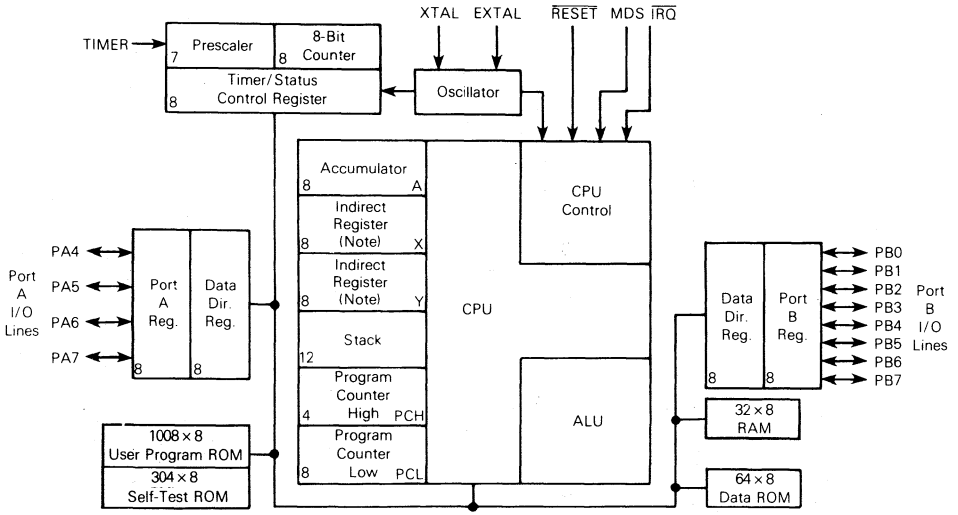
- Similar to M6805 HMOS Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction

SOFTWARE FEATURES (Continued)

- 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

- 12 Bidirectional I/O Lines with LSTTL, LSTTL/CMOS, or Open-Drain Interface
- Crystal or Low-Cost Resistor-Capacitor Oscillator
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin



NOTE: 8-Bit indirect registers X and Y, although shown as part of the CPU, are actually located in the 32 x 8 RAM at locations $80 and $81.

Figure 1-1. MC6804J2 MCU Block Diagram

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.

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

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 signals that a time out of the timer has occurred. 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 MC6804J2 to the beginning of a program. This pin, together with the MDS pin, is also used to select the operating mode of the MC6804J2. 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 MC6804J2. No special external diodes, switches, transistors, etc. are required in the MC6804J2.

2.1.7 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 **SECTION 5 INPUT/OUTPUT PORTS** for additional information.

2.2 MEMORY

The MCU operates in three different memory spaces: program space, data space, and stack space. A representation of these memory spaces is shown in Figure 2-1. The program space (Figure 2-1a) 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 (Figure 2-1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables and constants. The stack space (Figure 2-1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

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, 64 bytes ROM, 32 bytes RAM (which includes 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 304 bytes of self-test ROM, 1000 bytes program ROM, and eight bytes of vectors for self-test and user programs.

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 addresses, data, and control buses.

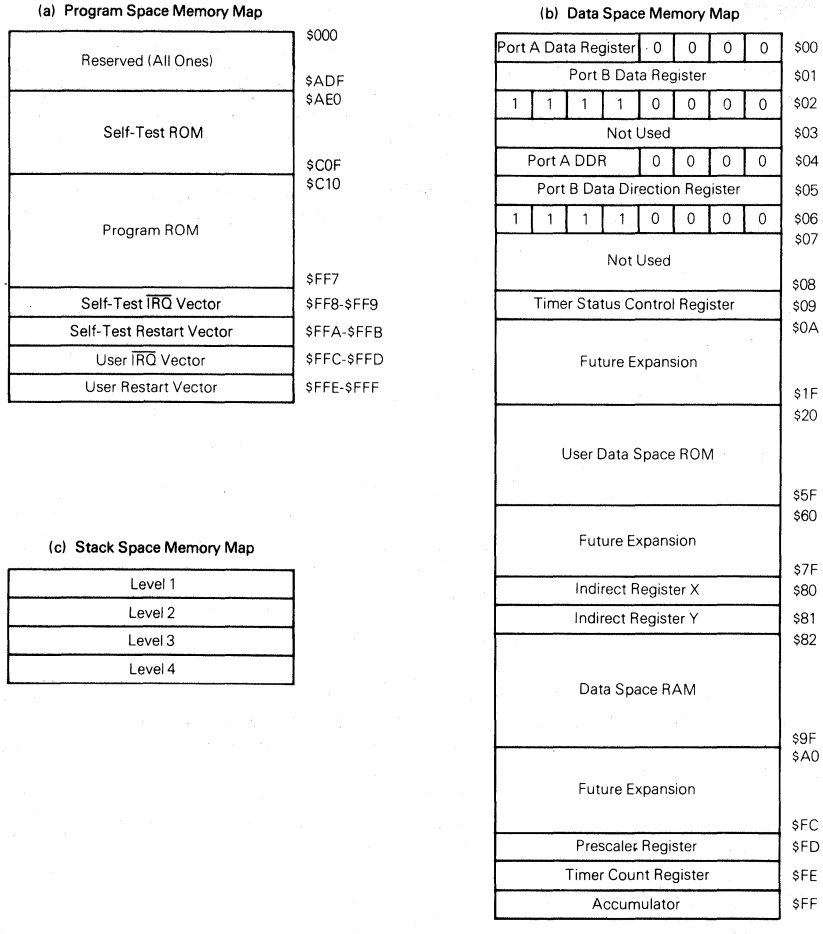


Figure 2-1. MC6804J2 MCU Address Map

2.4 REGISTERS

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

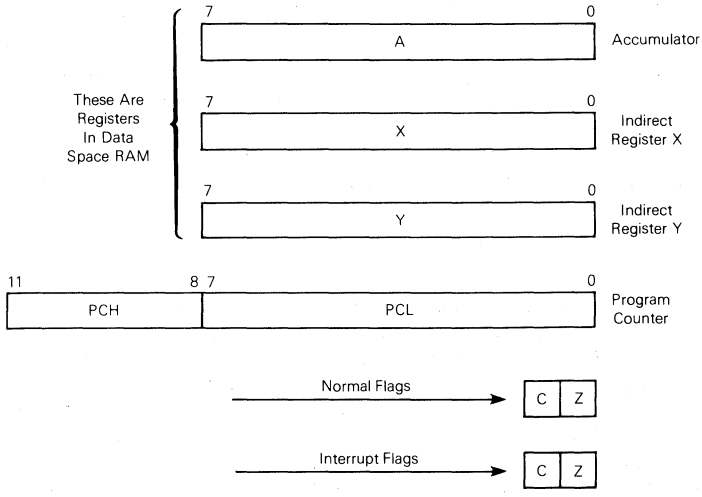


Figure 2-2. Programming Model

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 two of the 32 RAM locations ($80, $81) and as such generate implied instructions 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 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.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. 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. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5 Stack

There is a true LIFO stack incorporated in the MC6804J2 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in Figure 2-1c. 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 to the next 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 up 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 MC6804J2 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 loaded under program control, is decremented towards zero by a clock input (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) and prescaler are decremented on rising clock edges. 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

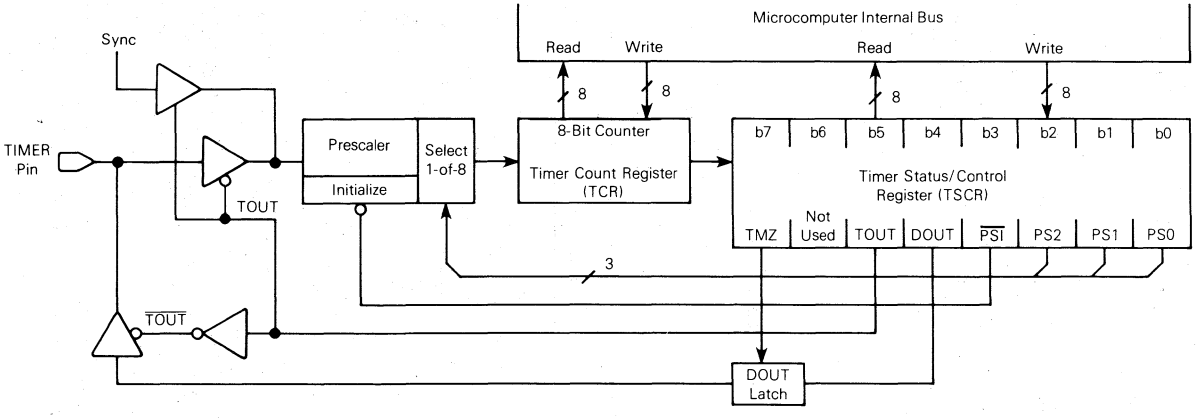
PS2	PS1	PS0	Divide By
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

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 latch the DOUT bit of the TSCR and provide it as output for 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

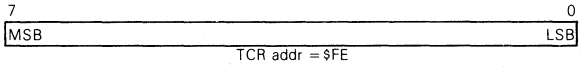
TOUT	Prescaler Clock	TIMER Pin
0	TIMER Pin	Input Mode
1	Sync	Output Mode

Figure 3-1. Timer Block Diagram

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.

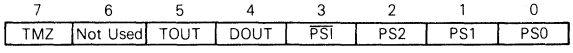
3.2 TIMER REGISTERS

3.2.1 Timer Count Register (TCR)



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

3.2.2 Timer Status/Control Register (TSCR)



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.

b6 — Not used.

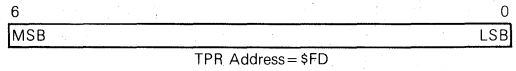
b5, TOUT — When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT — Data sent to the timer output pin when TMZ is set high (output mode only).

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 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 MC6804J2 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 (single chip mode= $FFC/$FFD, self-test mode= $FF8/$FF9) is loaded into the PC. 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. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialization routine as the first instruction in a program. The initialization routine

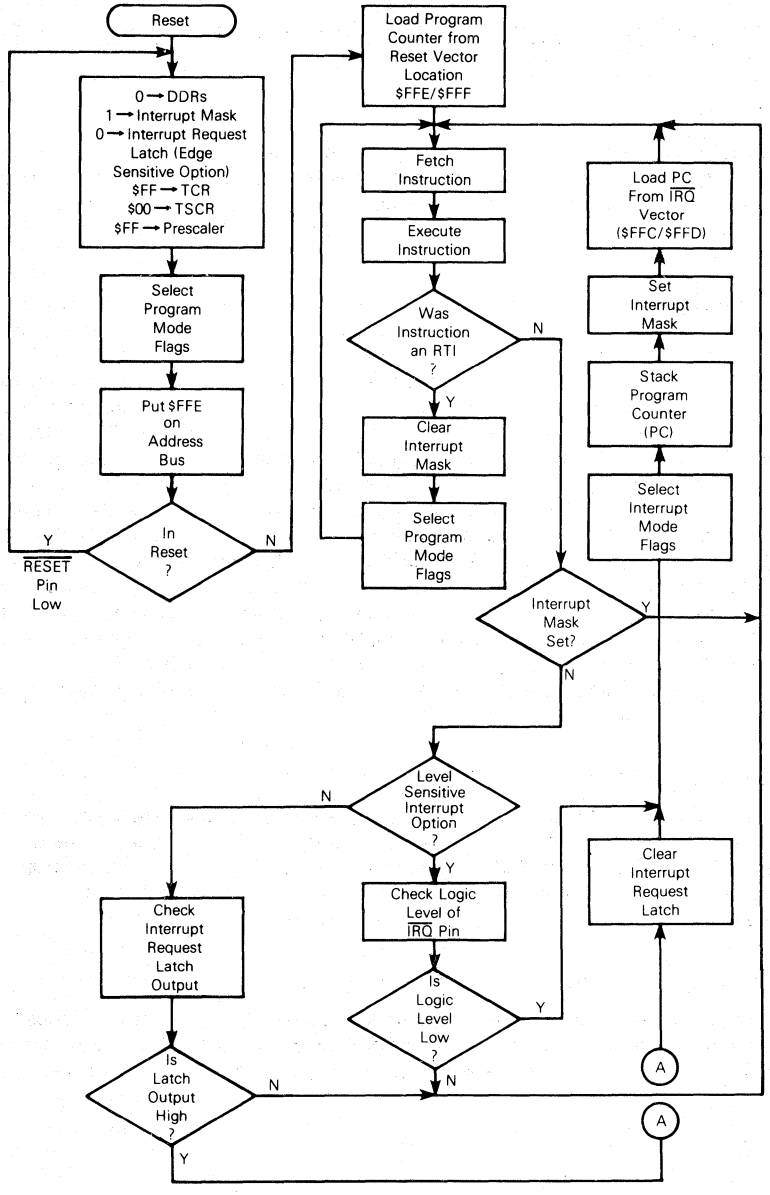


Figure 4-1. Reset and Interrupt Processing Flowchart

should end with an RTI (instead of RTS). 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)**).

4.2 SELF-TEST

The MC6804J2 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 pins 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 (LFSR) using the standard CCITT CRC16 polynomial. A schematic diagram of the self-test connections is shown in Figure 4-2. To perform a test of the MCU, connect it as shown in Figure 4-2a and monitor the LEDs for a 1100 ($0C) 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 4-2b. 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 MC6804J2. 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 circuit of Figure 4-2.

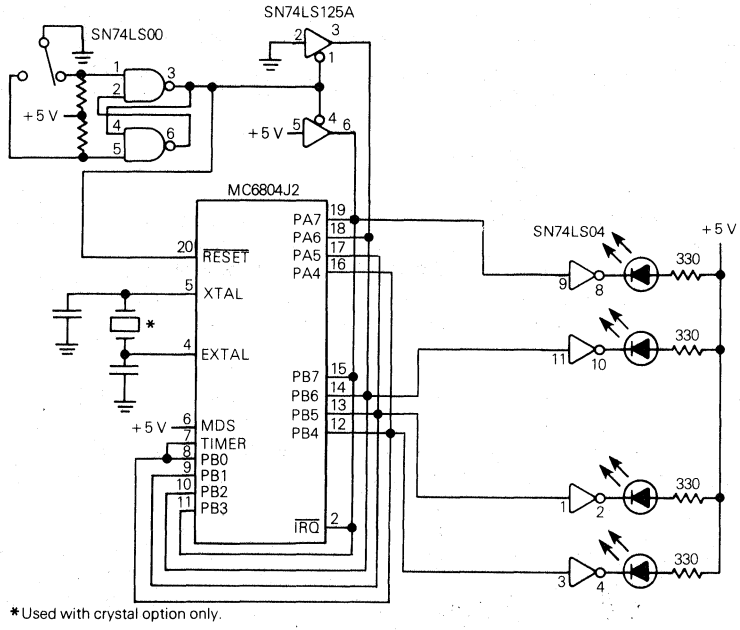
4.3 RESET

The MCU can be reset in two ways: by initial power up (see Figure 4-1) and by the external reset input ($\overline{RESET}$). During power up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the $\overline{RESET}$ input, as shown in Figure 4-3, typically provides sufficient delay.

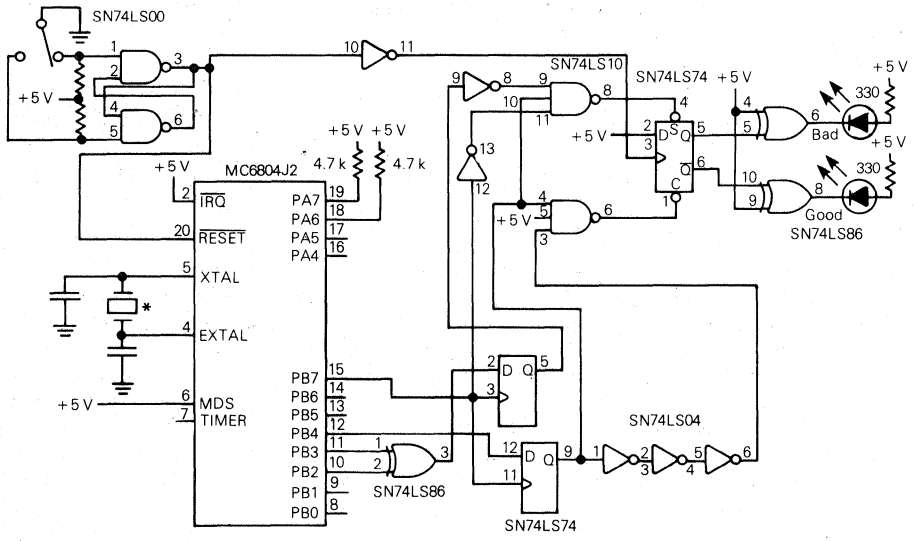
4.4 INTERNAL CLOCK GENERATOR OPTIONS

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 4-4, crystal specifications and suggested PC board layouts are given in Figure 4-5, resistor-capacitor selection graph is given in Figure 4-6, and a timing diagram is illustrated in Figure 4-7. 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 Test

Figure 4-2. Self-Test Circuit

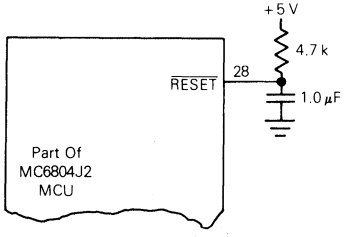
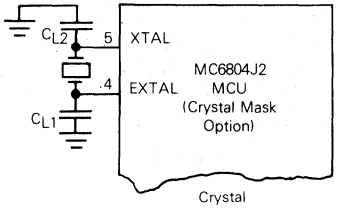


Figure 4-3. Power-Up Reset Delay Circuit



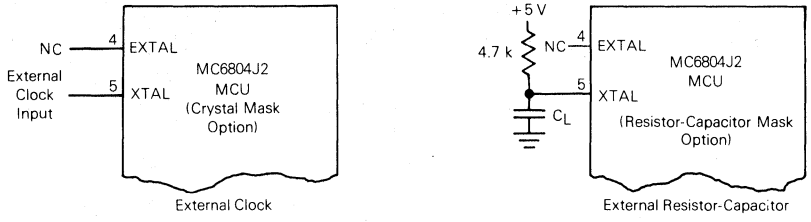


Figure 4-4. Clock Generator Options

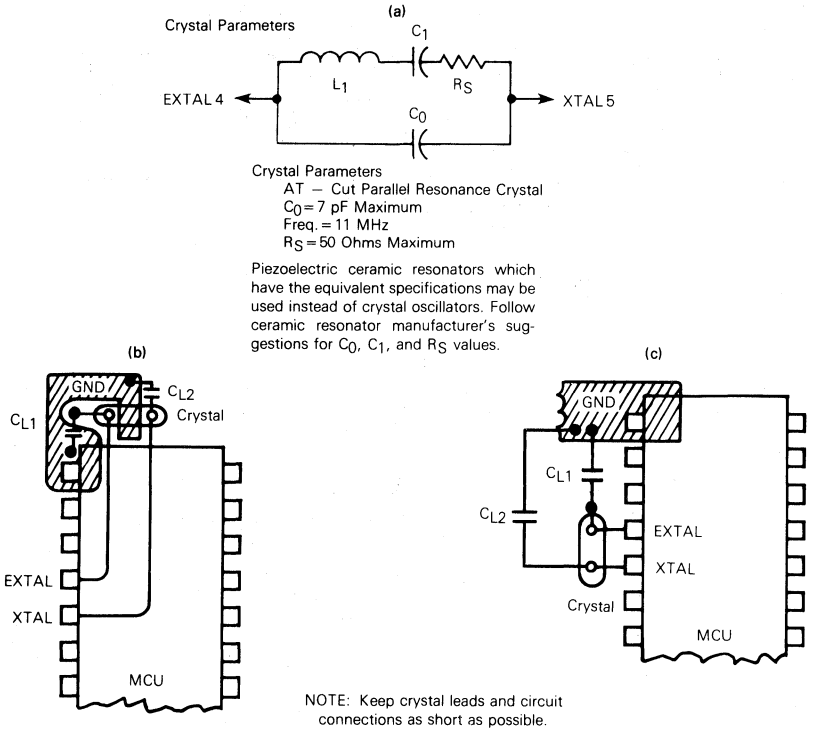


Figure 4-5. Crystal Motional Arm Parameters and Suggested PC Board Layout

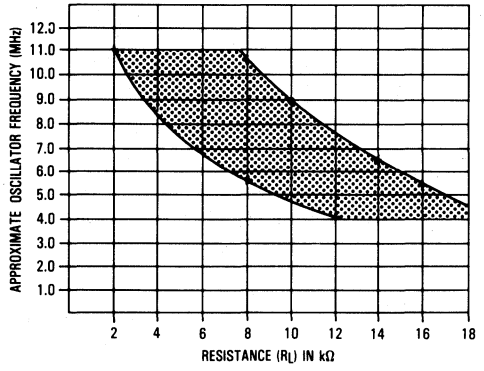


Figure 4-6. Typical Frequency Selection For Resistor-Capacitor Oscillator Option (C_L = 17 pF)

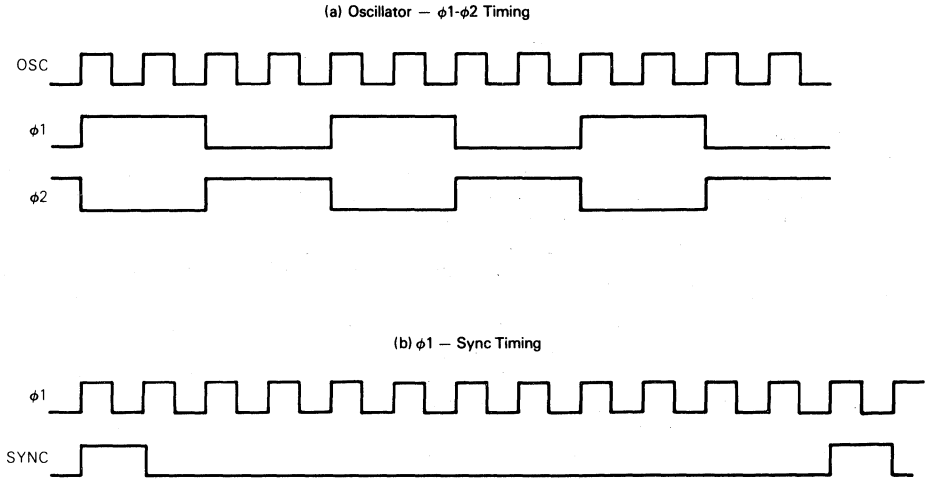
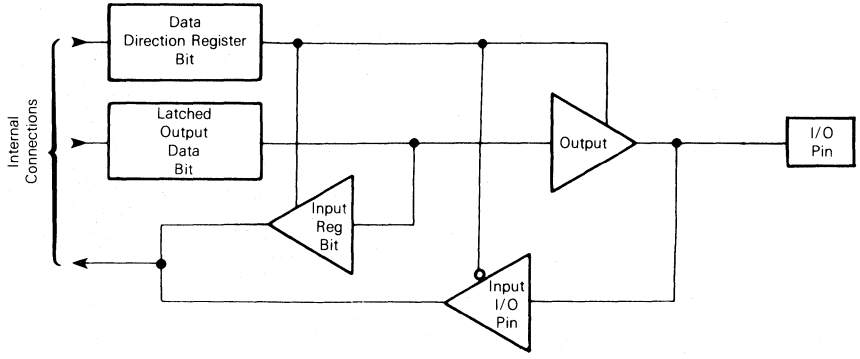


Figure 4-7. Clock Generator Timing Diagram

SECTION 5
INPUT/OUTPUT PORTS

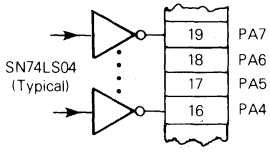
5.1 INPUT/OUTPUT

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). 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, both 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.

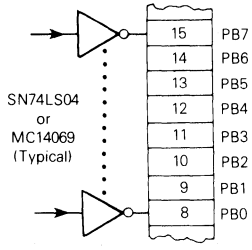


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

Figure 5-1. Typical I/O Port Circuitry

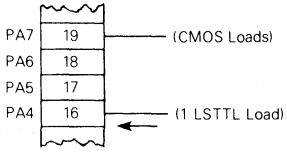


LSTTL Driving Port A Directly

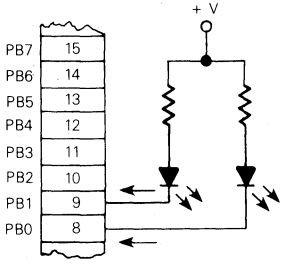


CMOS or LSTTL Driving Port B Directly

(a) Input Mode



Port A, bit 7 programmed as output, driving CMOS loads and bit 4 driving one LSTTL load directly (using CMOS output option).



Port B, bit 0, and bit 1 programmed as output, driving LEDs directly.

(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 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.

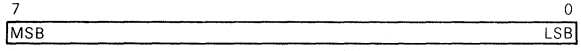
NOTE

The mask option only allows changes by port. For example, if the customer wishes PA7 to be open drain, then PA4-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.

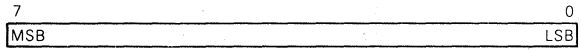
5.2.1 Port Data Register



Port A Address = $00
Port B Address = $01

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 Address = $04
Port B Address = $05

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.

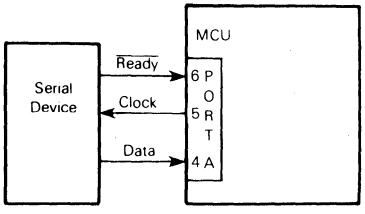
SECTION 6
SOFTWARE AND INSTRUCTION SET

6.1 SOFTWARE

6.1.1 Bit Manipulation

The MC6804J2 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit 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 is set to 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 RAM, ROM, or I/O allows 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   6,PORTA,SELF
                .
                .
                .
        BSET    5,PORTA
        BRCLR   4,PORTA,CONT
CONT    BCLR    5,PORTA
        ROLA    .
                .
                .
                .
```

Figure 6-1. Bit Manipulation Example

6.1.2 Addressing Modes

The MC6804J2 MCU has nine addressing modes which are explained briefly in the following paragraphs. The MC6804J2 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).

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.

6.1.2.3 SHORT DIRECT. The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the lower two bits of the opcode determine 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 or $81 respectively.)

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.

6.1.2.5 RELATIVE. The relative addressing mode is only used in conditional branch instructions. In relative addressing, the 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.

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.

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.

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.

6.2 INSTRUCTION SET

The MC6804J2 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.

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.

Table 6-1. Register/Memory Instructions

		Addressing Modes																			
		Indirect				Immediate			Direct			Inherent			Extended			Short-Direct			
Function	Mnem	Opcode XP	Opcode YP	# 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 XP from Memory	LDXI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Load YP 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 (TAR)	2	4	—	—	—	3
Jump Unconditional	JMP	—	—	—	—	—	—	—	—	—	—	—	—	—	9 (TAR)	2	4	—	—	—	3
Clear A	CLRA	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear XP	CLRX	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear YP	CLRY	—	—	—	—	—	—	—	FB	2	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.
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

		Addressing Modes										
		Indirect				Direct			Short-Direct			
		Opcode		#	#		#	#		#	#	Special
Function	Mnem	X	Y	Bytes	Cycles	Opcode	Bytes	Cycles	Opcode	Bytes	Cycles	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 XP	INCX	—	—	—	—	—	—	—	A8	1	4	—
Increment YP	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 XP	DECX	—	—	—	—	—	—	—	B8	1	4	—
Decrement YP	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.

Table 6-3. Branch Instructions

Function	Mnem	Relative Addressing Mode 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

Function	Mnem	Addressing Modes: Bit Set/Clear Opcode	# Bytes	# Cycles	Bit Test and Branch Opcode	# Bytes	# Cycles	Special Note
Branch IFF Bit n is set	BRSET n (n=0 . . . . 7)	–	–	–	C8+n	3	5	1
Branch IFF 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.

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

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

	Addressing Modes									Flags	
Mnemonic	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"									•	•
TAY	Assembler converts this to "STA $81"									•	•
TXA	Assembler converts this to "LDA $80"									•	•
TYA	Assembler converts this to "LDA $81"									•	•

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

Table 6-7. MC6804J2 Microcomputer 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

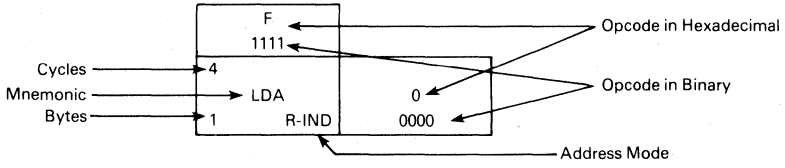
INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect

* 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 MC6804J2.

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 Range Plastic Ceramic Cerdip	T_J	 150 175 175	°C/W

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}).

7.3 THERMAL CHARACTERISTICS

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

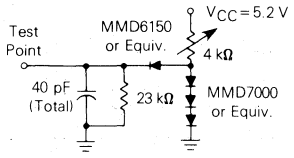


Figure 7-1. LSTTL Equivalent Test Load (Port B)

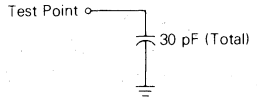


Figure 7-2. CMOS Equivalent Test Load (Ports A and B)

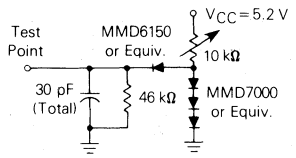


Figure 7-3. LSTTL Equivalent Test Load (Port A and TIMER)

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} = 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.

3

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

Characteristic	Symbol	Min	Typ	Max	Unit
Internal Power Dissipation – No Port Loading	P_{INT}	–	150	–	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}	$2xt_{byte}$	–	–	
$\overline{RESET}$ Pulse Width	t_{RWL}	$2xt_{byte}$	–	–	
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	100	–	–	ms

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
Timer and Port A (Standard)					
Output Low Voltage, I_{Load} = 0.4 mA	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = −50 μA	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
Port A (Open Drain)					
Output Low Voltage, I_{Load} = 0.4 mA	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
Port A (CMOS Drive)					
Output Low Voltage, I_{Load} = 0.4 mA (Sink)	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} − 1.0	–	–	V
Output High Voltage, I_{Load} = −100 μA	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	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	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	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	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} = −50 μA	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

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:

EPROM(s), MCM2716 or MCM2532
MDOS, disk file

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.

8.1.1 EPROMs

An MCM2716 or MCM2532 type EPROM, programmed with the customer program (positive logic sence for address and data), may be submitted for pattern generation. Since all program and data space information will fit on one MCM2716 or MCM2532 EPROM, the EPROM must be programmed as follows in order to emulate the MC6804J2 MCU. **For an MCM2716, start the data space ROM at EPROM address $020 and start program space ROM at EPROM address $410 and continue to memory space $7FF. Memory spaces $7F8 through $7FB are reserved for Motorola self-test vectors. For an MCM2532, the memory map shown in Figure 2-1 can be used. 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.

8.1.2 MDOS Disk File

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.**

8.2 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 contractural agreement for creation of the customer mask. If desired, Motorola will program a blank MCM2716, MCM2532, or MDOS disk (**supplied by the customer**) from the data file used to create the custom mask to aid in the verification process.

8.3 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 included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

8.4 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) from the M6804 cross assembler 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.

OPTION LIST

Select the options for the MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input

- ☐ Crystal
- ☐ Resistor-Capacitor

Interrupt Trigger

- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Output Drive (Select one Option per Port)

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

Customer Name_____

Address_____

City_____ State_____ Zip_____

Phone ( ) _____ Extension_____

Contact Ms/Mr_____

Customer Part Number_____

Pattern Media

- ☐ MCM2532 EPROM
- ☐ MCM2716 EPROM
- ☐ MDOS Disk File
- ☐ (Note)_____

Note: Other Media require prior factory approval.

Signature_____

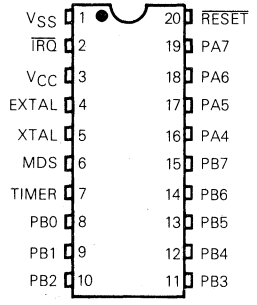
Title_____

Figure 8.1 Ordering Form

SECTION 9
MECHANICAL DATA

This section contains the pin assignment and package dimension diagrams for the MC6804J2 microcomputer.

9.1 PIN ASSIGNMENT



MOTOROLA

Advance Information

MC6804P2

8-BIT MICROCOMPUTER

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

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC6804P2 microcomputer unit (MCU) is a member of the M6804 Family of very low-cost single-chip microcomputers. This 8-bit microcomputer 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.

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 the MC6805P2 and MC68705P3
- 32 Bytes of RAM
- Memory Mapped I/O
- 1024 Bytes of Program ROM
- 64 Bytes of Data ROM
- 20 Bidirectional I/O Lines (Eight Lines with High Current Sink Capability)
- On-Chip Clock Generator
- Self-Test Mode
- Master Reset
- Complete Development System Support on EXORciser
- Software Programmable 8-Bit Timer Control Register and Timer Prescaler (7 Bits, 2^n)
- Timer Pin is Programmable as Input or Output
- On-Chip Circuit for ROM Verify

SOFTWARE FEATURES

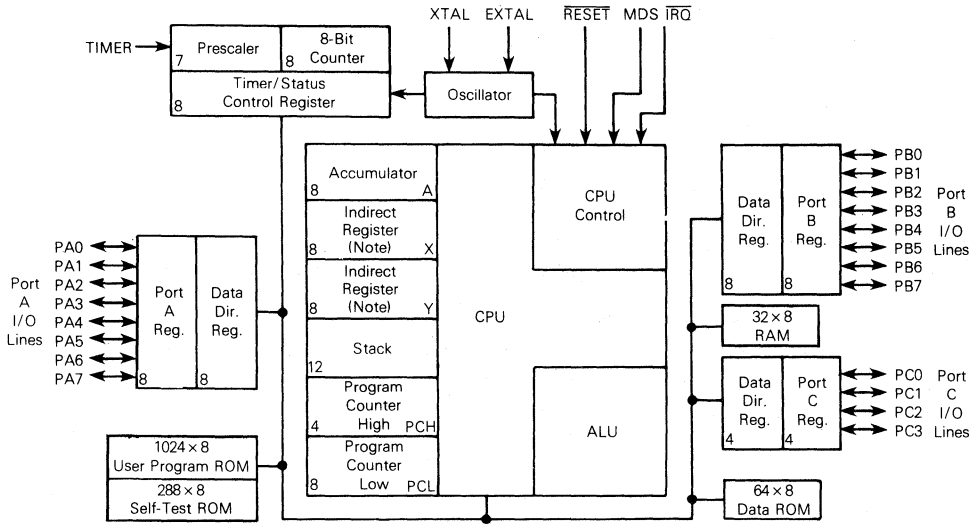
- Similar to M6805 HMOS Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction

SOFTWARE FEATURES (Continued)

- 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
- Crystal or Low-Cost Resistor-Capacitor Oscillator
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin



NOTE: 8-Bit indirect registers X and Y, although shown as part of the CPU, are actually located in the 32 x 8 RAM at locations $80 and $81.

Figure 1-1. MC6804P2 MCU Block Diagram

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.

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

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 signals that a time out of the timer has occurred. 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. 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 MC6804P2.

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 operates in three different memory spaces: program space, data space, and stack space. A representation of these memory spaces is shown in Figure 2-1. The program space (Figure 2-1a) 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 (Figure 2-1b) contains all of the RAM locations, plus I/O locations and some ROM used for storage of tables and constants. The stack space (Figure 2-1c) contains RAM which is used for stacking subroutine and interrupt return addresses.

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, 64 bytes ROM, 32 bytes RAM (which includes 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 including 288 bytes of self-test ROM, 1016 bytes program ROM, and eight bytes of vectors for self-test and user programs.

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 addresses, data, and control buses.

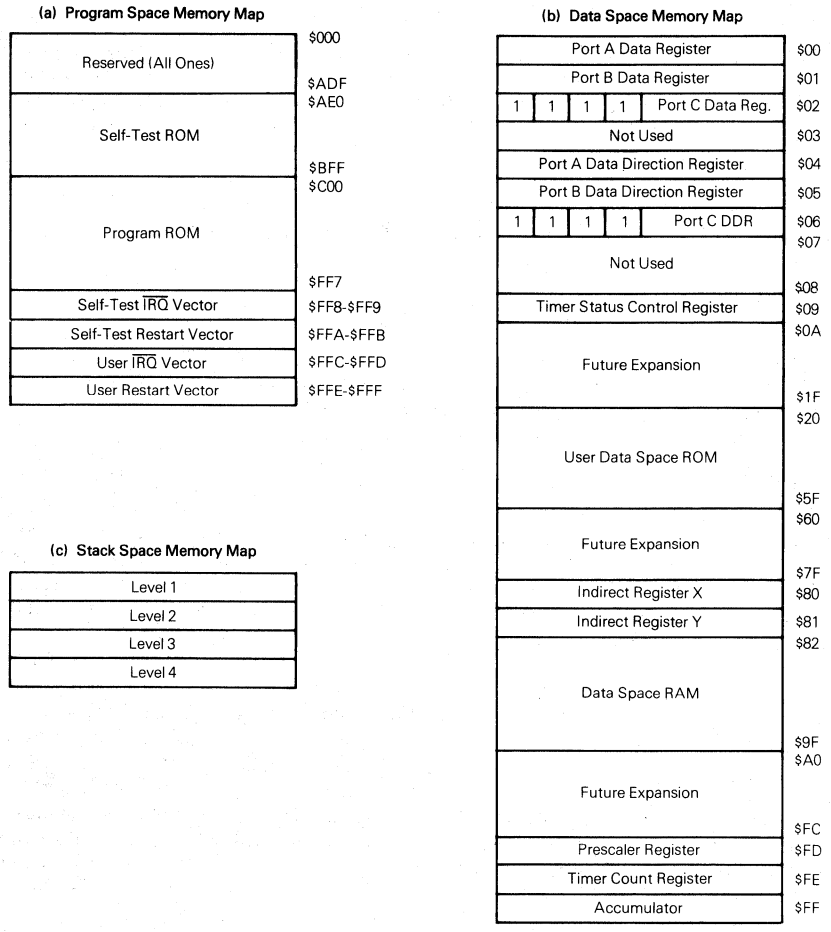


Figure 2-1. MC6804P2 MCU Address Map

2.4 REGISTERS

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

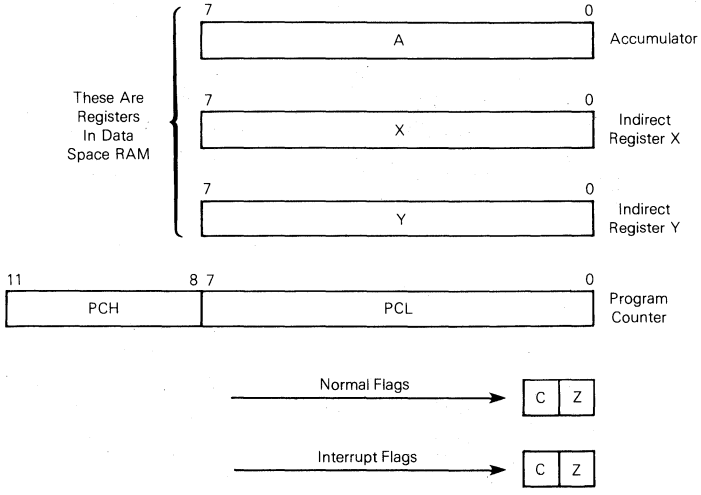


Figure 2-2. Programming Model

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 two of the 32 RAM locations ($80, $81) and as such generate implied instructions 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 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.

There are two sets of these flags, one set is for interrupt processing, the other for all other routines. 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. Further, the interrupt flags will not be cleared upon entering the interrupt mode. Instead, the flags will be as they were at the exit of the last interrupt mode. Both sets of flags are cleared by reset.

2.4.5 Stack

There is a true LIFO stack incorporated in the MC6804P2 which eliminates the need for a stack pointer. Stack space is implemented in separate RAM (12-bits wide) shown in Figure 2-1c. 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 to the next 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 up 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 loaded under program control, is decremented towards zero by a clock input (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) and prescaler are decremented on rising clock edges. 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

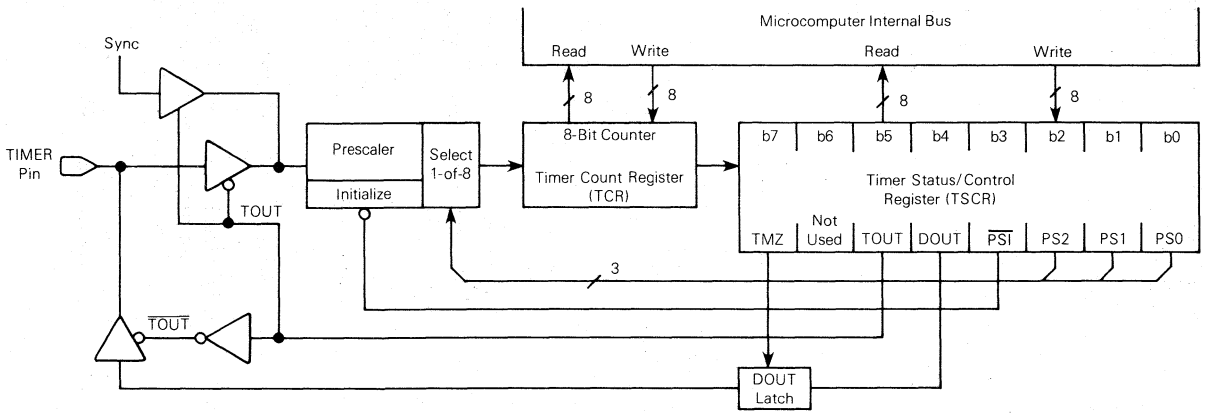
PS2	PS1	PS0	Divide By
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

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 latch the DOUT bit of the TSCR and provide it as output for 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

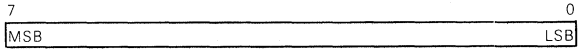
TOUT	Prescaler Clock	TIMER Pin
0	TIMER Pin	Input Mode
1	Sync	Output Mode

Figure 3-1. Timer Block Diagram

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.

b6 — Not used.

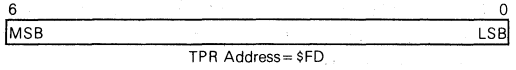
b5, TOUT — When low, this bit selects the input mode for the timer. When high, the output mode is selected.

b4, DOUT — Data sent to the timer output pin when TMZ is set high (output mode only).

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 (single chip mode= $FFC/$FFD, self-test mode= $FF8/$FF9) is loaded into the PC. 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. To clear the interrupt mask, the programmer should write a JSR (instead of a JMP) instruction to an initialization routine as the first instruction in a program. The initialization routine

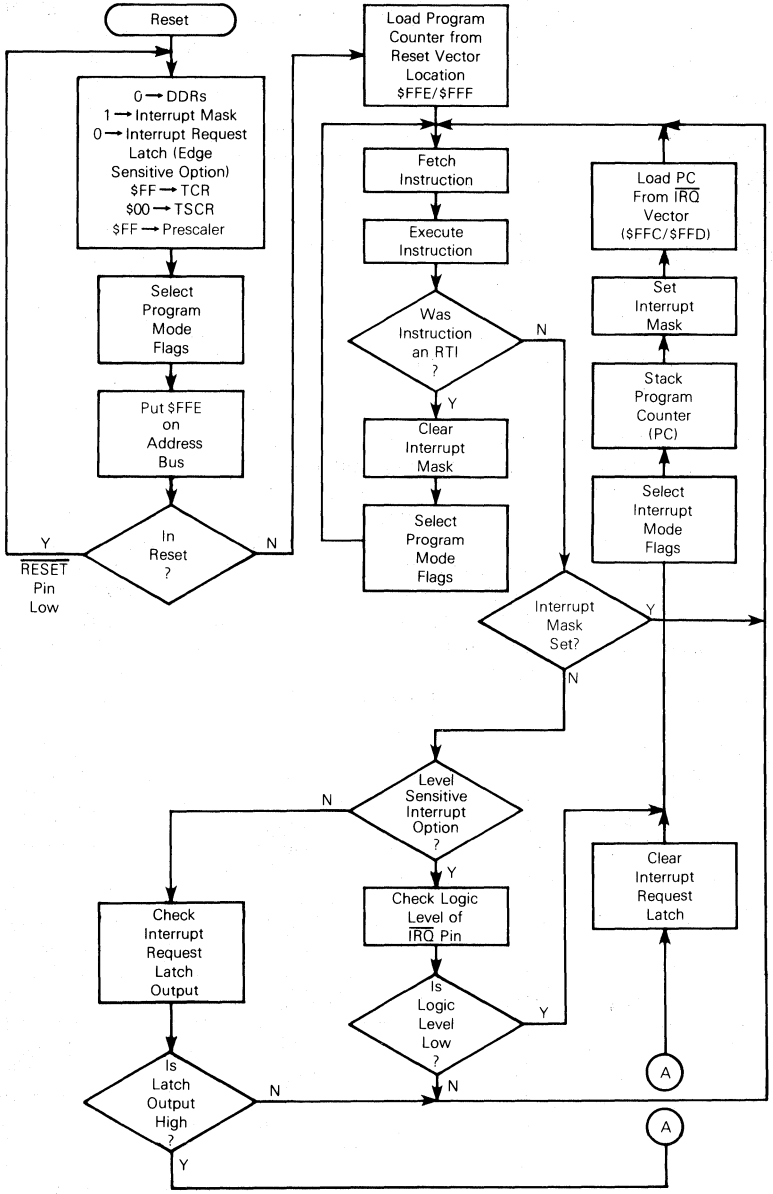


Figure 4-1. Reset and Interrupt Processing Flowchart

should end with an RTI (instead of RTS). 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)**).

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 pins 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 (LFSR) using the standard CCITT CRC16 polynomial. A schematic diagram of the self-test connections is shown in Figure 4-2. To perform a test of the MCU, connect it as shown in Figure 4-2a 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 4-2b. 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 circuit of Figure 4-2.

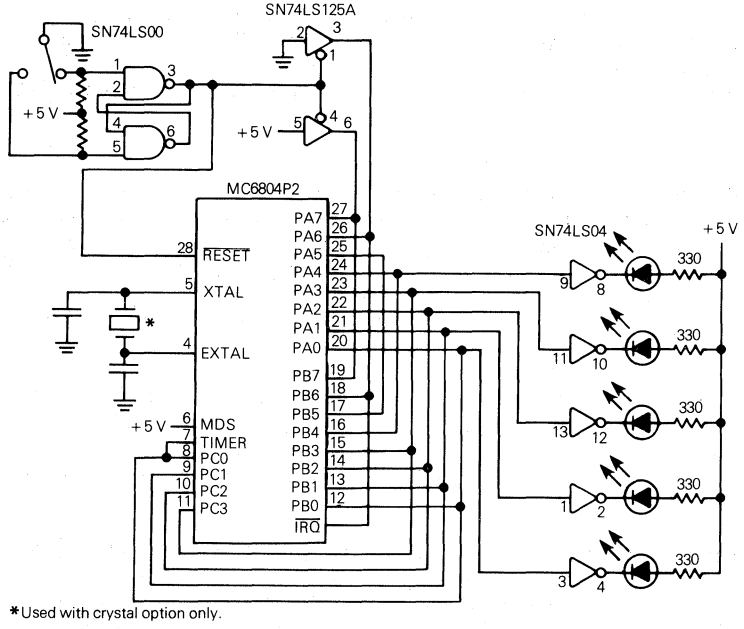
4.3 RESET

The MCU can be reset in two ways: by initial power up (see Figure 4-1) and by the external reset input ($\overline{RESET}$). During power up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time delay allows the internal clock generator to stabilize. Connecting a capacitor and resistor to the $\overline{RESET}$ input, as shown in Figure 4-3, typically provides sufficient delay.

4.4 INTERNAL CLOCK GENERATOR OPTIONS

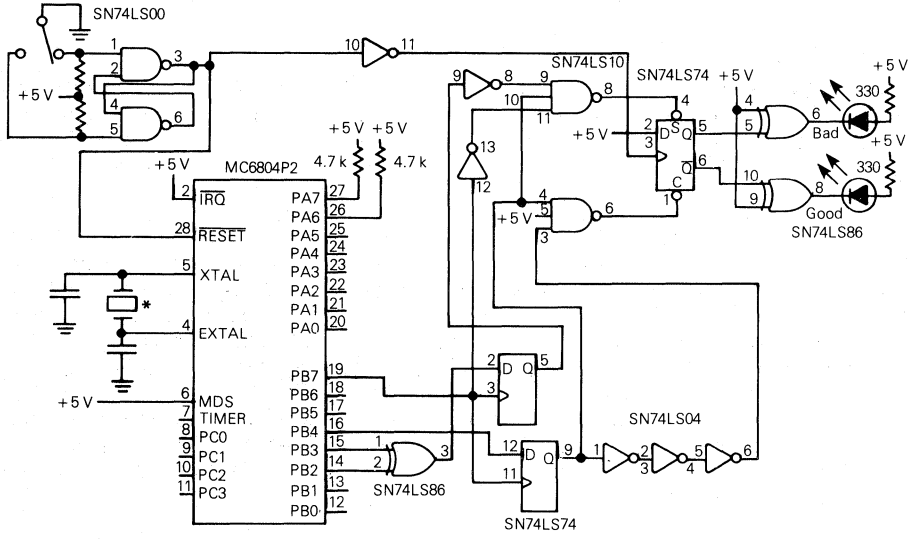
The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor-capacitor, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different clock generator option connection methods are shown in Figure 4-4, crystal specifications and suggested PC board layouts are given in Figure 4-5, resistor-capacitor selection graph is given in Figure 4-6, and a timing diagram is illustrated in Figure 4-7. 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.



*Used with crystal option only.

(a) Functional Test



*Used with crystal option only.

(b) Simple ROM Verify Test

Figure 4-2. Self-Test Circuit

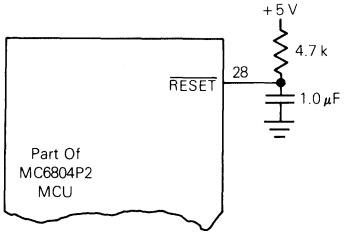


Figure 4-3. Power-Up Reset Delay Circuit

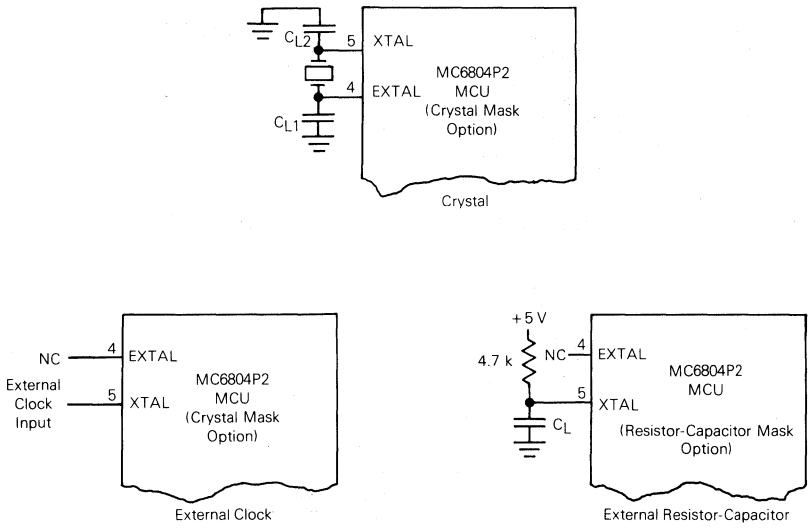


Figure 4-4. Clock Generator Options

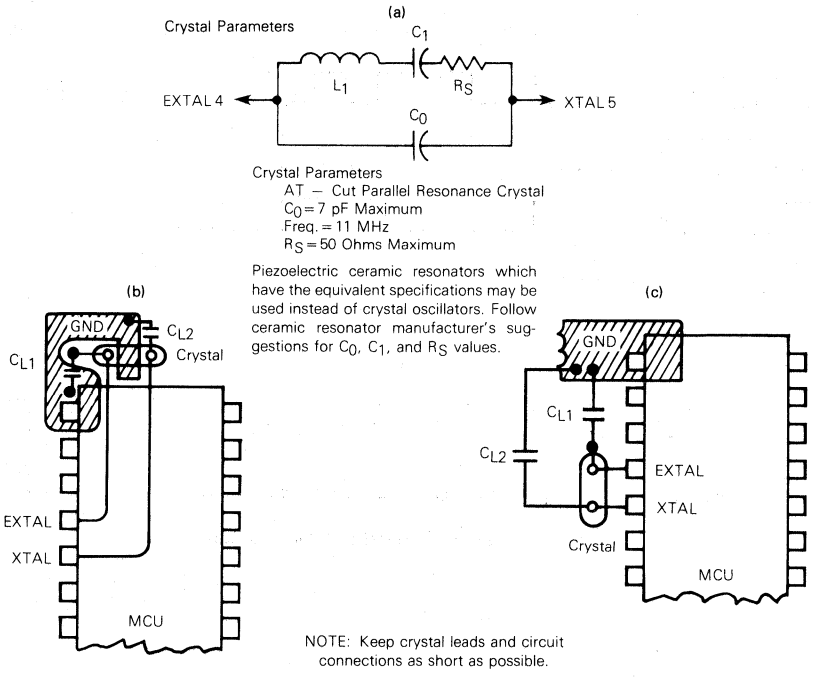


Figure 4-5. Crystal Motional Arm Parameters and Suggested PC Board Layout

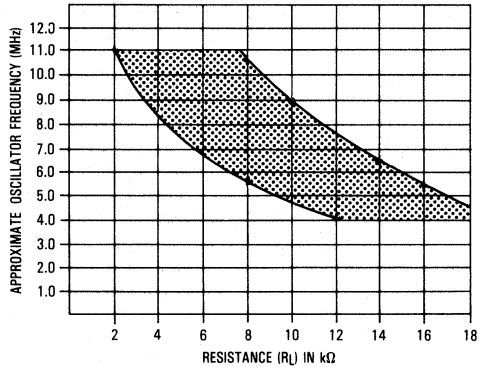


Figure 4-6. Typical Frequency Selection For Resistor-Capacitor Oscillator Option (C_L = 17 pF)

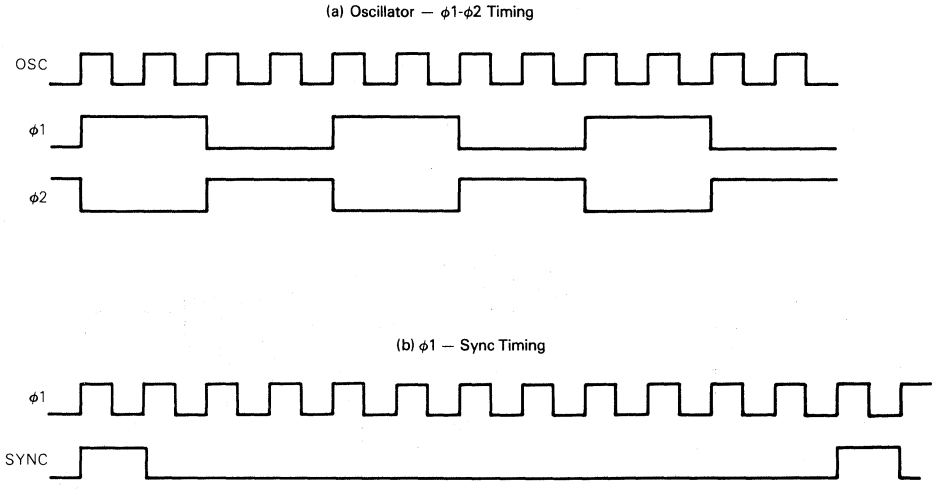
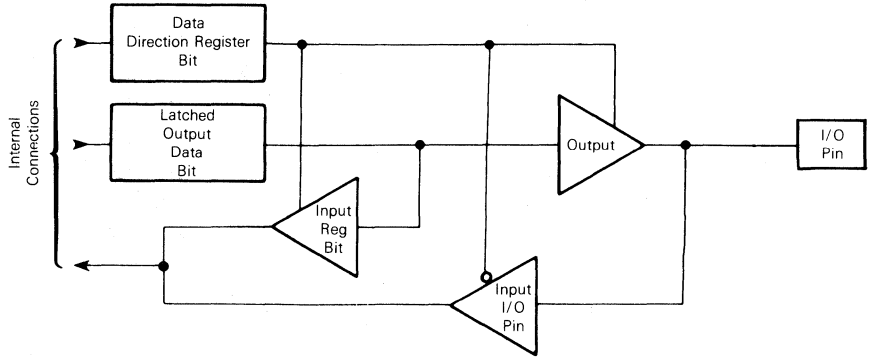


Figure 4-7. Clock Generator Timing Diagram

SECTION 5
INPUT/OUTPUT PORTS

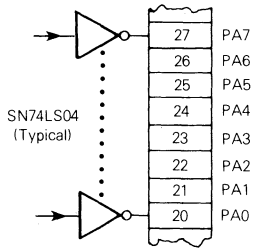
5.1 INPUT/OUTPUT

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.

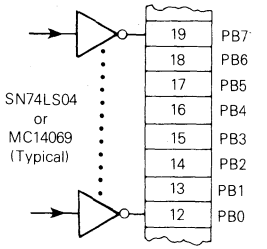


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z	Pin

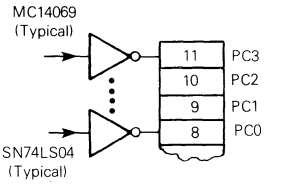
Figure 5-1. Typical I/O Port Circuitry



LSTTL Driving Port A Directly

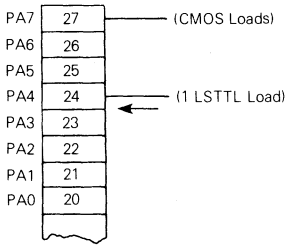


CMOS or LSTTL Driving Port B Directly

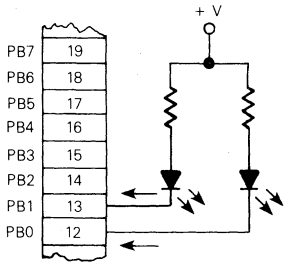


CMOS and LSTTL Driving Port C Directly

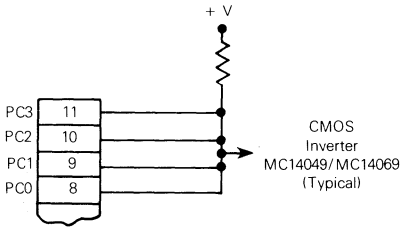
(a) Input Mode



Port A, bit 7 programmed as output, driving CMOS loads and bit 4 driving one LSTTL load directly (using CMOS output option).



Port B, bit 0, and bit 1 programmed as output, driving LEDs directly.



Port C open drain option, with bits 0-3 programmed as output, driving CMOS load via wired-ORed configuration.

(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.

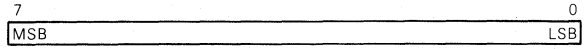
NOTE

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.

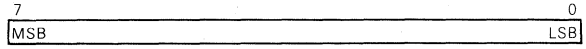
5.2.1 Port Data Register



Port A Address = $00
Port B Address = $01
Port C Address = $02 (Bits 0-3)

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 Address = $04
Port B Address = $05
Port C Address = $06 (Bits 0-3)

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.

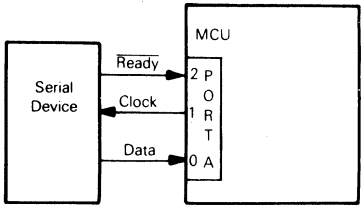
SECTION 6
SOFTWARE AND INSTRUCTION SET

6.1 SOFTWARE

6.1.1 Bit Manipulation

The MC6804P2 MCU has the ability to set or clear any register or single random access memory (RAM) writable bit 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 is set to 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 RAM, ROM, or I/O allows 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).

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.

6.1.2.3 SHORT DIRECT. The MCU also has four locations in data space RAM ($80, $81, $82, $83) which may be used in a short-direct addressing mode. In this mode the lower two bits of 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.)

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.

6.1.2.5 RELATIVE. The relative addressing mode is only used in conditional branch instructions. In relative addressing, the 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.

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.

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.

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.

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.

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.

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 XP from Memory	LDXI	—	—	—	—	B0	3	4	—	—	—	—	—	—	—	—	—	—	—	—	4
Load YP 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 (TAR)	2	4	—	—	—	3
Jump Unconditional	JMP	—	—	—	—	—	—	—	—	—	—	—	—	—	9 (TAR)	2	4	—	—	—	3
Clear A	CLRA	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear XP	CLRX	—	—	—	—	—	—	—	FB	2	4	—	—	—	—	—	—	—	—	—	—
Clear YP	CLRY	—	—	—	—	—	—	—	FB	2	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.
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

		Addressing Modes										
		Indirect				Direct			Short-Direct			
		Opcode		#	#		#	#		#	#	Special
Function	Mnem	X	Y	Bytes	Cycles	Opcode	Bytes	Cycles	Opcode	Bytes	Cycles	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.

Table 6-3. Branch Instructions

Function	Mnem	Relative Addressing Mode 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

Function	Mnem	Addressing Modes: Bit Set/Clear Opcode	# Bytes	# Cycles	Bit Test and Branch Opcode	# Bytes	# Cycles	Special Note
Branch IFF Bit n is set	BRSET n (n=0 . . . . 7)	—	—	—	C8+n	3	5	1
Branch IFF 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.

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

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.

3

Table 6-6. Instruction Set

	Addressing Modes									Flags	
Mnemonic	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"									•	•
TAY	Assembler converts this to "STA $81"									•	•
TXA	Assembler converts this to "LDA $80"									•	•
TYA	Assembler converts this to "LDA $81"									•	•

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

Table 6-7. MC6804P2 Microcomputer

	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

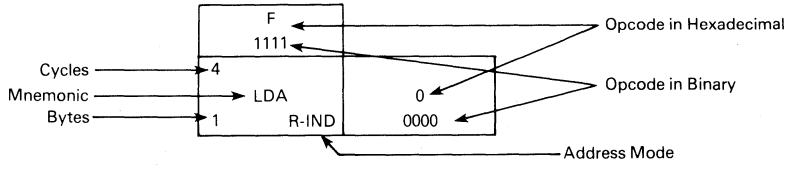
INH	Inherent
S-D	Short Direct
B-T-B	Bit Test and Branch
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
R-IND	Register Indirect

*	Indicates Instruction Reserved for Future Use
#	Indicates Illegal Instruction

Instruction Set Opcode Map

	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 Range			
Plastic	T_J	150	°C/W
Ceramic		175	
Cerdip		175	

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}).

7.3 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	70	°C/W
Ceramic		50	
Cerdip		60	

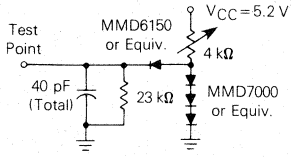


Figure 7-1. LSTTL Equivalent Test Load (Port B)

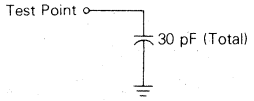


Figure 7-2. CMOS Equivalent Test Load (Ports A, B, C)

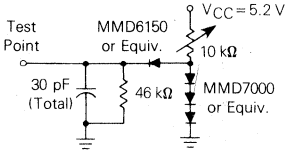


Figure 7-3. LSTTL Equivalent Test Load (Ports A, C, and TIMER)

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} = 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.

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

Characteristic	Symbol	Min	Typ	Max	Unit
Internal Power Dissipation – No Port Loading	P_{INT}	–	150	–	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}	$2xt_{byte}$	–	–	–
$\overline{RESET}$ Pulse Width	t_{RWL}	$2xt_{byte}$	–	–	–
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	100	–	–	ms

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 and C (Standard)					
Output Low Voltage, I_{Load} = 0.4 mA	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = – 50 μA	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	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)	V_{OL}	–	–	0.5	V
Output High Voltage, I_{Load} = – 10 μA	V_{OH}	V_{CC} – 1.0	–	–	V
Output High Voltage, I_{Load} = – 100 μA	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	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	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	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	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} = – 50 μA	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

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:

EPROM(s), MCM2716 or MCM2532
MDOS, disk file

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.

8.1.1 EPROMs

An MCM2716 or MCM2532 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 MCM2716 or MCM2532 EPROM, the EPROM must be programmed as follows in order to emulate the MC6804P2 MCU. **For an MCM2716, start the data space ROM at EPROM address $020 and start program space ROM at EPROM address $400 and continue to memory space $7FF. Memory space $7F8 through $7FB are reserved for Motorola self-test vectors. For an MCM2532, the memory map shown in Figure 2-1 can be used. 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.

8.1.2 MDOS Disk File

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.**

8.2 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 contractural agreement for creation of the customer mask. If desired, Motorola will program a blank MCM2716, MCM2532, or MDOS disk (**supplied by the customer**) from the data file used to create the custom mask to aid in the verification process.

8.3 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 included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

8.4 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) from the M6805 cross assembler 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.

OPTION LIST

Select the options for the MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input

- ☐ Crystal
- ☐ Resistor-Capacitor

Interrupt Trigger

- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Output Drive (Select one Option per Port)

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

Customer Name_____

Address_____

City_____ State_____ Zip_____

Phone ( )_____ Extension_____

Contact Ms/Mr _____

Customer Part Number_____

Pattern Media

- ☐ MCM2532 EPROM
- ☐ MCM2716 EPROM
- ☐ MDOS Disk File
- ☐ (Note)_____

Note: Other Media require prior factory approval.

Signature_____

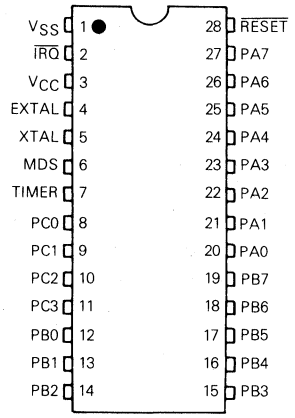
Title_____

Figure 8.1 Ordering Form

SECTION 9
MECHANICAL DATA

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

9.1 PIN ASSIGNMENT



MOTOROLA

MC68HC04P2
MC68HC04P3

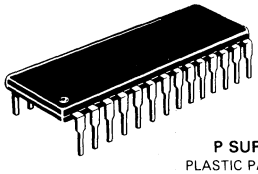
Product Preview

HCMOS
(HIGH-DENSITY CMOS
SILICON-GATE)

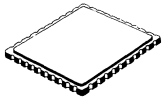
8-BIT HCMOS MICROCOMPUTERS

8-BIT HCMOS MICROCOMPUTER UNITS

The MC68HC04P2 and MC68HC04P3 HCMOS microcomputer units (MCUs) are members of the M68HC04 Family of very low-cost single-chip microcomputers. These 8-bit microcomputers contain a CPU, on-chip CLOCK, ROM, RAM, I/O, and TIMER. They are designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set. The following are some of the hardware and software highlights of the MC68HC04P2 and MC68HC04P3 microcomputers.

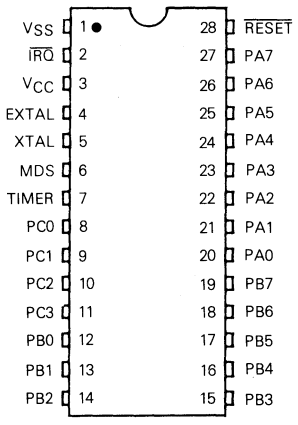


P SUFFIX
PLASTIC PACKAGE
CASE 710



Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT



HARDWARE FEATURES

- Low Power HCMOS
- Power Saving Stop and Wait Modes
- 8-Bit Architecture
- MC68HC04P2 and MC68HC04P3 are Pin Compatible With the MC6804P2
- RAM: MC68HC04P2 — 32 Bytes
  MC68HC04P3 — 128 Bytes
- Memory Mapped I/O
- User ROM: MC68HC04P2 — 1024 Bytes
  MC68HC04P3 — 2048 Bytes
- 72 Bytes of ROM for Look-Up Tables
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (Eight Lines are LED Compatible)
- On-Chip Clock Generator
- Self-Check Mode
- Master Reset
- Complete Development System Support on EXORciser
- Software Programmable Timer Prescaler
- 5 Volt Single Supply

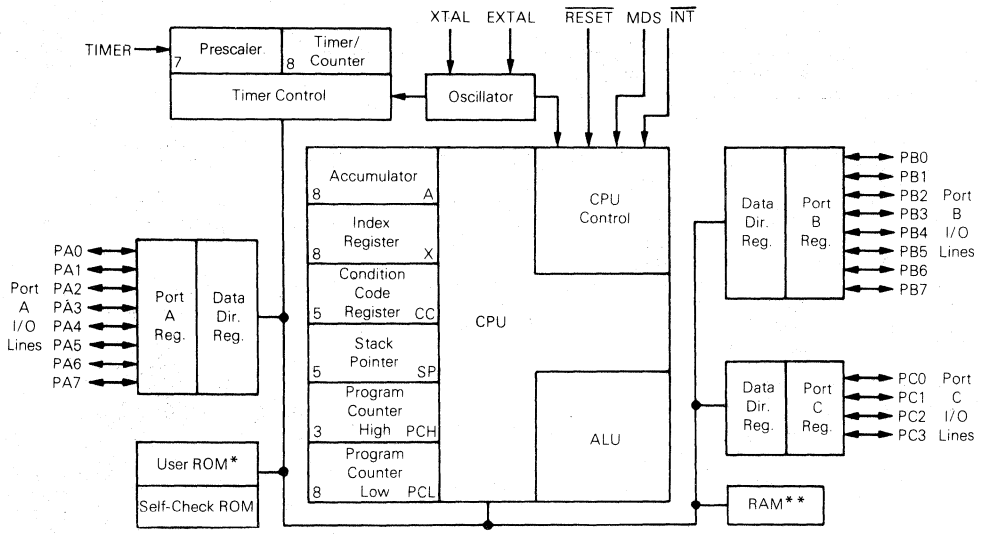
SOFTWARE FEATURES

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Versatile Interrupt Handling
- Versatile Indirect Registers
- Conditional Branches
- Single Instruction Memory Examine/Change
- Timer Pin is Software Programmable as Clock Input or Timer Input
- 10 Powerful Addressing Modes

USER SELECTABLE OPTIONS

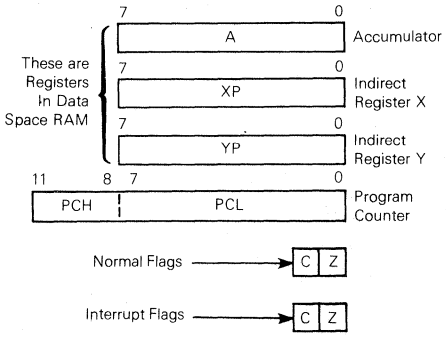
- 20 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Vectored Interrupts: Timer, Software, and External
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin

BLOCK DIAGRAM



*User ROM area: MC68HC04P2 = 1024 × 8 — MC68HC04P3 = 2048 × 8
**RAM area: MC68HC04P2 = 32 × 8 — MC68HC04P3 = 128 × 8

PROGRAMMING MODEL



MOTOROLA

MC6805K2
MC6805K3

Product Preview

8-BIT MICROCOMPUTER UNITS WITH SERIAL PERIPHERAL INTERFACE AND TWO TIMERS

The MC6805K2/MC6805K3 microcomputer units are members of the M6805 Family of low-cost single-chip microcomputers. These 8-bit microcomputers contain a CPU, on-chip clock, ROM, EEPROM, RAM, I/O, two timers, one programmable prescaler, and a serial peripheral interface. These units are designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set.

HARDWARE FEATURES

- 32 Bidirectional TTL I/O Lines
  - Eight CMOS I/O Compatible
  - Eight LED I/O Compatible
  - Eight Open Drain (Software Control)
- User ROM: MC6805K2 — 2K Bytes
  MC6805K3 — 3.6K Bytes
- 96 Bytes of User RAM, 16 Bytes on Standby via V_{STBY} Pin
- 128 Bytes of User EEPROM with Write/Erase Latches
- Self-Check Mode
- Serial Peripheral Interface
- Zero-Crossing Detect/Interrupt
- Two Cascadable 8-Bit Timers with 7-Bit Software Programmable Prescaler, Data Modulus Latch, and Capture Latch
- Auxiliary Counter with "Watchdog" Reset Feature
- 5-Volt Single Supply
- Two External Interrupts

SOFTWARE FEATURES

- 10 Powerful Addressing Modes
- Byte Efficient Instruction Set with True Bit Manipulation, Bit Test, and Branch Instructions
- Single Instruction Memory Examine/Change
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- User Callable Self-Check Subroutines
- Complete Development System Support on EXORciser, EXORset, and HDS-200

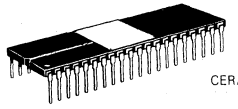
USER SELECTABLE OPTIONS

- Eight Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer/SPI, Software, and External
- Eight Byte Standby RAM Option

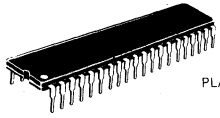
HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT MICROCOMPUTERS



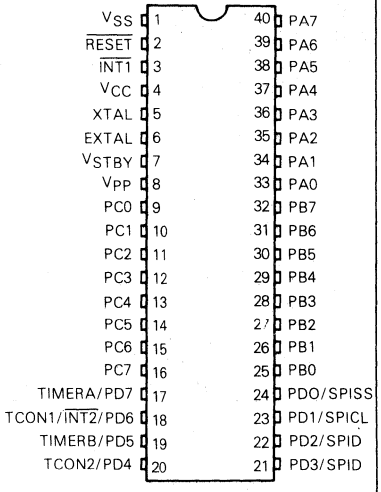
L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERPID PACKAGE
CASE 734

PIN ASSIGNMENT



3

FIGURE 1 — BLOCK DIAGRAM

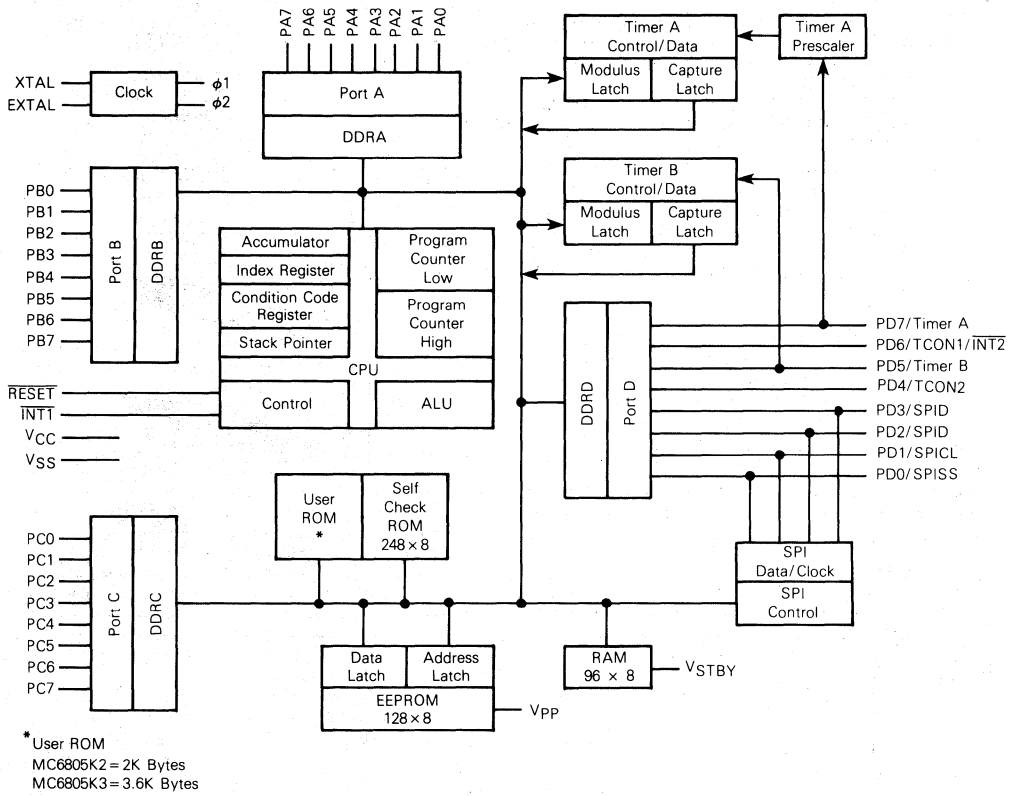
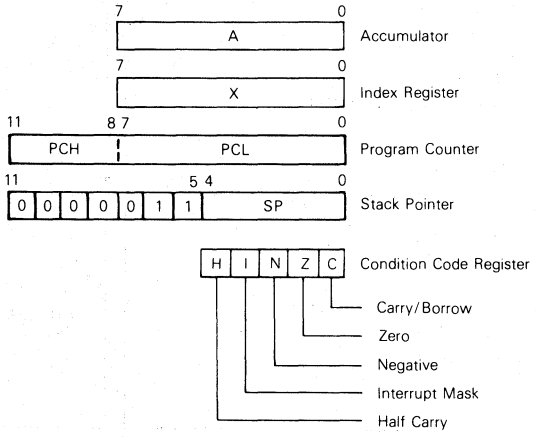


FIGURE 2 — PROGRAMMING MODEL



SIGNAL DESCRIPTION

The input and output signals for the MC6805K2 and MC6805K3 microcomputer units (MCUs) are described in the following paragraphs.

V_{CC}, V_{SS}, AND V_{STBY}

Power is supplied to the MCUs using these pins. V_{CC} provides the 5.0 volt ±5% power supply connection, V_{SS} is the ground connection, and V_{STBY} is the standby RAM power connection.

$\overline{INT1}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Zero-crossing detection capability is provided on this pin.

XTAL AND EXTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal or a capacitor-resistor network, 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.

V_{PP}

This pin is used to supply programming voltage (21 volts) to the EEPROM in the program mode. It should be connected to V_{CC} during normal operation.

TIMER A/PD7 — TIMER B/PD5

These pins allow an external input to be used to decrement the internal timers.

$\overline{RESET}$

This pin allows resetting of the MCU by an external source.

INPUT/OUTPUT PORTS (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers. All ports are CMOS and TTL input compatible and TTL output compatible.

MEMORY

As shown in Figure 3, the MC6805K2 and MC6805K3 microcomputers are capable of addressing 4096 bytes of memory space with their program counters. The MCUs have implemented 2048 bytes of ROM including eight interrupt vectors, 248 bytes of self-check ROM, 96 bytes of user RAM, 128 bytes of EEPROM, and 17 bytes of port I/O, control, data, and status registers. The user ROM is split into two areas. One area is the main memory (locations $700 to $EFF). The last eight user ROM locations, $FF8 to $FFF, are for the interrupt vectors.

The MCUs reserve the first 17 memory locations for I/O and hardware features. These locations are used for the ports, the port data direction registers, the timers, the miscellaneous register, the serial peripheral interface, and the EEPROM program control. Of the 96 RAM bytes, 31 ($061 through $07F) are shared with the stack area. The stack must be used with care when data shares the stack area. The lower sixteen bytes of RAM, between $20 and $2F, are powered through the V_{STBY} pin.

The shared stack area is used during the processing of an interrupt or subroutine calls, to save the contents of the CPU state. Since the register contents are pushed onto the stack, the stack pointer decrements during pushes. The low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack, since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed (see Figure 4).

FIGURE 3 — MEMORY MAP

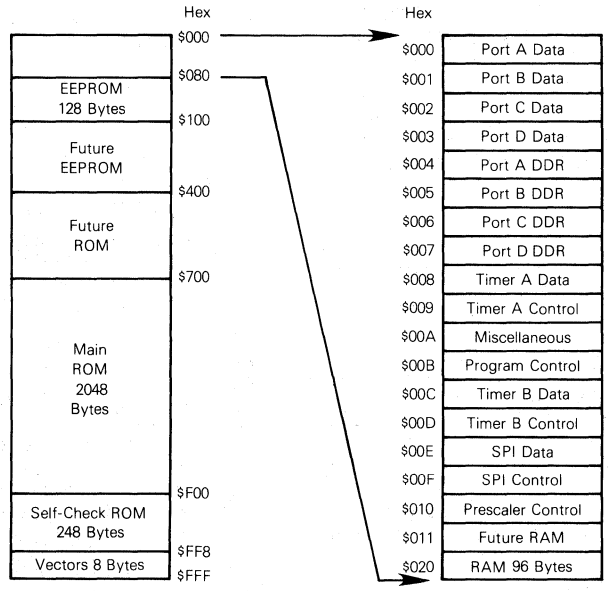
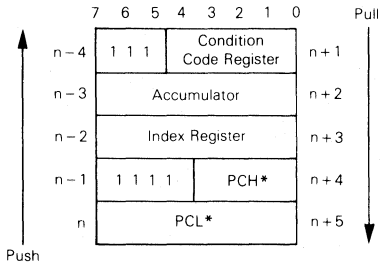


FIGURE 4 — INTERRUPT STACKING DIAGRAM



*For subroutine calls, only PCL and PCH are stacked.

MOTOROLA

MC6805P2

Advance Information

8-BIT MICROCOMPUTER UNIT

The MC6805P2 Microcomputer Unit (MCU) is a member of the M6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer 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 following are some of the hardware and software highlights of the MC6805P2 MCU.

HARDWARE FEATURES

- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 1100 Bytes of User ROM
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Self-Check Mode
- Zero Crossing Detection
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply

SOFTWARE FEATURES

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O

USER SELECTABLE OPTIONS

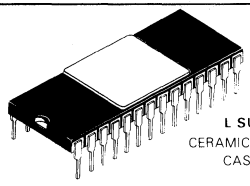
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits, 2n)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer, Software, and External

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

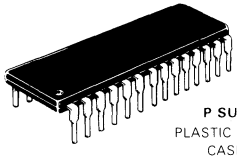
HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

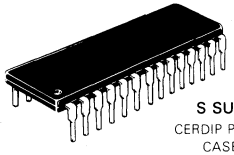
8-BIT MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719

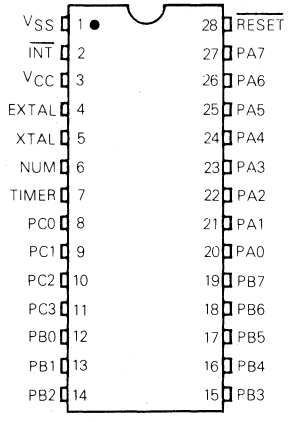


P SUFFIX
PLASTIC PACKAGE
CASE 710



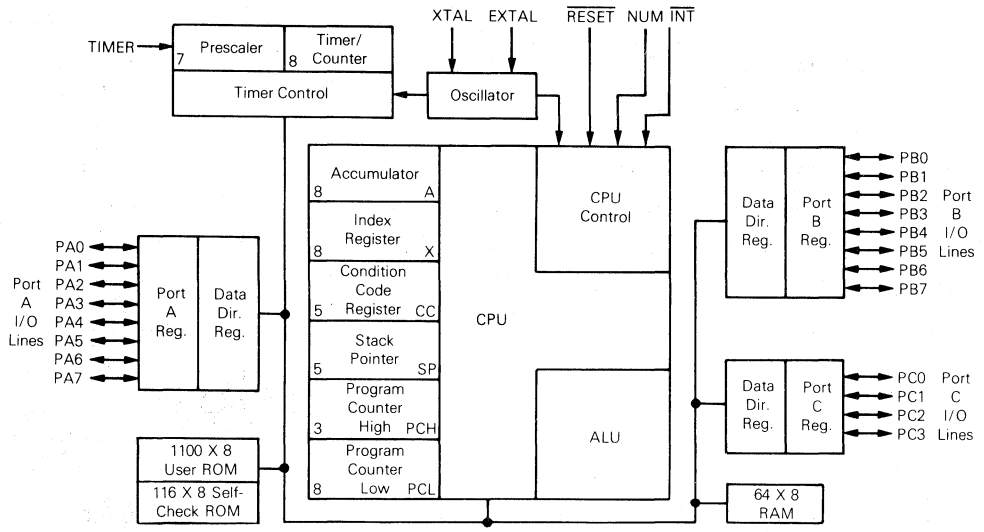
S SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT



3

FIGURE 1 — MC6805P2 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage (Except Pin 6)	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature			
Plastic	T_J	150	°C
Ceramic		175	
Cerdip		175	

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	72	°C/W
Ceramic		50	
Cerdip		60	

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} – 0.5	–	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} < 4.75)		V_{CC} – 0.5	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer					
Timer Mode	V_{IH}	2.0	–	V_{CC} + 1	V
Self-Check Mode		–	10.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltage (See Figures 10, 11, and 12)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{INT}$ Zero Crossing Input Voltage, Through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p-p}$
Internal Power Dissipation – No Port Loading V_{CC} = 5.75 V, T_A = 0°C	P_{INT}	–	400	690	mW
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit 0°C to 70°C	V_{LVI}	2.75	3.5	–	V
–40°C to 85°C		3.1	3.5	–	
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{INT}$ (V_{in} = 2.4 V to V_{CC})		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC}, Crystal Option)		–	–	10	
(V_{in} = 0.4 V, Crystal Option)		–	–	–1600	
$\overline{RESET}$ (V_{in} = 0.8 V)		–4.0	–	–40	
(External Capacitor Sharing Current)					

*Due to internal biasing, this input (when unused) floats to approximately 2.0 Vdc.

PORT DC ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = –100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = –10 μA	V_{OH}	V_{CC} – 1	–	–	V
Input High Voltage, I_{Load} = –300 μA (max.)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltate, I_{Load} = –500 μA (max.)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	–300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	–500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = –200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	–1.0	–	–10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port C and Port A with CMOS Drive Disabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = –100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA

See MC68(7)05R/U Series Data Sheet for port I/V curves and input protection schematics.

SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Oscillator Frequency	MC6805P2	f_{osc}	0.4	–	4.2	MHz
	MC68A05P2		0.4	–	6.0	
	MC68B05P2		0.4	–	8.0	
Cycle Time ($4/f_{osc}$)		t_{cyc}	0.95	–	10	μs
$\overline{INT}$ and TIMER Pulse Width (See Interrupt Section)		t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width		t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)		t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency		f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)		–	40	50	60	%

FIGURE 2 — TTL EQUIVALENT TEST LOAD (PORT B)

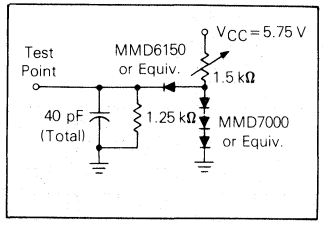


FIGURE 3 — CMOS EQUIVALENT TEST LOAD (PORT A)

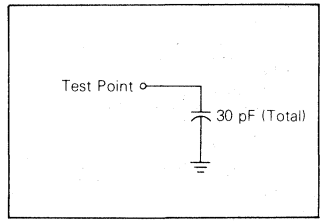
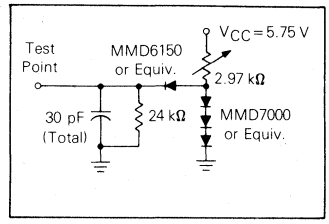


FIGURE 4 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

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.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL

These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, 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 Internal Clock Generator Options section for recommendations about these inputs.

TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

$\overline{RESET}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

NUM

This pin is not for user application and must be connected to V_{SS}.

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 Inputs/Outputs section for additional information.

MEMORY

As shown in Figure 5, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC6805P2 MCU has implemented 1288 of these locations. This consists of: 1100 bytes of user ROM, 116 bytes of self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, and 2 timer registers.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 6. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

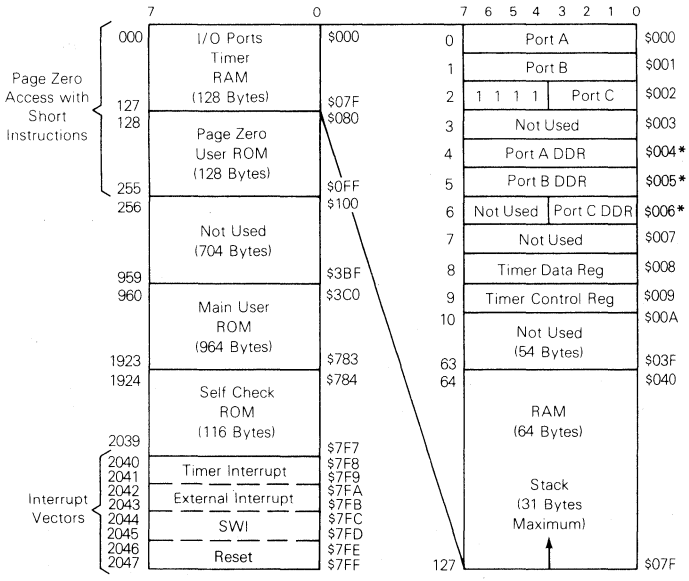
CENTRAL PROCESSING UNIT

The CPU of the M6805 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 address, data, and control buses.

REGISTERS

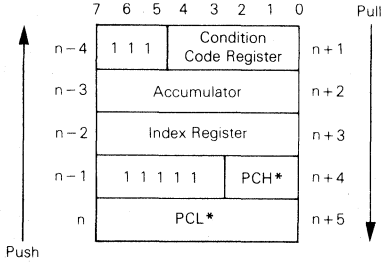
The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 7 and are explained in the following paragraphs.

FIGURE 5 — MCU ADDRESS MAP



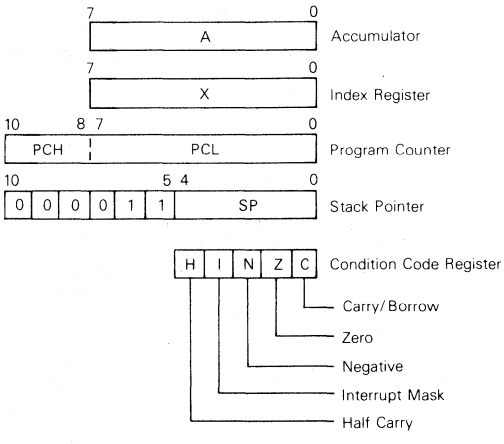
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 6 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH are stacked.

FIGURE 7 — PROGRAMMING MODEL



ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The six most significant bits of the stack pointer are permanently configured to 000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) — Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) — This bit is set to mask (disable) the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt is cleared.

NEGATIVE (N) — Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) — Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

CARRY/BORROW (C) — Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The MC6805P2 MCU timer circuitry is shown in Figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and executing the interrupt routine; see the Interrupts section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

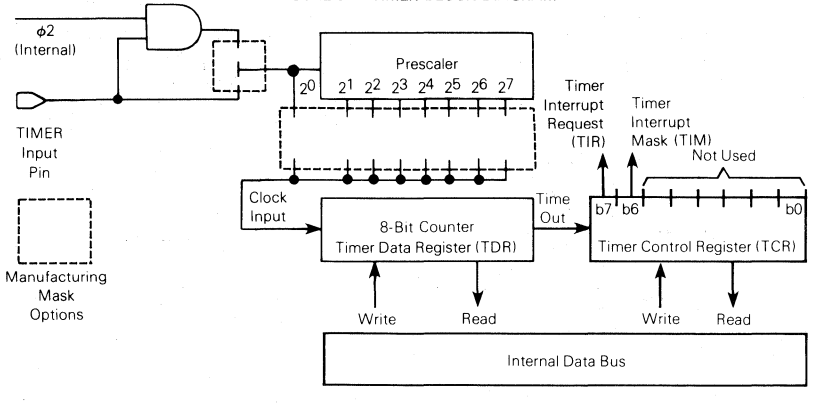
The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE: For ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER

FIGURE 8 — TIMER BLOCK DIAGRAM



pin should be tied to V_{CC}.) The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture.

The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the MC6805P2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 7 Hz. A 9-volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{RESET}$), and by an optional internal low voltage detect circuit; see Figure 10. The internal circuit connected to the $\overline{RESET}$ pin consists of a Schmitt trigger which senses the $\overline{RESET}$ line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic "0" on the $\overline{RESET}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{RESET}$ pin voltage rises to V_{IRES+}. When the $\overline{RESET}$ pin voltage falls to a logical "0" for a period longer than one t_{CYC}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 11.

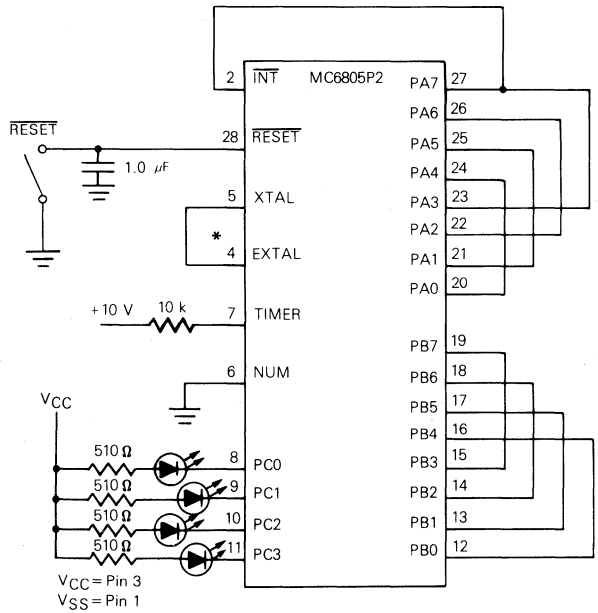
During power-up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{RESET}$ input, as shown in Figure 12, typically provides sufficient delay. See Figure 16 under Interrupts section for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, 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 oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in Figure 13. The crystal specifications and suggested PC board layouts

FIGURE 9 — SELF-CHECK CONNECTIONS



*This connection depends on the clock oscillator user selectable mask option. Use crystal if crystal option is selected.

are given in Figure 15. A resistor selection graph is given in Figure 16.

The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

FIGURE 10 — POWER AND $\overline{\text{RESET}}$ TIMING

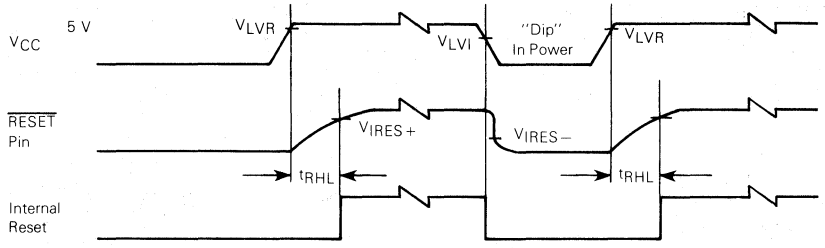


FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

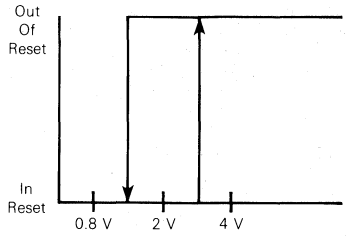


FIGURE 12 — POWER-UP RESET DELAY CIRCUIT

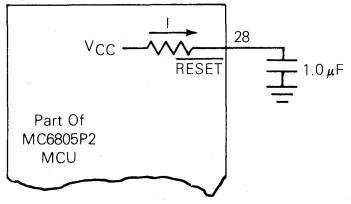
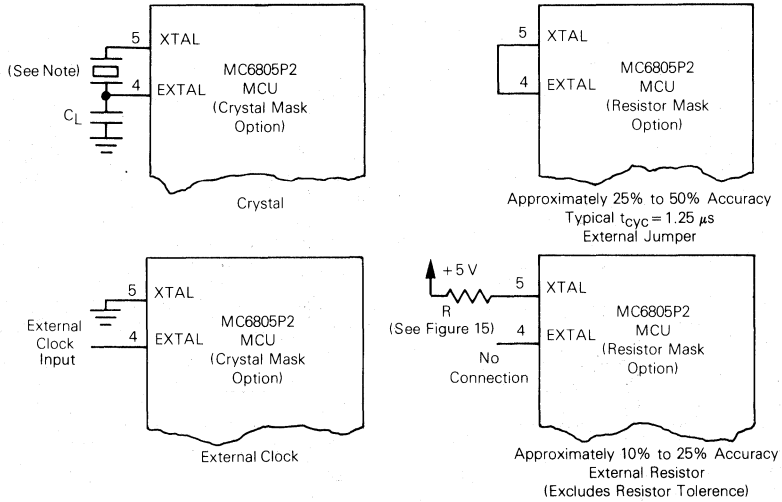
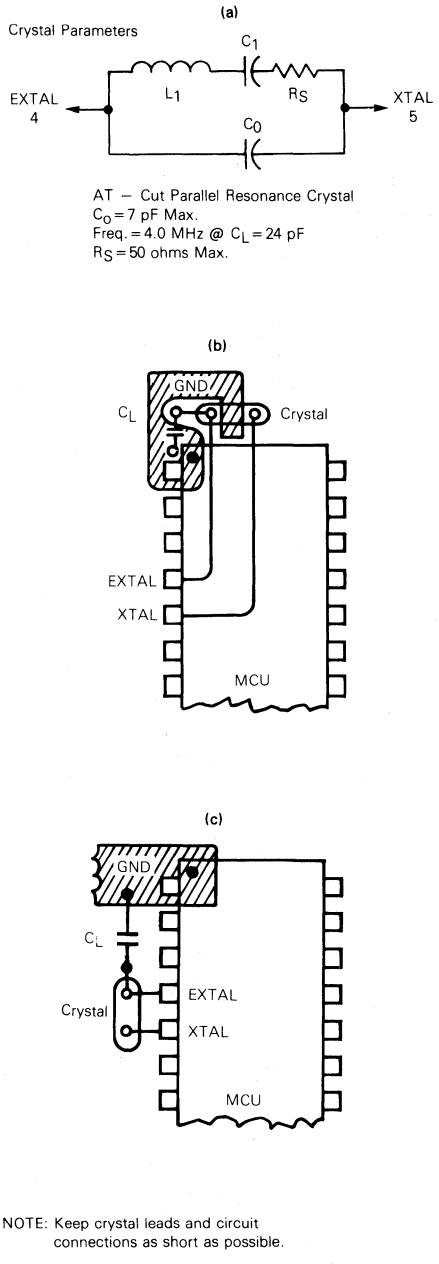


FIGURE 13 — CLOCK GENERATOR OPTIONS



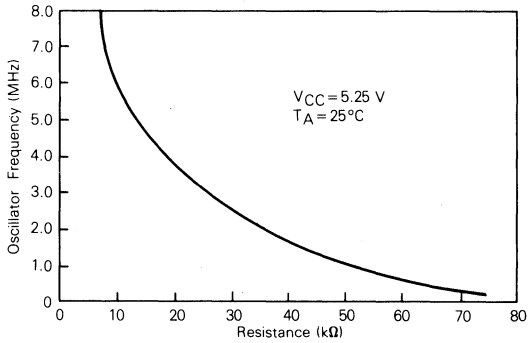
NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

FIGURE 14 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



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

FIGURE 15 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P2 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{CYC} periods for completion.

A flowchart of the interrupt sequence is shown in Figure 16. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{RESET}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 17(a), for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also

FIGURE 16 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

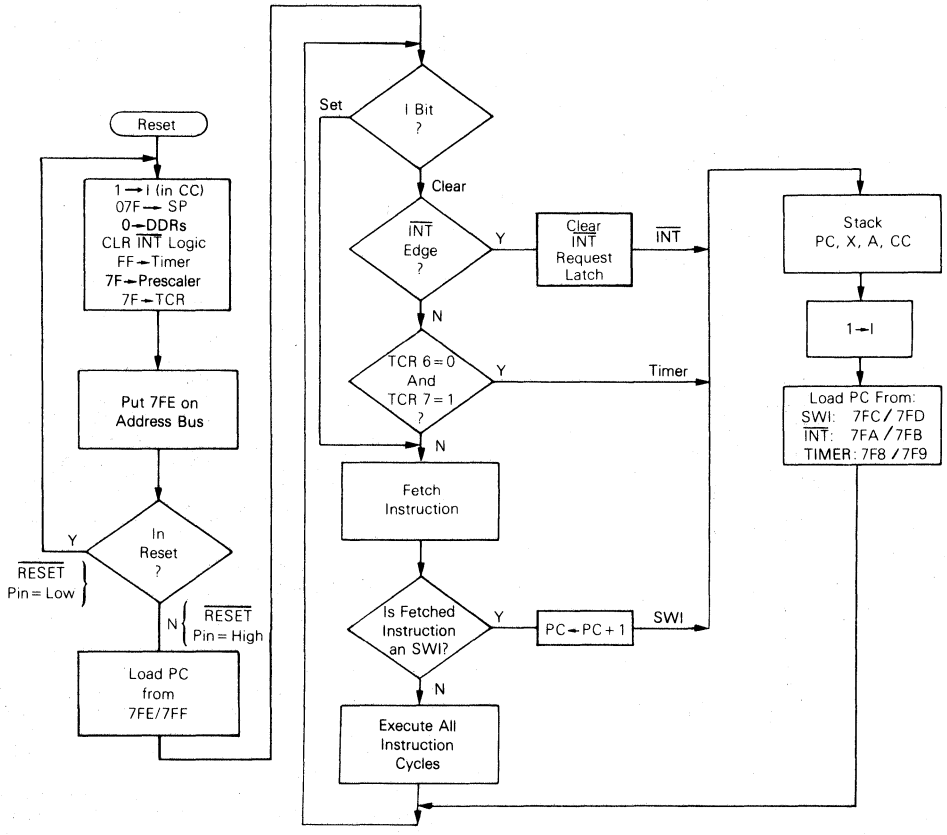
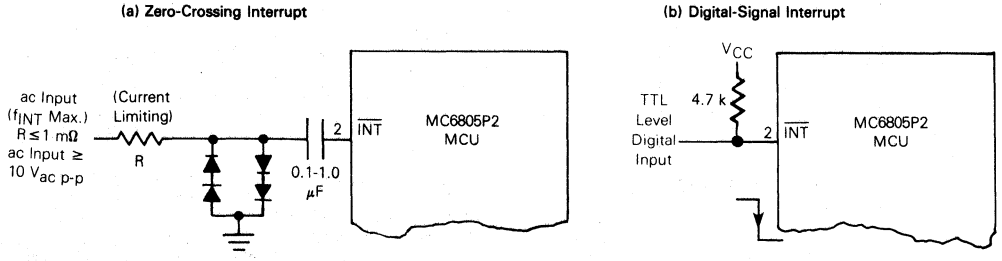


FIGURE 17 — TYPICAL INTERRUPT CIRCUITS



recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 17(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. Note that if the I bit is zero SWI executes after the other interrupts. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. 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 "1" for output or a logic "0" for input. On reset, all the DDRs are initialized to a logic "0" state to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting 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 18. When port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

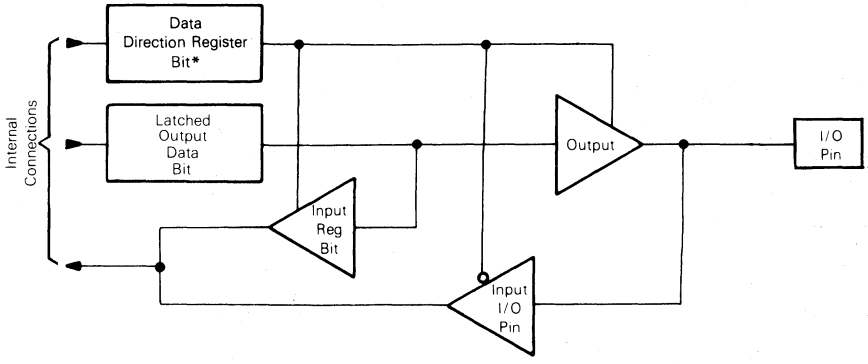
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 5 gives the address of data registers and DDRs. The register configuration is provided in Figure 19 and Figure 20 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see Figure 18) may always be written. Therefore, any write to a port writes 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").

FIGURE 18 — TYPICAL PORT I/O CIRCUITRY



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

*DDR is a write-only register and reads as all "1s".

**Ports A (with CMOS drive disabled), B, and C are three state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

FIGURE 19 — MCU REGISTER CONFIGURATION

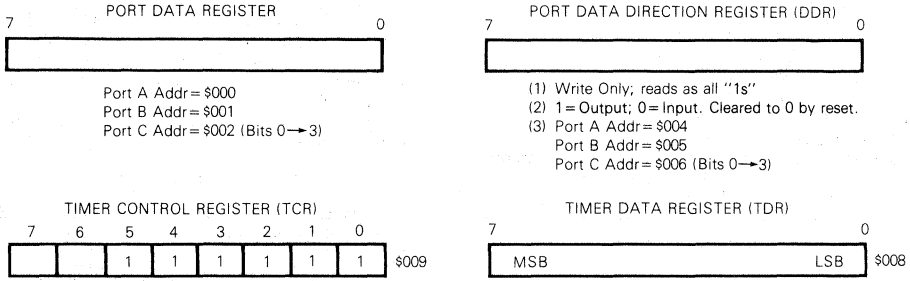
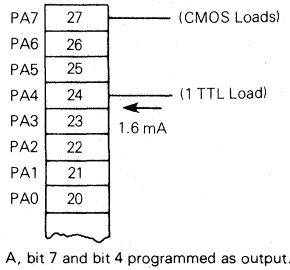
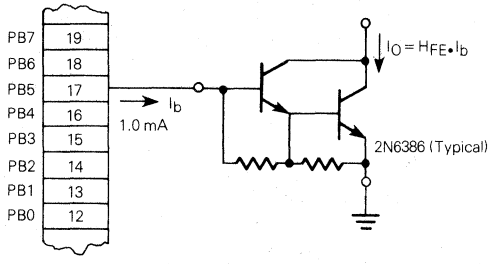


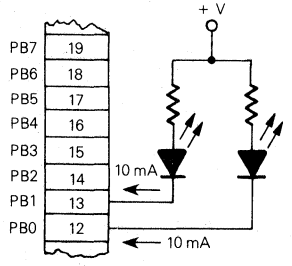
FIGURE 20(a) — TYPICAL OUTPUT MODE PORT CONNECTIONS



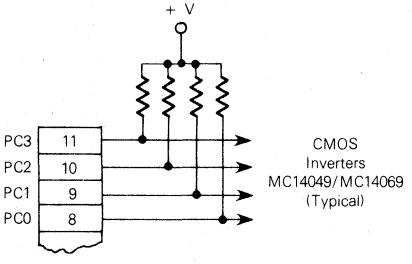
Port A, bit 7 and bit 4 programmed as output. Bit 7 driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.



Port B, bit 5 programmed as output, driving Darlington-base directly.

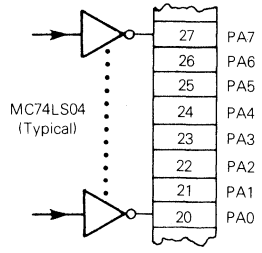


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

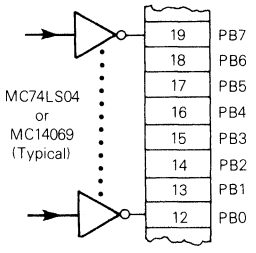


Port C, bits 0-3 programmed as output, driving CMOS loads, using external pullup resistors.

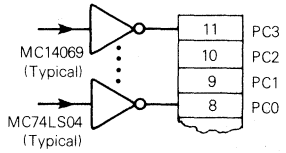
FIGURE 20(b) — TYPICAL INPUT MODE PORT CONNECTIONS



TTL driving port A directly.



CMOS or TTL driving port B directly.



CMOS and TTL driving port C directly.

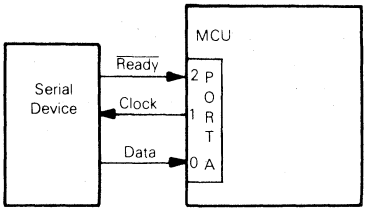
SOFTWARE

BIT MANIPULATION

The MC6805P2 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register, see Caution under Input/Output section), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches 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 RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 21 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, LSB 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 a RAM location.

FIGURE 21 — BIT MANIPULATION EXAMPLE




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

ADDRESSING MODES

The MC6805P2 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family User's Manual.

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.

IMMEDIATE — In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE — The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 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.

INDEXED, NO OFFSET — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET — In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET — In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the Input/Output section.

BIT TEST AND BRANCH — The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the condition code register. See Caution under the Input/Output section.

INHERENT — In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC6805P2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS — Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 1.

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 (see Caution under Input/Output section). The test for negative or zero (TST) instruction is included in read-modify-write instructions though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 3.

BIT MANIPULATION INSTRUCTIONS — These instructions are used on any bit in the first 256 bytes of the memory (see Caution under Input/Output section). One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS — The control instructions control the MCU operations during program execution. Refer to Table 5.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 6.

OPCODE MAP SUMMARY — Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

		Relative Addressing Mode		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0....7)	—	—	—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0....7)	—	—	—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0...7)	10 + 2 • n	2	7	—	—	—
Clear bit n	BCLR n (n = 0....7)	11 + 2 • n	2	7	—	—	—

TABLE 5 — CONTROL INSTRUCTIONS

		Inherent		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	↕
BRSET										X	•	•	•	•	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	↕	↕	↕
COM	X		X			X	X				•	•	↕	↕	1
CPX		X	X	X		X	X	X			•	•	↕	↕	↕
DEC	X		X			X	X				•	•	↕	↕	•
EOR		X	X	X		X	X	X			•	•	↕	↕	•
INC	X		X			X	X				•	•	↕	↕	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	↕	↕	•
LDX		X	X	X		X	X	X			•	•	↕	↕	•
LSL	X		X			X	X				•	•	↕	↕	↕
LSR	X		X			X	X				•	•	0	↕	↕
NEQ	X		X			X	X				•	•	↕	↕	↕
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	↕	↕	•
ROL	X		X			X	X				•	•	↕	↕	↕
RSP	X										•	•	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
↕ Test and Set if True, Cleared Otherwise
• Not Affected

TABLE 6 — INSTRUCTION SET (CONTINUED)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
Λ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack

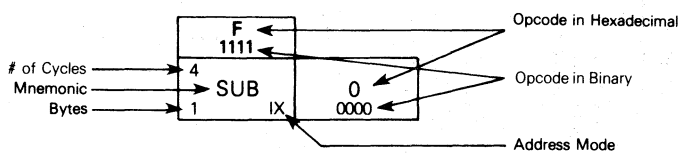
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DIR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



ORDERING INFORMATION

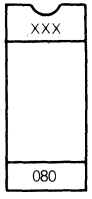
The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the MCU it is necessary to first contact your local Motorola representative or Motorola distributor.

EPROMs

The MC68705P3 EPROM MCU programmed with the customer program may be used to submit the ROM pattern. Note that while the MC6805P2 has 1.1K Bytes of ROM, the MC68705P3 contains 1.8K of EPROM memory.

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below:



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

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 on blank EPROM from the data file used to create the custom mask and aid in the verification process.

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 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by Motorola Quality Assurance and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name on the disk with a felt-tip pen. The minimum MDOS system files as well as the absolute binary object file (filename L0 type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: filename, LX (EXORciser loadable format) and filename, SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

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

GENERIC INFORMATION

Package Type	Internal Clock Frequency (MHz)	Temperature	Generic Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6805P2L
	1.5	0°C to 70°C	MC68A05P2L
	2.0	0°C to 70°C	MC68B05P2L
Plastic P Suffix	1.0	0°C to 70°C	MC6805P2P
	1.5	0°C to 70°C	MC68A05P2P
	2.0	0°C to 70°C	MC68B05P2P
Cerdip S Suffix	1.0	0°C to 70°C	MC6805P2S
	1.5	0°C to 70°C	MC68A05P2S
	2.0	0°C to 70°C	MC68B05P2S

MC6805P2 MCU CUSTOM ORDERING INFORMATION

Date _____ Customer PO Number _____

Customer Company _____ Motorola Part Numbers

Address _____ MC _____

SC _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Pattern Media (All other media requires prior factory approval.)
- ☐ EPROMs (MCM2716 or MCM2532)
- ☐ EPROM MCU (MC68705P3)
- ☐ Floppy Disk
- ☐ Other _____

Clock Freq. _____

Temp. Range _____ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C

Marking Information (12 Characters Maximum)

Title _____

Signature _____

MOTOROLA

MC6805P4

Advance Information

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT MICROCOMPUTER

8-BIT MICROCOMPUTER UNIT

The MC6805P4 Microcomputer Unit (MCU) is a member of the M6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer 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 following are some of the hardware and software highlights of the MC6805P4 MCU.

HARDWARE FEATURES

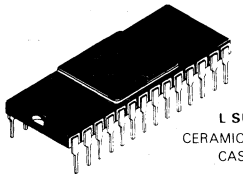
- 8-Bit Architecture
- 112 Bytes of Standby RAM
- Standby RAM Power Pin
- Memory Mapped I/O
- 1100 Bytes of User ROM
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Self-Check Mode
- Zero-Crossing Detection
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply

SOFTWARE FEATURES

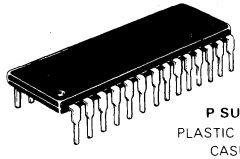
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O

USER SELECTABLE OPTIONS

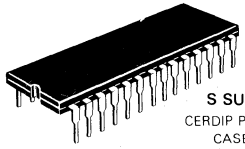
- Standby RAM Size is Mask Programmable
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits, 2^n)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer, Software, and External
- Open Drain Port Option on Ports B and C



L SUFFIX
CERAMIC PACKAGE
CASE 719

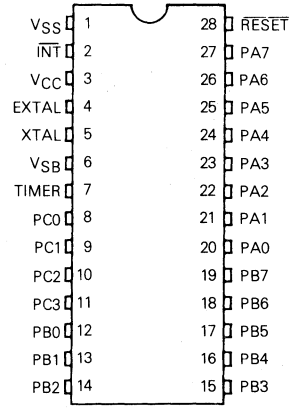


P SUFFIX
PLASTIC PACKAGE
CASE 710



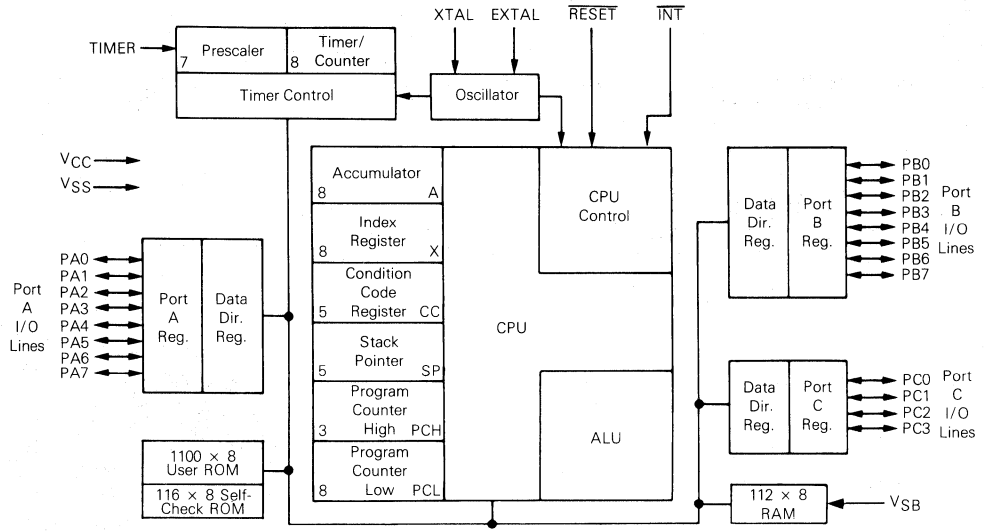
S SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT



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

FIGURE 1 — MC6805P4 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage (Except TIMER in Self-Check Mode)	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature Plastic Ceramic Cerdip	T_J	 150 175 175	°C

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip	θ_{JA}	 72 50 60	°C/W

POWER CONSIDERATIONS

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

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

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

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

P_{PORT} = Port Power Dissipation, Watts — User Determined

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

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

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

Solving equations 1 and 2 for K gives:

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

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

SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time ($4/f_{osc}$)	t_{cyc}	0.95	–	10	μs
$\overline{INT}$ and TIMER Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)	–	40	50	60	%

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} – 0.5	–	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} < 4.75)		V_{CC} – 0.5	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer					
Timer Mode	V_{IH}	2.0	–	V_{CC} + 1	V
Self-Check Mode		–	10.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltage (See Figures 10, 11, and 12)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{INT}$ Zero-Crossing Input Voltage, Through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p\text{-}p}$
Internal Power Dissipation – No Port Loading V_{CC} = 5.75 V, T_A = 0°C	P_{INT}	–	400	TBD	mW
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit 0°C to 70°C	V_{LVI}	2.75	3.5	–	V
–40°C to 80°C		3.1	3.5	–	
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{INT}$ (V_{in} = 2.4 V to V_{CC})		–	20	50	
EXTAL (V_{in} = 0.4 V, Crystal Option)		–	–	10	
_____ (V_{in} = 0.4 V, Crystal Option)		–	–	–1600	
$\overline{RESET}$ (V_{in} = 0.8 V)		–4.0	–	–40	
(External Capacitor Charging Current)					

* Due to internal biasing, this input (when unused) floats to approximately 2.0 Vdc.

PORT DC ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = – 10 μA	V_{OH}	V_{CC} – 1	–	–	V
Input High Voltage, I_{Load} = – 300 μA (max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = – 500 μA (max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	– 300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	– 500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = – 200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	– 1.0	–	– 10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port C and Port A with CMOS Drive Disabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port B and Port C with Open-Drain Option					
Output High Voltage	V_{OH}	2.4	–	13.0	V
Hi-Z State Input Current	I_{TSI}	–	–	20	μA

See MC68(7)05 Series Data Sheet for port I/V curves and input protection schematics.

STANDBY RAM CHARACTERISTICS (T_A = 0°C to 70°C)

Characteristic	Symbol	Min	Typ	Max	Unit
Standby Current					
8 Bytes	I_{SB}	–	1.0	TBD	mA
32 Bytes		–	2.2	TBD	
64 Bytes		–	3.4	TBD	
112 Bytes		–	5.2	TBD	
RAM Standby Voltage	V_{SB}	3.0	5.25	5.75	V
V_{CC} Turn-off Rate	V_{CCTO}	–	–	1/100	V/μs

FIGURE 2 — TTL EQUIVALENT TEST LOAD (PORT B)

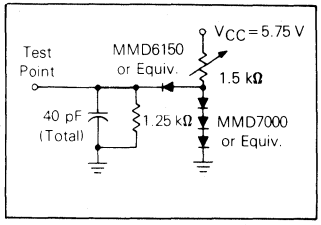


FIGURE 3 — CMOS EQUIVALENT TEST LOAD (PORT A)

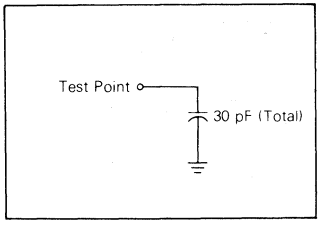
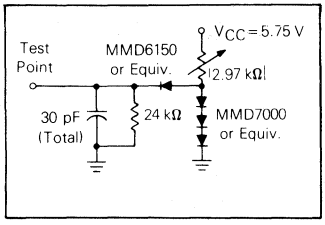


FIGURE 4 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

V_{CC}, V_{SS}

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

V_{SB}

This pin supplies the standby RAM voltage. In order to allow orderly transition into the standby mode, the turn-off rate of V_{CC} must not exceed 1 volt per 100 μs.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL

These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, 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 Internal Clock Generator Options section for recommendations about these inputs.

TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

$\overline{RESET}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

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 Inputs/Outputs section for additional information.

MEMORY

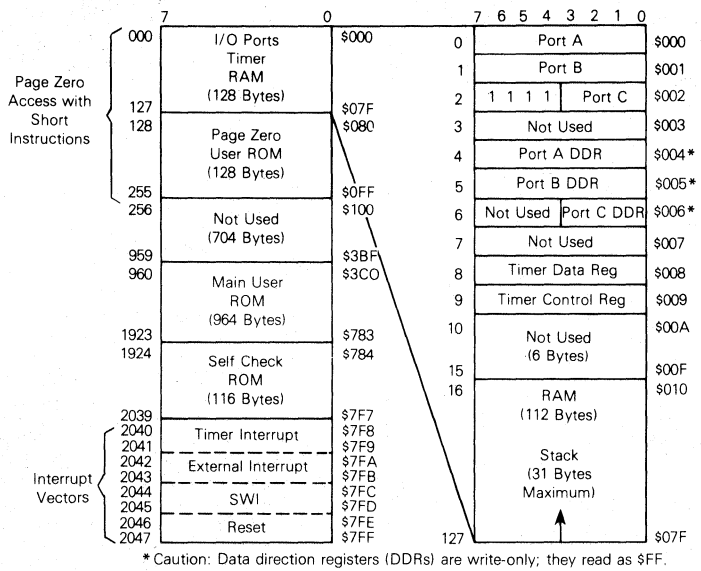
As shown in Figure 5, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC6805P4 MCU has implemented 1336 of these locations. This consists of: 1100 bytes of user ROM, 116 bytes of self-check ROM, 112 bytes of user RAM, 6 bytes of port I/O, and 2 timer registers.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 6. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first, then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

CENTRAL PROCESSING UNIT

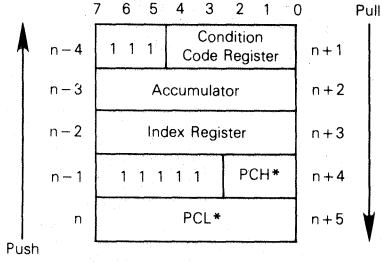
The CPU of the M6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

FIGURE 5 — MC6805P4 MCU ADDRESS MAP



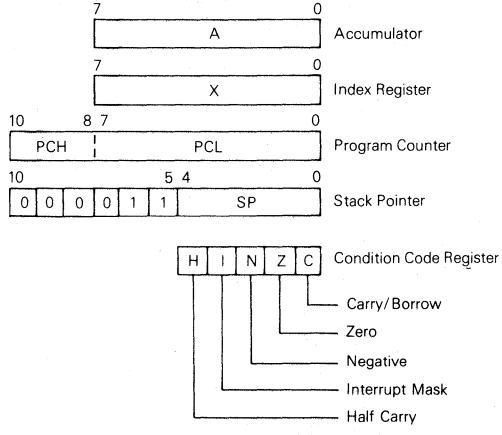
* Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 6 — INTERRUPT STACKING ORDER



* For subroutine calls, only PCL and PCH are stacked.

FIGURE 7 — PROGRAMMING MODEL



REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 7 and are explained in the following paragraphs.

ACCUMULATOR (A) — The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The six most significant bits of the stack pointer are permanently configured to 000011. During an MCU reset or the Reset Stack Pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 31 levels of subroutine calls.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) — Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) — This bit is set to mask (disable) the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

NEGATIVE (N) — Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) — Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

CARRY/BORROW (C) — Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The MC6805P4 MCU timer circuitry is shown in Figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and executing the interrupt routine; see the Interrupts section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be an internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

FIGURE 8 — TIMER BLOCK DIAGRAM

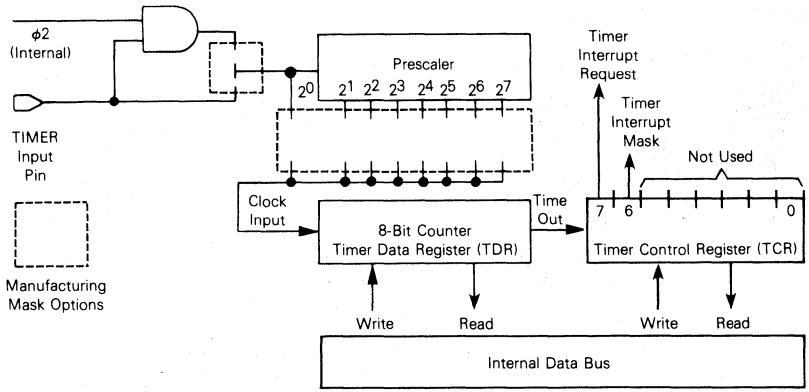
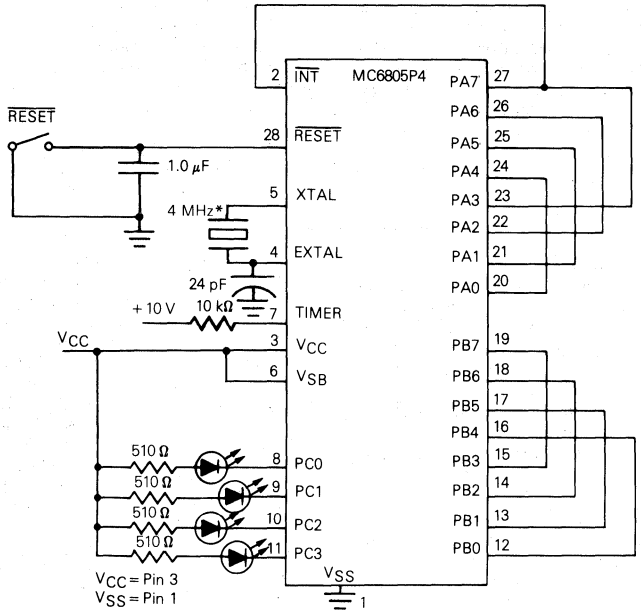


FIGURE 9 — SELF-CHECK CONNECTIONS



*NOTE: For RC user selectable mask option, omit the crystal and the 24 pF capacitor and connect pins 4 and 5 together with a jumper or resistor to V_{CC}.

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (Note: for ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER pin should be tied to V_{CC}.) The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

A prescaler option, divide by 2^n, can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture.

The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the MC6805P4 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 7 Hz. A 9-volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{RESET}$), and by an optional internal low voltage detect circuit; see Figure 10. The internal circuit connected to the $\overline{RESET}$ pin consists of a Schmitt trigger which senses the $\overline{RESET}$ line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic "0" on the $\overline{RESET}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{RESET}$ pin voltage rises to V_{IRES+}. When the $\overline{RESET}$ pin voltage falls to a logical "0" for a period longer than one t_{CYC}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 11.

During power-up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{RESET}$ input, as shown in Figure 12, typically provides sufficient delay. See Figure 16 under the Interrupts section for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a

jumper wire, 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 oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in Figure 13. The crystal specifications and suggested PC board layouts are given in Figure 14. A resistor selection graph is given in Figure 15.

The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

FIGURE 10 — POWER AND $\overline{\text{RESET}}$ TIMING

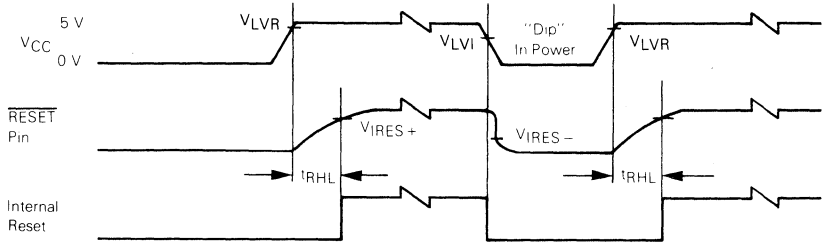


FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

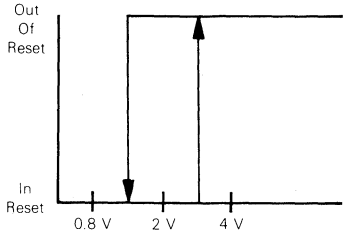


FIGURE 12 — POWER-UP RESET DELAY CIRCUIT

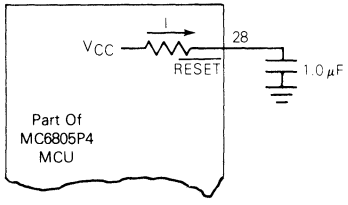
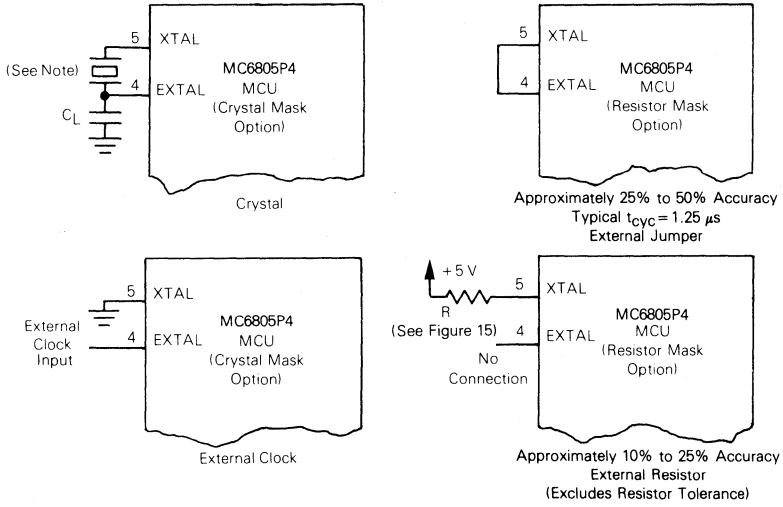
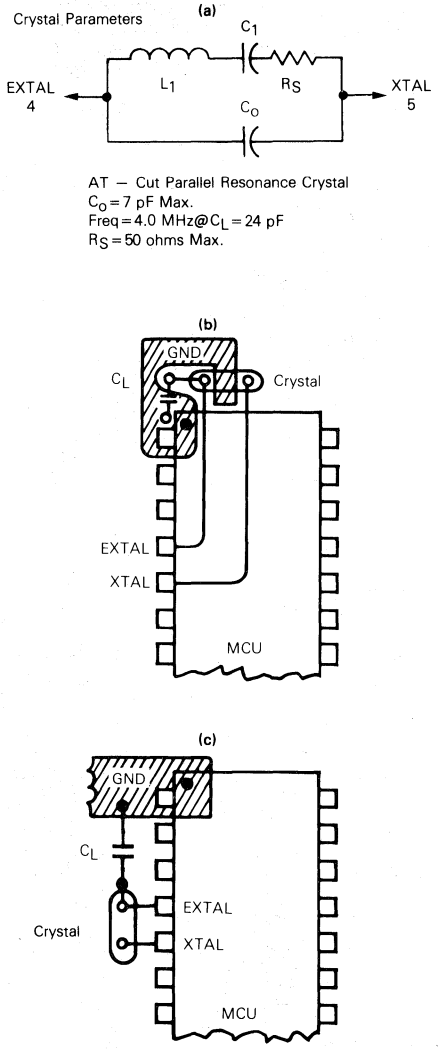


FIGURE 13 — CLOCK GENERATOR OPTIONS



NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

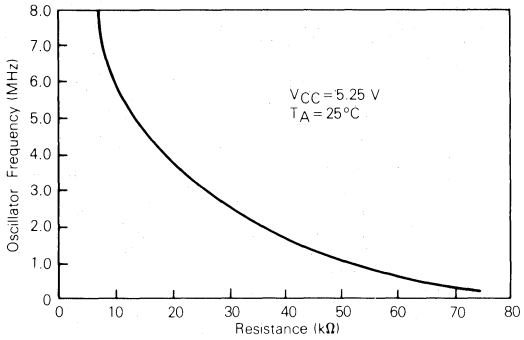
FIGURE 14 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



AT — Cut Parallel Resonance Crystal
C_O = 7 pF Max.
Freq = 4.0 MHz@C_L = 24 pF
R_S = 50 ohms Max.

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

FIGURE 15 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P4 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching require a total of 11 t_{CYC} periods for completion.

A flowchart of the interrupt sequence is shown in Figure 16. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{RESET}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 17(a), for use as a Zero-Crossing Detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices.

FIGURE 16 — RESET AND INTERRUPT PROCESSING FLOWCHART

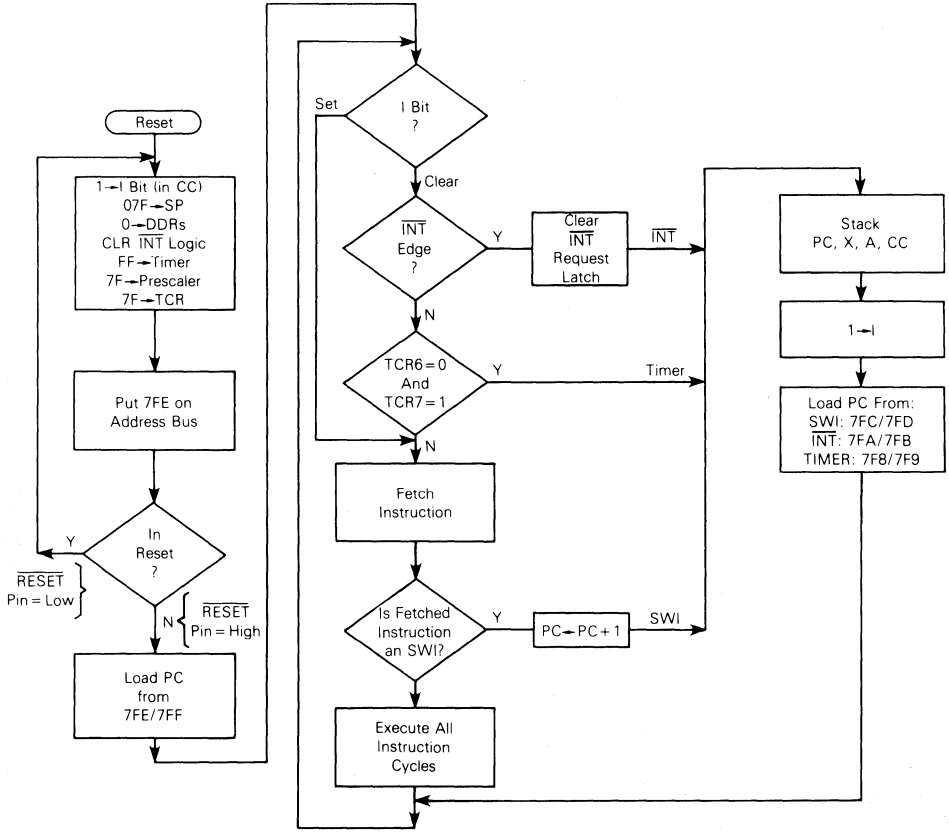
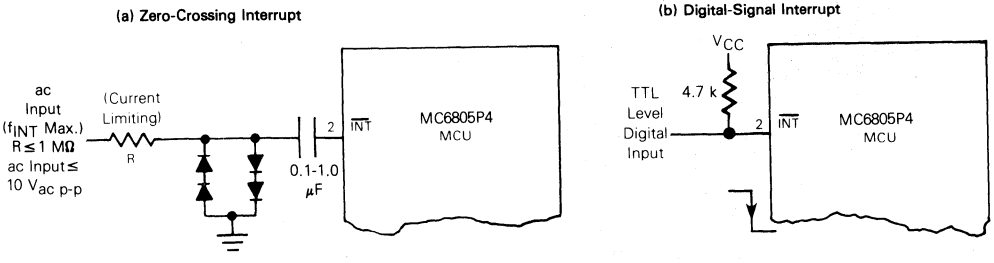


FIGURE 17 — TYPICAL INTERRUPT CIRCUITS



For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 17(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. Note that if the I bit is zero SWI executes after the other interrupts. SWIs are usually used as breakpoints for debugging or as system calls.

3

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. All pins (Ports 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 "1" for output or a logic "0" for input. On reset, all the DDRs are initialized to a logic "0" state to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting 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 18. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

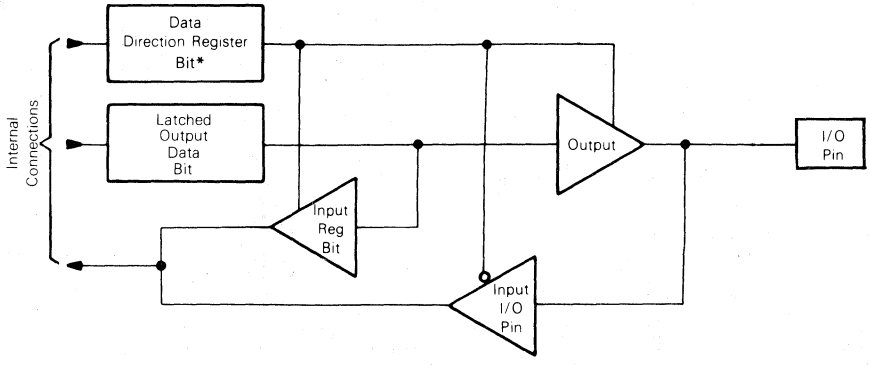
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 5 gives the address of data registers and DDRs. The register configuration is provided in Figure 19 and Figure 20 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see Figure 18) may always be written. Therefore, any write to a port writes 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).

FIGURE 18— TYPICAL PORT I/O CIRCUITRY

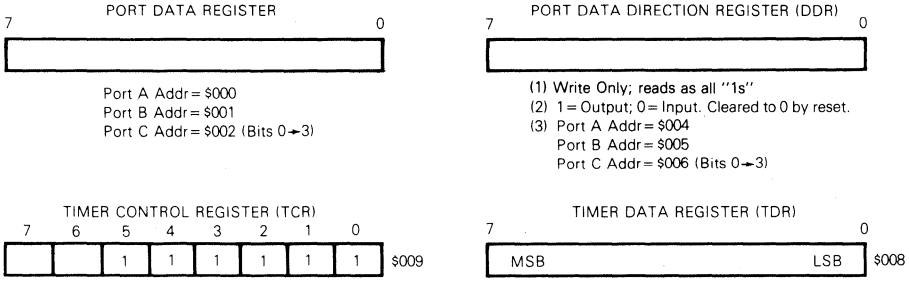


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

*DDR is a write-only register and reads as all "1s"

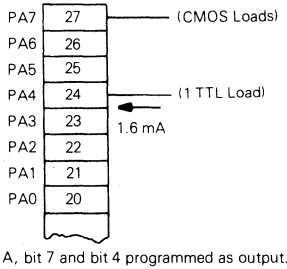
**Ports A (with CMOS drive disabled), B, and C are three-state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

FIGURE 19 — MCU REGISTER CONFIGURATION

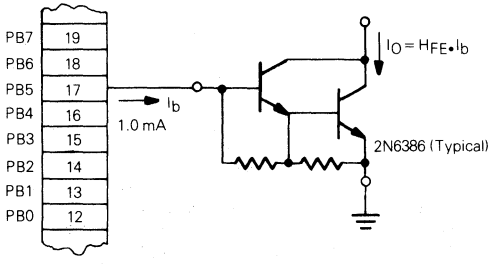


TCR7 — Timer Interrupt Status Request Bit: Set when TDR goes to zero; must be cleared by software. Cleared to 0 by reset.
TCR6 Bit 6 — Timer Interrupt Mask Bit: 1 = timer interrupt masked (disabled). Set to 1 by reset.
TCR Bits 5, 4, 3, 2, 1, 0 read as "1s" — unused bits.

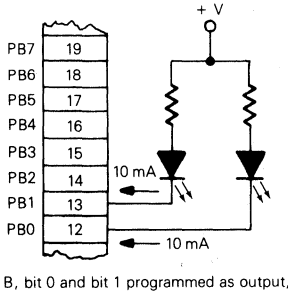
FIGURE 20(a) — TYPICAL OUTPUT MODE PORT CONNECTIONS



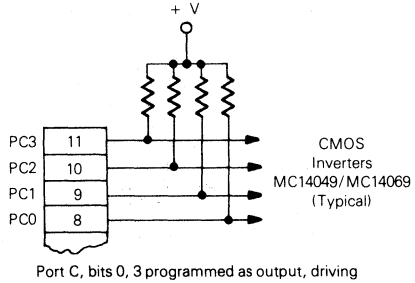
Port A, bit 7 and bit 4 programmed as output. Bit 7 driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.



Port B, bit 5 programmed as output, driving Darlington-base directly.

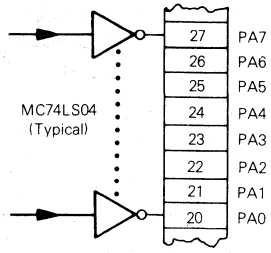


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

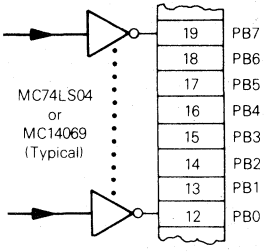


Port C, bits 0, 3 programmed as output, driving CMOS loads, using external pullup resistors.

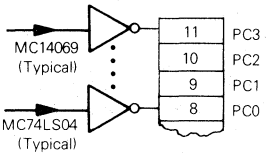
FIGURE 20(b) — TYPICAL INPUT MODE PORT CONNECTIONS



TTL Driving Port A Directly



CMOS or TTL Driving Port B Directly



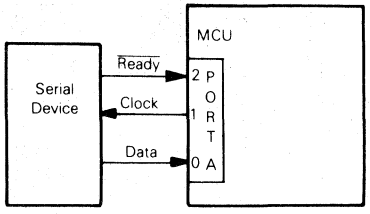
CMOS and TTL Driving Port C Directly

BIT MANIPULATION

The MC6805P4 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register; see Caution under Input/Output section) with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches 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 RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 21 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 clock line to clock data one bit at a time, LSB 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 a RAM location.

FIGURE 21 — BIT MANIPULATION EXAMPLE




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

ADDRESSING MODES

The MC6805P4 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family User's Manual.

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.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 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.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEX, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the Input/Output section.

BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the condition code register. See Caution under the Input/Output section.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC6805P4 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 1.

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 (see Caution under Input/Output section). The test for negative or zero (TST) instruction is included in the read-modify-write instructions though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS

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

BIT MANIPULATION INSTRUCTIONS

These instructions are used on any bit in the first 256 bytes of the memory (see Caution under Input/Output section). One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS

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

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 7.

OPCODE MAP SUMMARY

Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	10
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	10
Set Bit n	BSET n (n=0...7)	10+2•n	2	7	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	7	—	—	—

TABLE 5 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	•	^	^	^
ADD		X	X	X		X	X	X			^	•	^	^	^
AND		X	X	X		X	X	X			•	•	^	^	•
ASL	X		X			X	X				•	•	^	^	^
ASR	X		X			X	X				•	•	^	^	^
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	^	^	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	^
BRSET										X	•	•	•	•	^
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	^	^	^
COM	X		X			X	X				•	•	^	^	1
CPX		X	X	X		X	X	X			•	•	^	^	^
DEC	X		X			X	X				•	•	^	^	•
EOR		X	X	X		X	X	X			•	•	^	^	•
INC	X		X			X	X				•	•	^	^	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	^	^	•
LDX		X	X	X		X	X	X			•	•	^	^	•
LSL	X		X			X	X				•	•	^	^	^
LSR	X		X			X	X				•	•	0	^	^
NEQ	X		X			X	X				•	•	^	^	^
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	^	^	•
ROL	X		X			X	X				•	•	^	^	^
RSP	X										•	•	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected

TABLE 6 — INSTRUCTION SET (CONTINUED)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols:

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

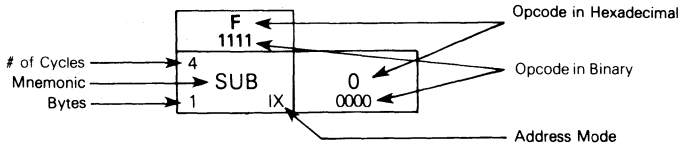
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read/Modify/Write DIR 3 0011	Read/Modify/Write INH 4 0100	Read/Modify/Write INH 5 0101	Read/Modify/Write IX1 6 0110	Read/Modify/Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



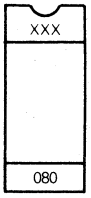
ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local Motorola representative or Motorola distributor.

EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below:



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

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 on blank EPROM from the data file used to create the custom mask and aid in the verification process.

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 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by Motorola Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The minimum MDOS system files, as well as the absolute binary object file (filename L0 type of file) from the M6805 cross assembler, must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: filename, LX (EXORciser loadable format) and filename, SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

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

GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6805P4L
Plastic P Suffix	1.0	0°C to 70°C	MC6805P4P
Cerdip S Suffix	1.0	0°C to 70°C	MC6805P4S

MC6805P4 MCU CUSTOM ORDERING INFORMATION

Date _____ Customer PO Number _____

Custorner Company _____ Motorola Part Numbers

MC _____

Address _____ SC _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
- ☐ 2^0 (divided by 1)
- ☐ 2^1 (divided by 2)
- ☐ 2^2 (divided by 4)
- ☐ 2^3 (divided by 8)
- ☐ 2^4 (divided by 16)
- ☐ 2^5 (divided by 32)
- ☐ 2^6 (divided by 64)
- ☐ 2^7 (divided by 128)

Port B Output Drive
- ☐ TTL
- ☐ Open Drain

Port C Output Drive
- ☐ TTL
- ☐ Open Drain

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Standby RAM
- ☐ 8 Bytes
- ☐ 32 Bytes
- ☐ 64 Bytes
- ☐ 112 Bytes

Pattern Media (All other media requires prior factory approval.)
- ☐ EPROMS (MCM2716 or MCM2532)
- ☐ Floppy Disk
- ☐ Other _____

Clock Freq. _____

Temp. Range _____ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C

Marking Information (12 Characters Maximum)

Title _____

Signature _____

MOTOROLA

MC6805P6

Advance Information

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT MICROCOMPUTER

3

8-BIT MICROCOMPUTER UNIT

The MC6805P6 Microcomputer Unit (MCU) is a member of the M6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer 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 following are some of the hardware and software highlights of the MC6805P6 MCU.

HARDWARE FEATURES

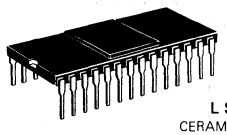
- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 1796 Bytes of User ROM
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Self-Check Mode
- Zero Crossing Detection
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply

SOFTWARE FEATURES

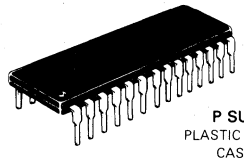
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O

USER SELECTABLE OPTIONS

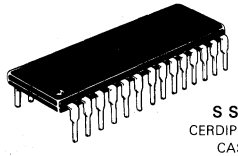
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits, 2^n)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer, Software, and External
- Port B Open Drain Drive Option



L SUFFIX
CERAMIC PACKAGE
CASE 719

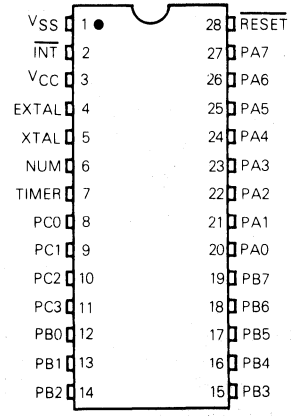


P SUFFIX
PLASTIC PACKAGE
CASE 710



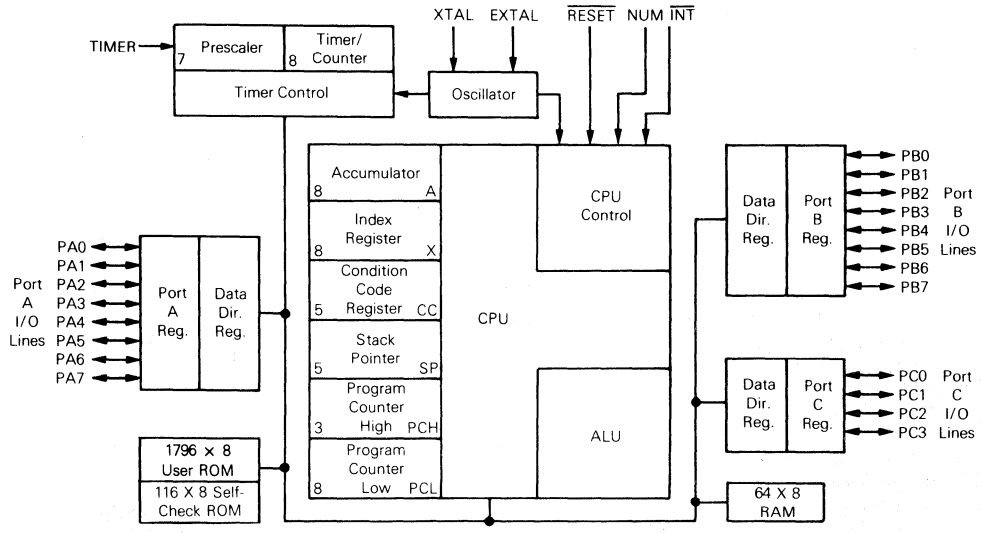
S SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT



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

FIGURE 1 — MC6805P6 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage (Except Pin 6)	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature			
Plastic	T_J	150	°C
Ceramic		175	
Cerdip		175	

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	72	°C/W
Ceramic		50	
Cerdip		60	

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±5.0 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	–	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{INT}$ ($4.75 \le V_{CC} \le 5.75$)	V_{IH}	4.0	*	V_{CC}	V
($V_{CC} < 4.75$)		$V_{CC}-0.5$	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer					
Timer Mode	V_{IH}	2.0	–	$V_{CC}+1$	V
Self-Check Mode		–	10.0	15.0	
Input Low Voltage					
$\overline{INT}$	V_{IL}	V_{SS}	*	1.5	V
All Other		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltage (See Figures 10, 11, and 12)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{INT}$ Zero Crossing Input Voltage, Through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p\text{-}p}$
Internal Power Dissipation – No Port Loading V_{CC} = 5.75 V, T_A = 0°C	P_{INT}	–	400	690	mW
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit 0°C to 70°C	V_{LVI}	2.75	3.5	–	V
–40°C to 85°C		3.1	3.5	–	
Input Current (External Capacitor Charging Current)					
TIMER (V_{in} = 0.4 V)		–	–	20	
$\overline{INT}$ (V_{in} = 2.4 V to V_{CC})		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC}, Crystal Option)	I_{in}	–	–	10	μA
(V_{in} = 0.4 V, Crystal Option)		–	–	–1600	
$\overline{RESET}$ (V_{in} = 0.8 V)		–4.0	–	–40	

*Due to internal biasing, this input (when unused) floats to approximately 2.0 Vdc.

PORT DC ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = –100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = –10 μA	V_{OH}	$V_{CC}-1$	–	–	V
Input High Voltage, I_{Load} = –300 μA (max.)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = –500 μA (max.)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	–300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	–500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = –200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	–1.0	–	–10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port C and Port A with CMOS Drive Disabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = –100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port B with Open-Drain Option					
Output High Voltage	V_{OH}	2.4	–	13.0	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

See MC68(7)05 Series Data Sheet for port I/V curves and input protection schematics.

SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Oscillator Frequency	MC6805P6	f_{osc}	0.4	–	4.2	MHz
	MC68A05P6		0.4	–	6.0	
	MC68B05P6		0.4	–	8.0	
Cycle Time ($4/f_{osc}$)		t_{cyc}	0.95	–	10	μs
$\overline{INT}$ and TIMER Pulse Width (See INTERRUPTS)		t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width		t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)		t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency		f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)		–	40	50	60	%

FIGURE 2 — TTL EQUIVALENT TEST LOAD (PORT B)

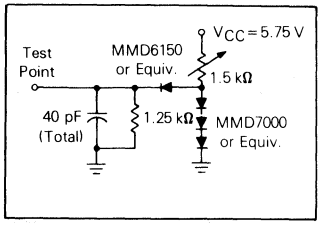


FIGURE 3 — CMOS EQUIVALENT TEST LOAD (PORT A)

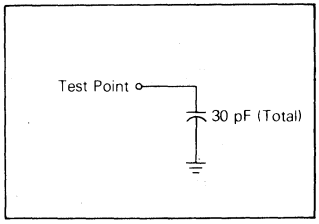
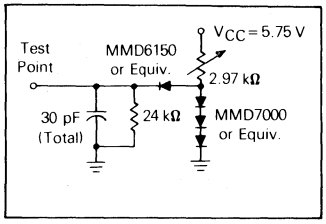


FIGURE 4 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

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.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to Interrupts section for additional information.

XTAL AND EXTAL

These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, 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 Internal Clock Generator Options section for recommendations about these inputs.

TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to Timer section for additional information about the timer circuitry.

$\overline{RESET}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to Resets section for additional information.

NUM

This pin is not for user application and must be connected to V_{SS}.

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 Inputs/Outputs section for additional information.

MEMORY

As shown in Figure 5, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC6805P6 MCU has implemented 1984 of these locations. This consists of: 1796 bytes of user ROM, 116 bytes of self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, and 2 timer registers.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 6. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly, during pulls from the stack, since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack. The remaining CPU registers are not pushed.

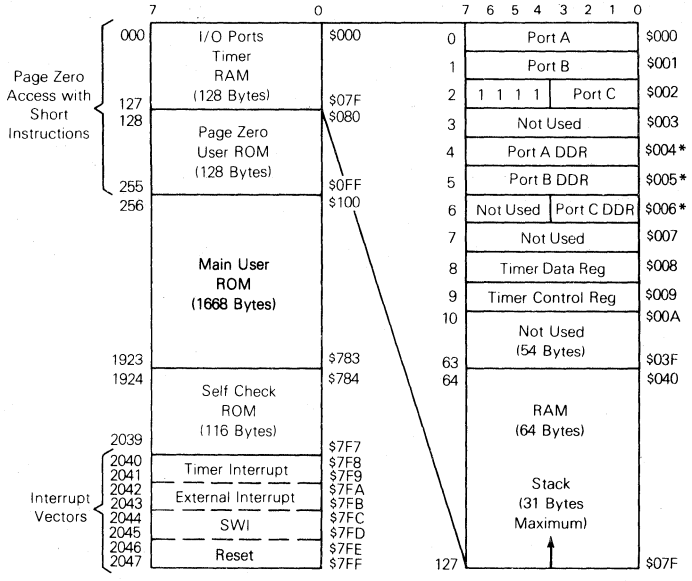
CENTRAL PROCESSING UNIT

The CPU of the M6805 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 address, data, and control buses.

REGISTERS

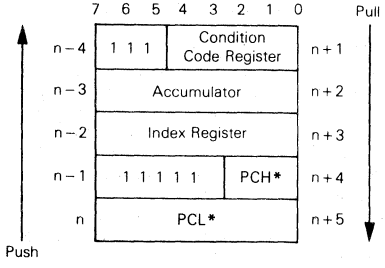
The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 7 and are explained in the following paragraphs.

FIGURE 5 — MCU ADDRESS MAP



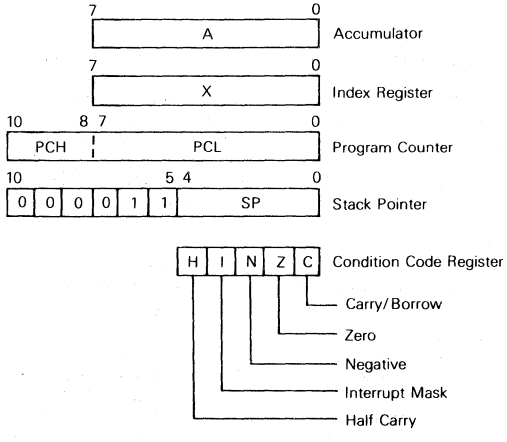
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 6 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH are stacked.

FIGURE 7 — PROGRAMMING MODEL



ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The six most significant bits of the stack pointer are permanently configured to 000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) – Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) – This bit is set to mask (disable) the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt is cleared.

NEGATIVE (N) – Used to indicate that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logical one).

ZERO (Z) – Used to indicate that the result of the last arithmetic, logical, or data manipulation was zero.

CARRY/BORROW (C) – Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The MC6805P6 MCU timer circuitry is shown in Figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9, and executing the interrupt routine; see the Interrupts section. THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE.

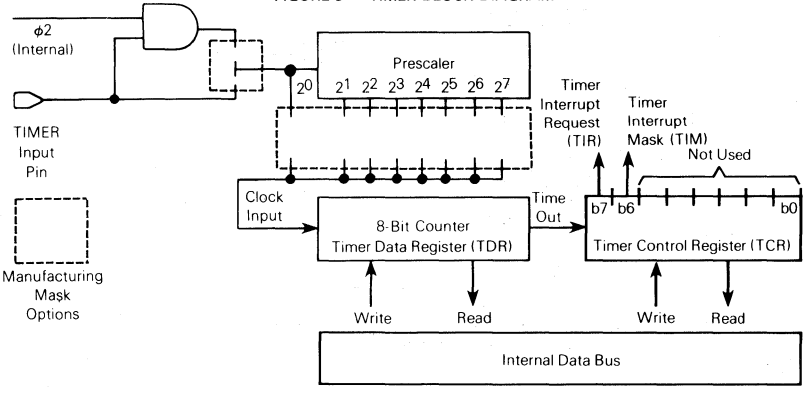
The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE: For ungated $\phi 2$ clock inputs to the timer prescaler, the TIMER

FIGURE 8 – TIMER BLOCK DIAGRAM



pin should be tied to V_{CC}.) The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture.

The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones, the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

SELF-CHECK

The self-check capability of the MC6805P6 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 7 Hz. A 9-volt level on the TIMER input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports.

RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{RESET}$), and by an optional internal low voltage detect circuit; see Figure 10. The internal circuit connected to the $\overline{RESET}$ pin consists of a Schmitt trigger which senses the $\overline{RESET}$ line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic "0" on the $\overline{RESET}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{RESET}$ pin voltage rises to V_{IRES+}. When the $\overline{RESET}$ pin voltage falls to a logical "0" for a period longer than one t_{CYC}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 11.

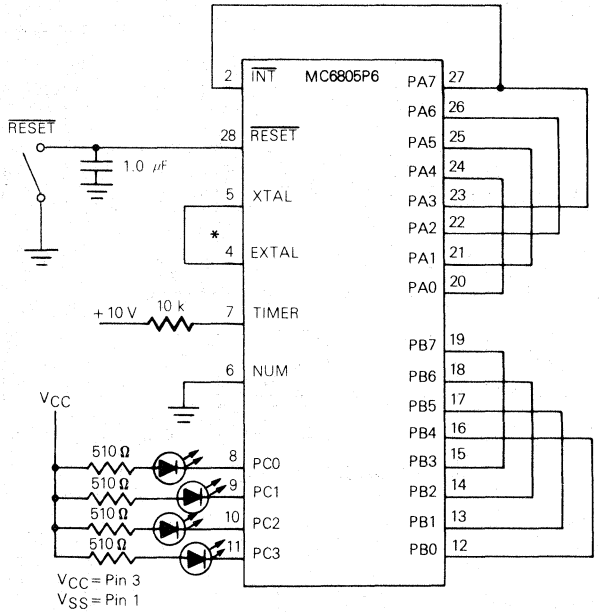
During power-up, a delay of t_{RHL} is needed before allowing the $\overline{RESET}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{RESET}$ input, as shown in Figure 12, typically provides sufficient delay. See Figure 16 under Interrupts section for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, 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 oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in Figure 13. The crystal specifications and suggested PC board layouts

FIGURE 9 — SELF-CHECK CONNECTIONS



*This connection depends on the clock oscillator user selectable mask option. Use crystal if crystal option is selected.

are given in Figure 14. A resistor selection graph is given in Figure 15.

The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

FIGURE 10 — POWER AND $\overline{\text{RESET}}$ TIMING

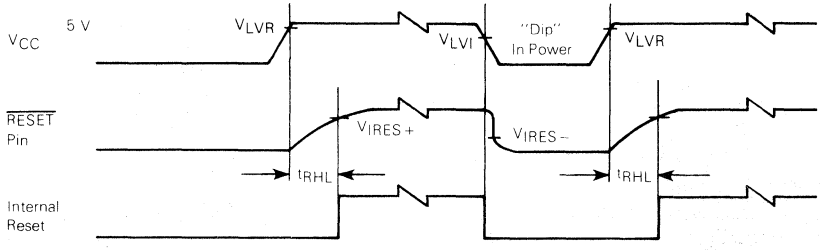


FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

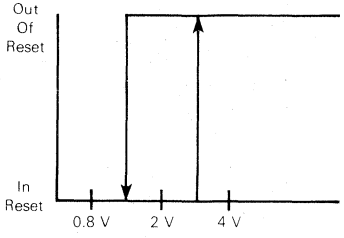


FIGURE 12 — POWER-UP RESET DELAY CIRCUIT

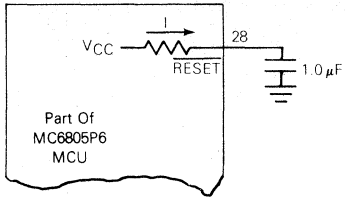
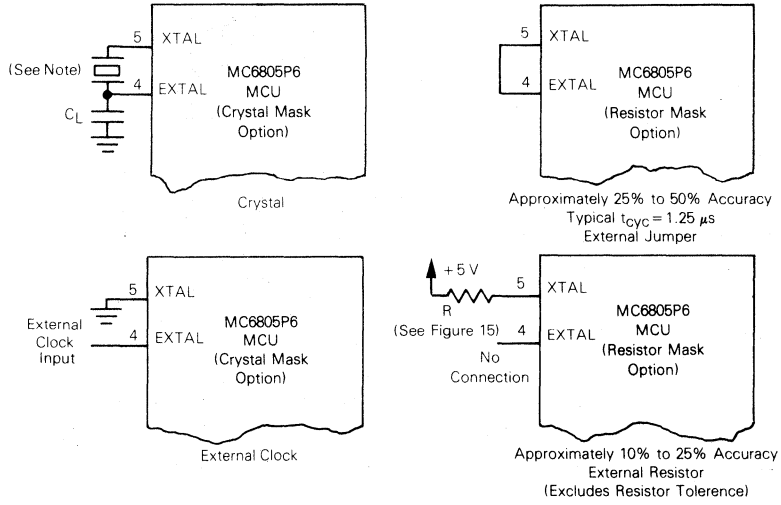
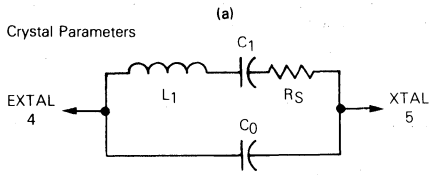


FIGURE 13 — CLOCK GENERATOR OPTIONS

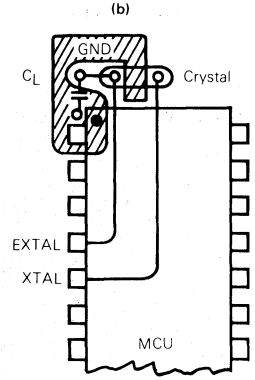


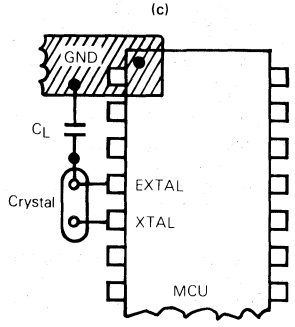
NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

FIGURE 14 — CYRSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



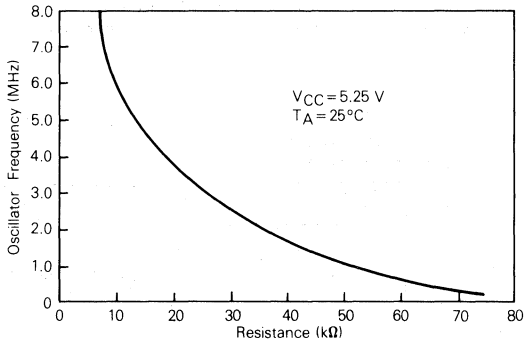
AT — Cut Parallel Resonance Crystal
C_O = 7 pF Max.
Freq. = 4.0 MHz @ C_L = 24 pF
R_S = 50 ohms Max.





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

FIGURE 15 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P6 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{CYC} periods for completion.

A flowchart of the interrupt sequence is shown in Figure 16. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{RESET}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{INT}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 17(a), for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-

FIGURE 16 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

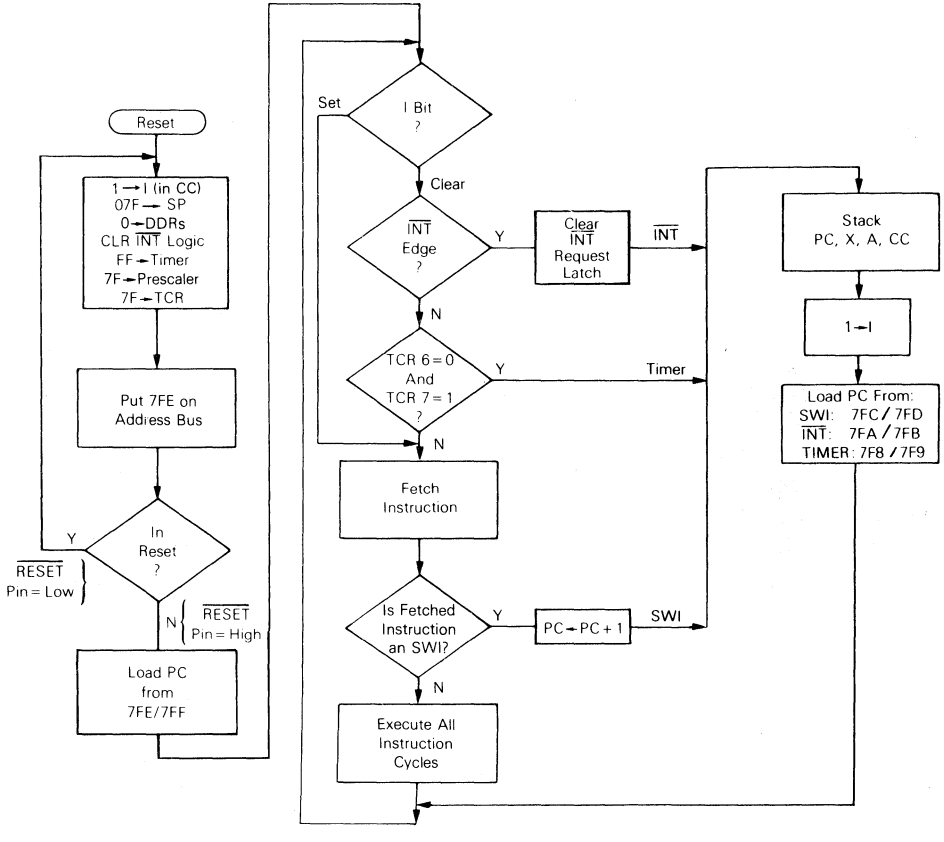
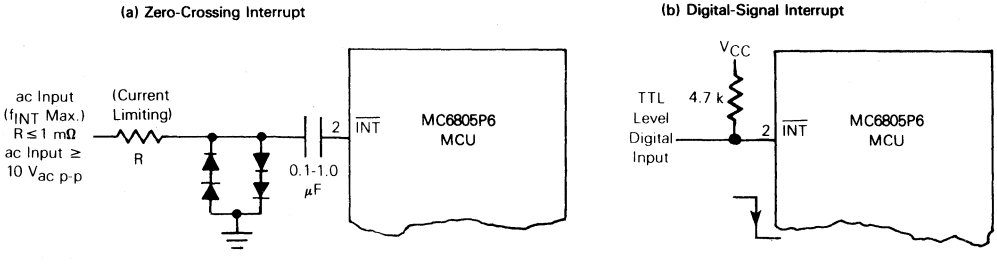


FIGURE 17 — TYPICAL INTERRUPT CIRCUITS



arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 17(b).

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. Note that if the I bit is zero SWI executes after the other interrupts. SWIs are usually used as break-points for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may also be polled with branch instructions to provide an additional input pin. 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 "1" for output or a logic "0" for input. On reset, all the DDRs are initialized to a logic "0" state to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting 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 18. When port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

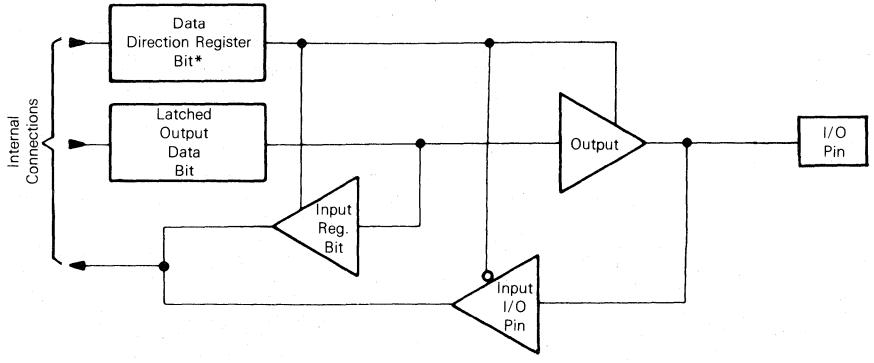
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 5 gives the address of data registers and DDRs. The register configuration is provided in Figure 19 and Figure 20 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see Figure 18) may always be written. Therefore, any write to a port writes 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").

FIGURE 18 — TYPICAL PORT I/O CIRCUITRY

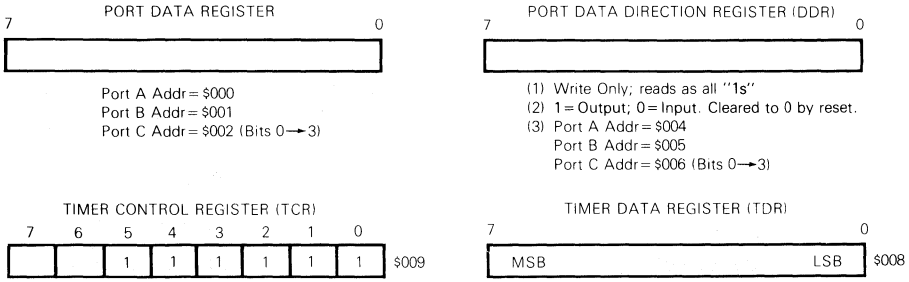


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

*DDR is a write-only register and reads as all "1s".

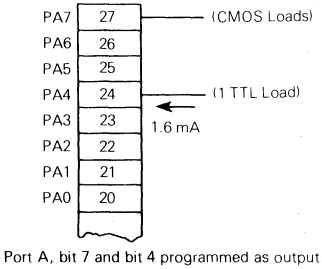
**Port A (with CMOS drive disabled), B, and C are three state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics table for complete information.

FIGURE 19 — MCU REGISTER CONFIGURATION

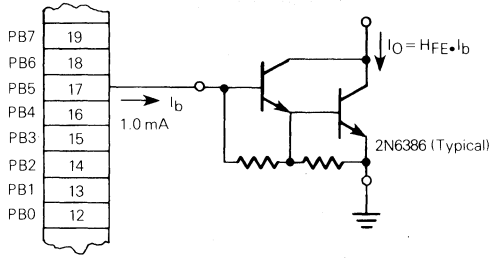


3

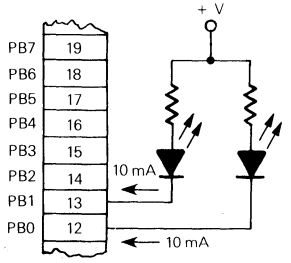
FIGURE 20(a) — TYPICAL OUTPUT MODE PORT CONNECTIONS



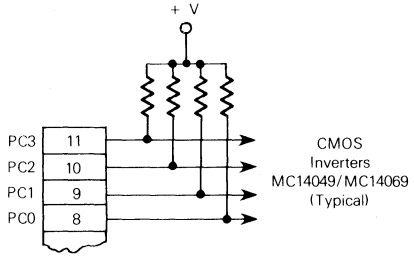
Port A, bit 7 and bit 4 programmed as output. Bit 7 driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.



Port B, bit 5 programmed as output, driving Darlington-base directly.

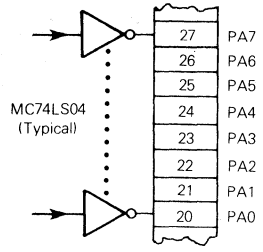


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

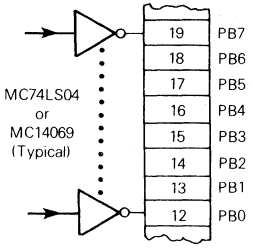


Port C, bits 0-3 programmed as output, driving CMOS loads, using external pullup resistors.

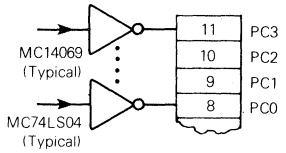
FIGURE 20(b) — TYPICAL INPUT MODE PORT CONNECTIONS



TTL driving port A directly.



CMOS or TTL driving port B directly.



CMOS and TTL driving port C directly.

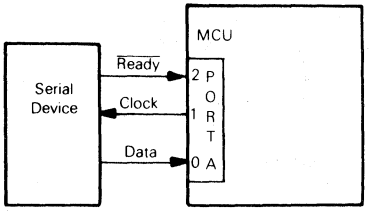
SOFTWARE

BIT MANIPULATION

The MC6805P6 MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction register, see Caution under Input/Output section), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches 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 RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 21 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, LSB 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 a RAM location.

FIGURE 21 — BIT MANIPULATION EXAMPLE




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

ADDRESSING MODES

The MC6805P6 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family User's Manual.

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.

IMMEDIATE — In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory with a single 2-byte instruction. This includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE — The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 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.

INDEXED, NO OFFSET — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET — In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET — In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the Input/Output section.

BIT TEST AND BRANCH — The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit and condition (set or clear) which is to be tested is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the condition code register. See Caution under the Input/Output section.

INHERENT — In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC6805P6 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS — Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 1.

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 (see Caution under Input/Output section). The test for negative or zero (TST) instruction is included in read-modify-write instructions though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 3.

BIT MANIPULATION INSTRUCTIONS — These instructions are used on any bit in the first 256 bytes of the memory (see Caution under Input/Output section). One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS — The control instructions control the MCU operations during program execution. Refer to Table 5.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 6.

OPCODE MAP SUMMARY — Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

		Relative Addressing Mode		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0....7)	—	—	—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0....7)	—	—	—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0....7)	10 + 2 • n	2	7	—	—	—
Clear bit n	BCLR n (n = 0....7)	11 + 2 • n	2	7	—	—	—

TABLE 5 — CONTROL INSTRUCTIONS

		Inherent		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	●	^	^	^
ADD		X	X	X		X	X	X			^	●	^	^	^
AND		X	X	X		X	X	X			●	●	^	^	●
ASL	X		X			X	X				●	●	^	^	^
ASR	X		X			X	X				●	●	^	^	^
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	^	^	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	^
BRSET										X	●	●	●	●	^
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	^	^	^
COM	X		X			X	X				●	●	^	^	1
CPX		X	X	X		X	X	X			●	●	^	^	^
DEC	X		X			X	X				●	●	^	^	●
EOR		X	X	X		X	X	X			●	●	^	^	●
INC	X		X			X	X				●	●	^	^	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEQ	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected

TABLE 6 — INSTRUCTION SET (CONTINUED)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols:

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

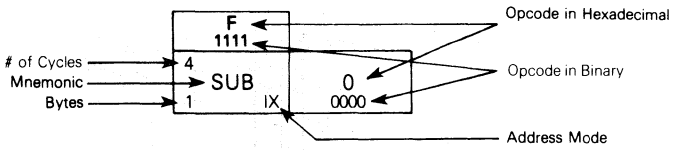
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the MCU it is necessary to first contact your local Motorola representative or Motorola distributor.

EPROMs

The MC68705P3 EPROM MCU programmed with the customer program may be used to submit the ROM pattern. Note that while the MC6805P6 has 1796 bytes of ROM, the MC68705P3 contains 1.8K of EPROM memory.

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure is illustrated below:

XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

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 on blank EPROM from the data file used to create the custom mask and aid in the verification process.

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 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by Motorola Quality Assurance and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name on the disk with a felt-tip pen. The minimum MDOS system files as well as the absolute binary object file (filename L0 type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: filename, LX (EXORciser loadable format) and filename, SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

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

GENERIC INFORMATION

Package Type	Internal Clock Frequency (MHz)	Temperature	Generic Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6805P6L
	1.5	0°C to 70°C	MC68A05P6L
	2.0	0°C to 70°C	MC68B05P6L
Plastic P Suffix	1.0	0°C to 70°C	MC6805P6P
	1.5	0°C to 70°C	MC68A05P6P
	2.0	0°C to 70°C	MC68B05P6P
Cerdip P Suffix	1.0	0°C to 70°C	MC6805P6S
	1.5	0°C to 70°C	MC68A05P6S
	2.0	0°C to 70°C	MC68B05P6S

MC6805P6 CUSTOM ORDERING INFORMATION

Date _____ Customer PO Number _____

Customer Company _____ Motorola Part Numbers

Address _____ MC _____

SC _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Timer Clock Source
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Internal Clock Frequency
- ☐ 0.1 to 1.0 MHz
- ☐ 0.1 to 1.5 MHz
- ☐ 0.1 to 2.0 MHz

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Port B Output Drive
- ☐ TTL
- ☐ Open Drain

Pattern Media (All other media requires prior factory approval.)
- ☐ EPROMs (MCM2716 or MCM2532)
- ☐ EPROM MCU (MC68705P3)
- ☐ Floppy Disk
- ☐ Other _____

Clock Freq. _____

Temp. Range _____ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C

Marking Information (12 Characters Maximum)

Title _____

Signature _____

MOTOROLA

Advance Information

3

MC68(7)05R/U SERIES

8-BIT MICROCOMPUTERS

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

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

SECTION 1
INTRODUCTION

The M6805 Family of low-cost single-chip microcomputers was designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set. This rapidly expanding family includes a number of memory and package sizes with various I/O functions in both HMOS and CMOS.

This document describes the eight 8-bit high-density N-channel silicon-gate microcomputers which comprise the MC68(7)05R/U series. These devices are listed below:

MC6805R2	MC6805R3	MC68705R3	MC68705R5
MC6805U2	MC6805U3	MC68705U3	MC68705U5

These eight devices are 8-bit high-density N-channel silicon-gate microcomputers. They are available in 40-pin dual-in-line packages.

1.1 DEVICE FEATURES

The following tables summarize the hardware and software features of each device. Differences between the devices will be highlighted throughout this document when applicable.

HARDWARE FEATURES

	MC6805R2	MC6805R3	MC6805U2	MC6805U3	MC68705R3	MC68705U3	MC68705R5	MC68705U5
24 Bidirectional I/O Lines	X	X	X	X	X	X	X	X
Eight Input-Only Lines	X	X	X	X	X	X	X	X
A/D Converter	X	X	–	–	X	–	X	–
User ROM (Bytes)	2048	3776	2048	3776	–	–	–	–
User EPROM (Bytes)	–	–	–	–	3776	3776	3776	3776
RAM (Bytes)	64	112	64	112	112	112	112	112
Self-Check Mode	X	X	X	X	–	–	–	–
Zero/Crossing Detect/Interrupt	X	X	X	X	X	X	X	X
Timer with 7-Bit Prescaler	X	X	X	X	X	X	X	X
Programmable Prescaler	–	–	X	X	X	X	X	X
5-Volt Single Supply	X	X	X	X	X	X	X	X
Memory Mapped I/O	X	X	X	X	X	X	X	X
On-Chip EPROM Programmer	–	–	–	–	X	X	X	X
EPROM Security Feature	–	–	–	–	–	–	X	X

SOFTWARE FEATURES

	MC6805R2	MC6805R3	MC6805U2	MC6805U3	MC68705R3	MC68705U3	MC68705R5	MC68705U5
Addressing Modes	10	10	10	10	10	10	10	10
Byte Efficient Instruction Set	X	X	X	X	X	X	X	X
True Bit Manipulation	X	X	X	X	X	X	X	X
Bit Test and Branch Instructions	X	X	X	X	X	X	X	X
Versatile Interrupt Handling	X	X	X	X	X	X	X	X
Versatile Index Register	X	X	X	X	X	X	X	X
Powerful Indexed Addressing for Tables	X	X	X	X	X	X	X	X
Full Set of Conditional Branches	X	X	X	X	X	X	X	X
Memory Usable as Registers/Flags	X	X	X	X	X	X	X	X
Single Instruction Memory Examine/Change	X	X	X	X	X	X	X	X
User Callable Self-Check Subroutines	X	X	X	X	–	–	–	–
Complete Development System Support on EXORciser	X	X	X	X	X	X	X	X
Supported by EPROM Version	X	X	X	X	–	–	–	–

1.2 HARDWARE

Every M6805 Family microcomputer contains hardware common to all versions, plus a combination of options unique to a particular version. Figures 1-1 through 1-6 illustrate the unique options available on the eight versions described in this document.

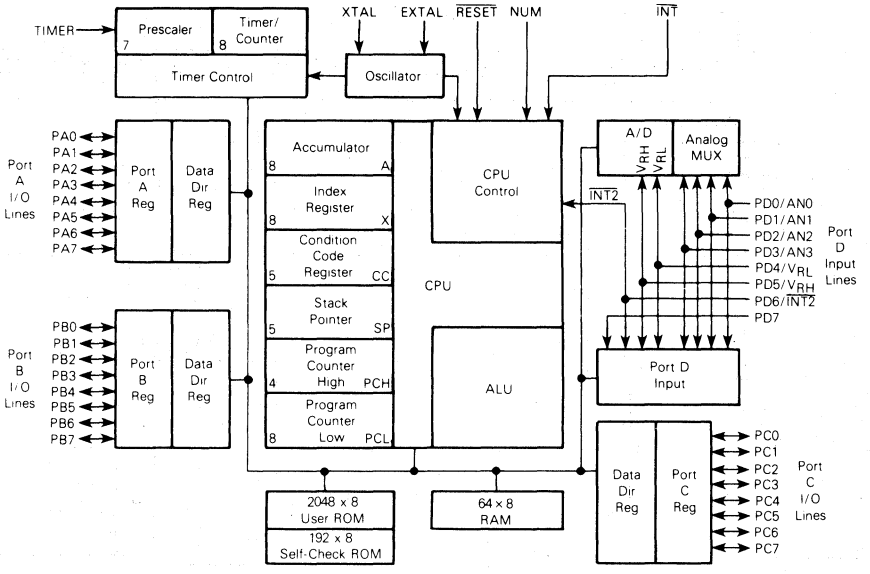


Figure 1-1. MC6805R2 Block Diagram

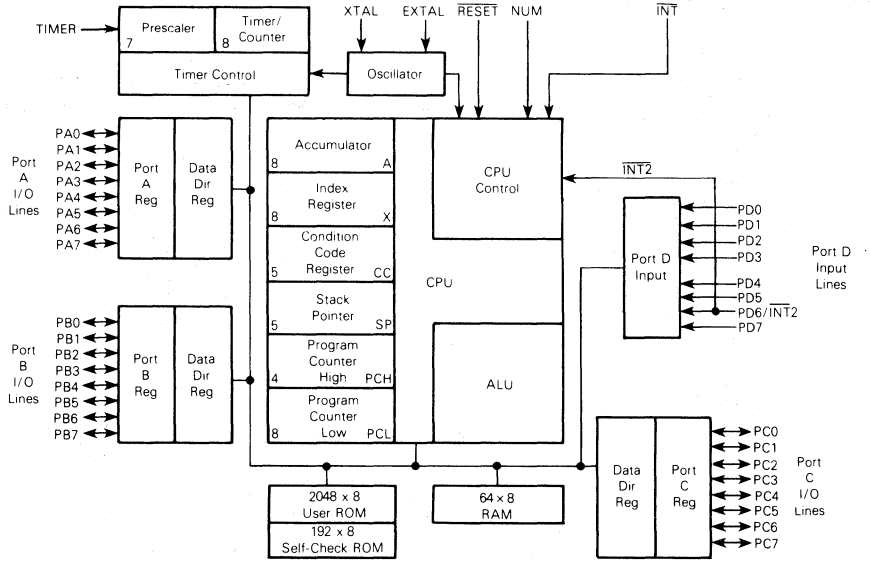


Figure 1-2. MC6805U2 Block Diagram

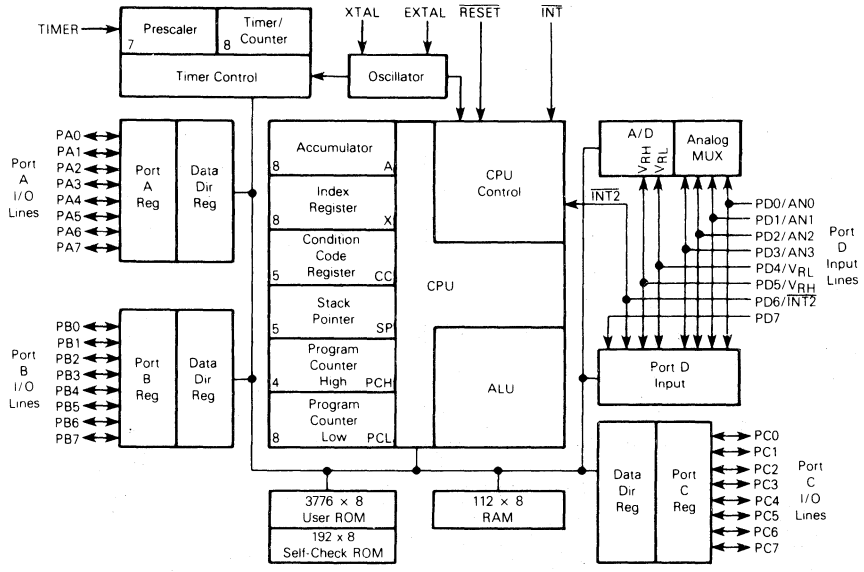


Figure 1-3. MC6805R3 Block Diagram

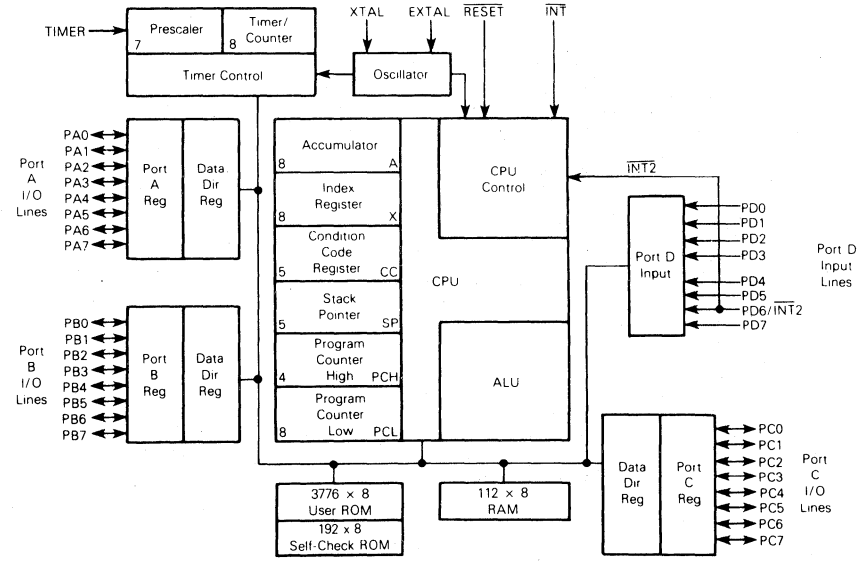


Figure 1-4. MC6805U3 Block Diagram

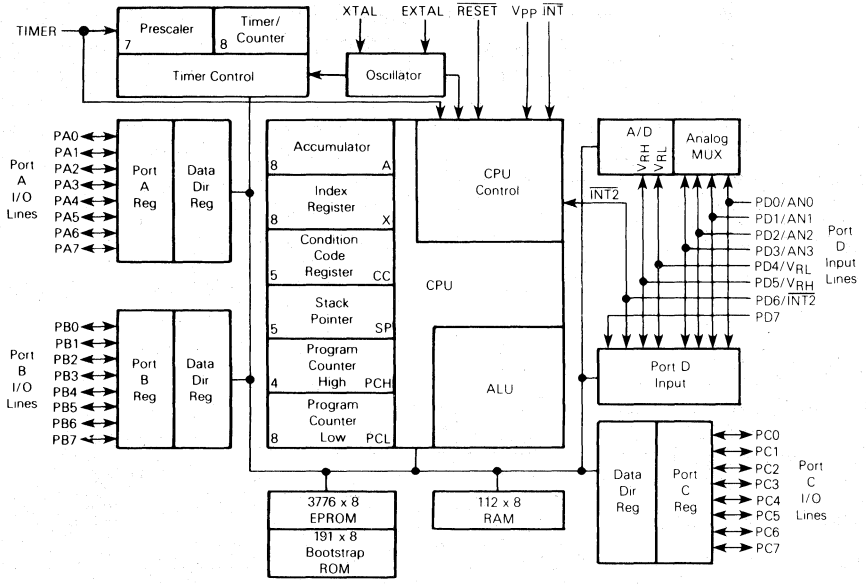


Figure 1-5. MC68705R3 and MC68705R5 Block Diagram

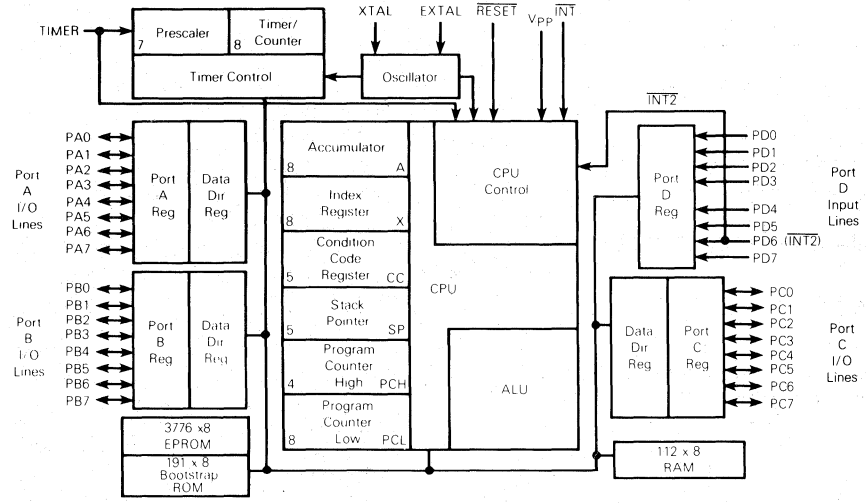


Figure 1-6. MC68705U3 and MC68705U5 Block Diagram

SECTION 2
SIGNAL DESCRIPTION

The following paragraphs contain brief descriptions of the input and output signals. Where applicable reference has been made to other sections that contain more detail about the function being performed.

2.1 V_{CC} AND V_{SS}

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

2.2 $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the microcomputer. Refer to **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE** for additional information regarding the interrupt operation.

2.3 EXTAL AND XTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on user selectable manufacturing mask option, can be connected to these pins to provide a system clock with various degrees of stability/cost tradeoffs. Lead length and stray capacitance on these two pins should be minimized. Refer to **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE** for recommendations about these inputs.

2.4 TIMER

This pin is used as an external input to control the internal timer/counter circuitry. On the MC68705R3, MC68705U3, MC68705R5, and MC68705U5 versions, this pin also detects a higher voltage level used to initiate the bootstrap program for loading the internal EPROM (see **SECTION 10 SOFTWARE**). On the MC6805R2, MC6805U2, MC6805R3, and MC6805U3 this pin also detects a higher voltage level used to initiate the self-test program (see **SECTION 6 SELF CHECK**).

Refer to **SECTION 5 TIMER** for more detailed information about the timer circuitry.

2.5 $\overline{RESET}$

This pin has a Schmitt trigger input and an on-chip pullup. The microcomputer can be reset by pulling $\overline{RESET}$ low. Refer to **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE** for additional information.

2.6 NUM (NON-USER MODE)

Pin 7 of the MC6805R2 and MC6805U2 is identified as NUM (non-user mode). This pin is not for user application and must be connected to V_{SS}.

2.7 V_{PP}

This pin is used when programming the EPROM versions (MC68705R3, MC68705U3, MC68705R5, and MC68705U5). By applying the programming voltage to this pin, one of the requirements is met for programming the EPROMs. In normal operation, this pin is connected to V_{CC}. Refer to **SECTION 9 MASK OPTIONS AND PROGRAMMING** for more detailed information.

2.8 INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programming as either inputs or outputs under software control of the data direction registers.

For the MC6805U2, MC6805U3, MC68705U3, and MC68705U5 port D is for digital input only and bit 6 may be used for a second interrupt ($\overline{INT2}$). Refer to **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE** and **SECTION 8 INPUT/OUTPUT CIRCUITRY AND ANALOG-TO-DIGITAL CONVERTER** for additional information.

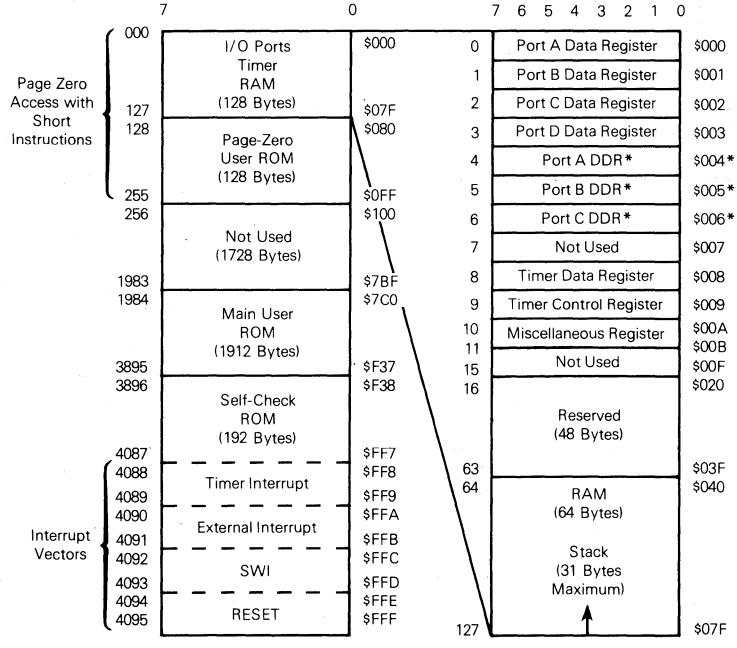
For the MC6805R2, MC6805R3, MC68705R3, and MC68705R5 port D has up to four analog inputs, plus two voltage reference inputs when the analog-to-digital converter is used (PD5/V_{RH}, PD4/V_{RL}) and an $\overline{INT2}$ input. All port D lines can be read directly and used as binary inputs. If any analog input is used, then the voltage reference pins (PD5/V_{RH}, PD4/V_{RL}) must be used in the analog mode. Refer to **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE** and **SECTION 8 INPUT/OUTPUT CIRCUITRY AND ANALOG-TO-DIGITAL CONVERTER** for additional information.

SECTION 3
MEMORY CONFIGURATIONS

Each member of the MC68(7)05R/U series of microcomputers is capable of addressing 4096 bytes of memory and I/O registers. The memory maps for the eight versions of the M6805 Family described in this document are shown in Figures 3-1 through 3-6. The amount of ROM, EPROM, and RAM for each device is detailed in **1.1 DEVICE FEATURES.**

3.1 MC6805U2 MEMORY MAP

The memory map for the MC6805U2 is shown in Figure 3-1. From $FF8 to $FFF are the interrupt and RESET vectors. A self-check ROM occupies 192 bytes from $F38 to $FF7. The user ROM is



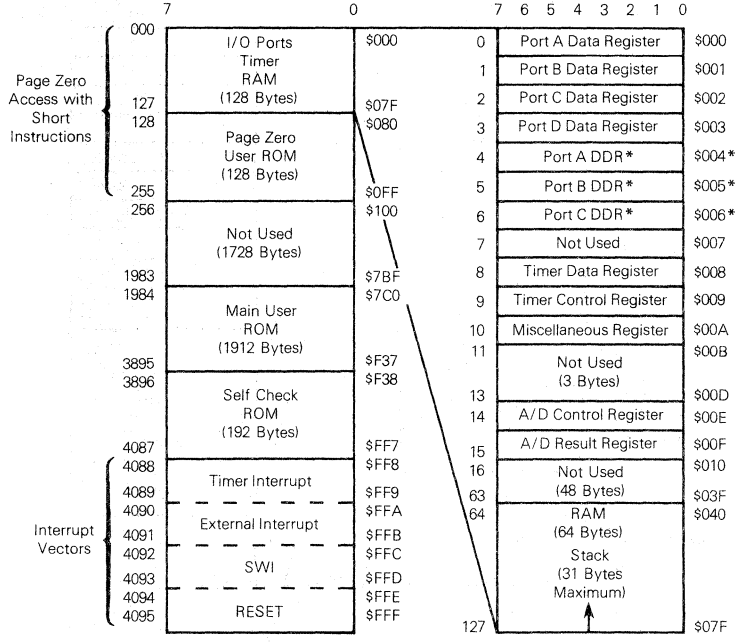
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 3-1. MC6805U2 Memory Map

divided into two portions located from $080 to $0FF and $7C0 to $F37. The portioning allows 128 bytes of ROM to be addressed with direct instructions. A RAM area of 64 bytes occupies $040 to $07F. Only the 31 bytes from $061 to $07F can be used for the stack RAM due to the limitation imposed by the 5-bit stack pointer. The data direction, peripheral data, timer, and miscellaneous registers are located from $000 to $00F.

3.2 MC6805R2 MEMORY MAP

The memory map for the MC6805R2 is shown in Figure 3-2 and is identical to the MC6805U2 except that two additional registers, the analog-to-digital control register and the analog-to-digital result register, have been added at locations $00E and $00F, respectively.

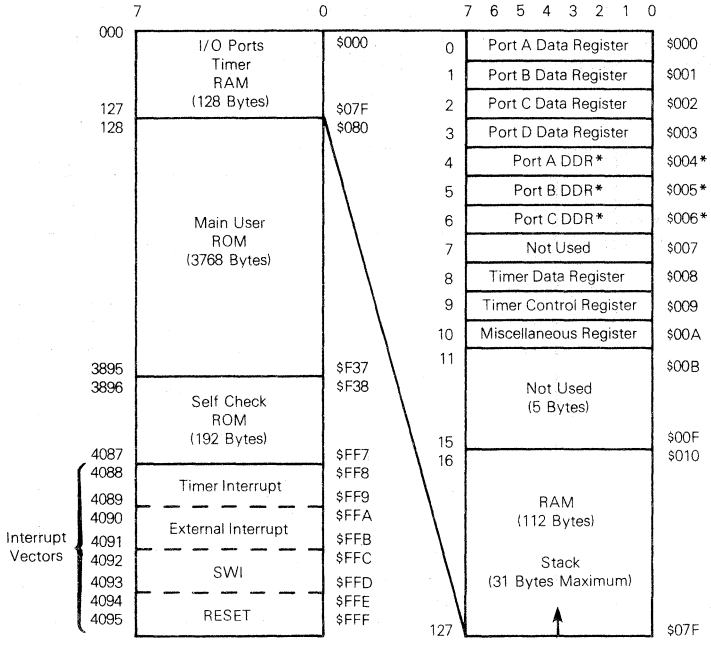


* Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 3-2. MC6805R2 Memory Map

3.3 MC6805U3 MEMORY MAP

The memory map for the MC6805U3 is shown in Figure 3-3. The MC6805U3 has an expanded ROM and RAM area over the MC6805U2. The user ROM in the MC6805U3 consists of 3768 bytes from $080 to $F37. The RAM is expanded to 112 bytes from $010 to $07F. All other registers remain identical to the MC6805U2. The 5-bit stack pointer still allows only 31 bytes of RAM to be used as stack area.

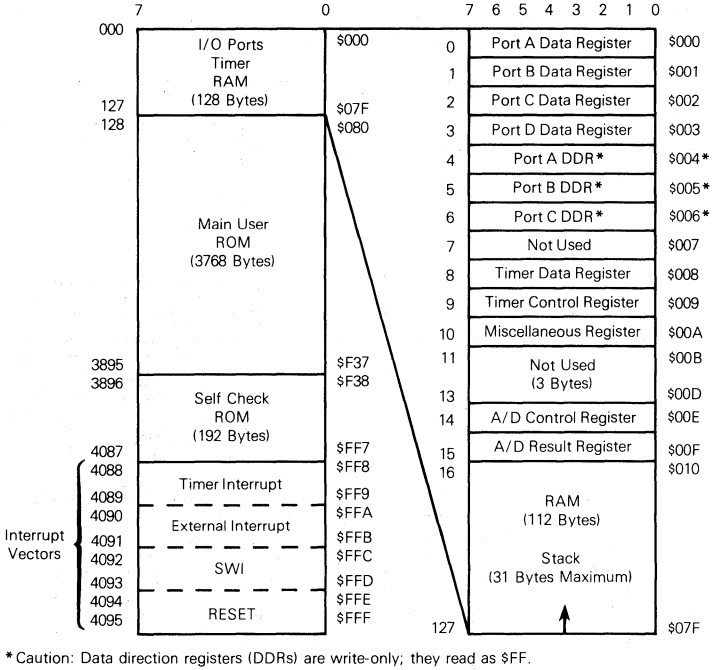


*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 3-3. MC6805U3 Memory Map

3.4 MC6805R3 MEMORY MAP

The memory map for the MC6805R3 is shown in Figure 3-4 and is identical to the MC6805U3 except that two additional registers, the analog-to-digital control register and the analog-to-digital result register, have been added at locations $00E and $00F, respectively.



*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

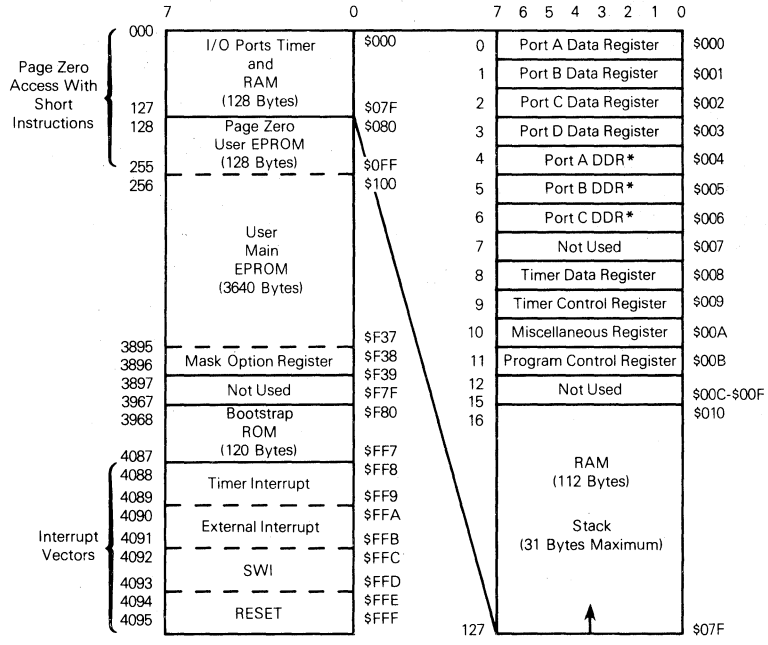
Figure 3-4. MC6805R3 Memory Map

3.5 MC68705U3 and MC68705U5 MEMORY MAP

The memory maps for the MC68705U3 and MC68705U5 are shown in Figure 3-5 and are identical to the masked programmed equivalent, the MC6805U3, with respect to RAM, ROM, I/O, special purpose registers, and interrupt and RESET vectors. The ROM area ($080 to $F37) of the MC68705U3 and MC68705U5 is an ultraviolet erasable EPROM.

A bootstrap ROM is located between $F39 and $FF7 which allows the MC68705U3 and MC68705U5 to program their own EPROMs. The bootstrap is a mask programmed ROM.

At location $F38 is the mask option register (MOR) which is an EPROM byte. It allows the user to set up the MC68705U3 and the MC68705U5 for a crystal or RC oscillator, set the timer prescaler, the clock source, etc. In addition, the mask option register allows the user to select the secure mode offered by the MC68705U5.



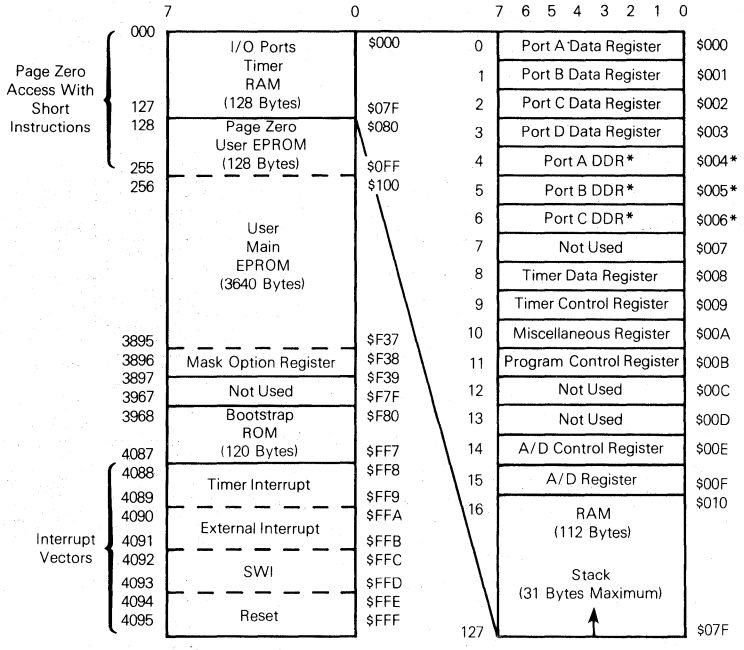
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 3-5. MC68705U3 and MC68705U5 Memory Map

3.6 MC68705R3 and MC68705R5 MEMORY MAP

The memory maps for the MC68705R3 and MC68705R5 are shown in Figure 3-6 and are identical to the MC68705U3 and MC68705U5 except that two additional registers, the analog-to-digital control register and the analog-to-digital result register have been added at locations $00E and $00F, respectively.

The MC68705U3/MC68705U5 and MC68705R3/MC68705R5 are intended to exactly emulate the MC6805U3 and MC6805R3 respectively.

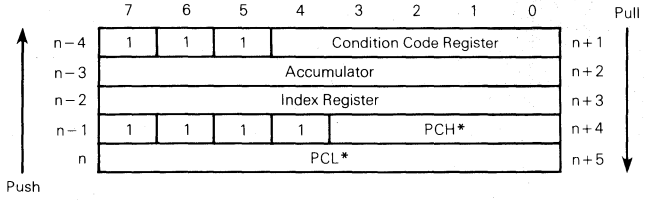


*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

Figure 3-6. MC68705R3 and MC68705R5 Memory Map

3.7 SHARED STACK AREA

The shared stack area (RAM locations $061-$07F) is used during the processing of an interrupt or subroutine call to save the contents of the central processing unit state. The register contents are pushed onto the stack in the order shown in Figure 3-7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first, then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed. The shared stack area must be used with care when it is used for data storage or temporary work locations to protect it from being overwritten due to stacking from an interrupt or subroutine call.



*For subroutine calls, only PCH and PCL are stacked.

Figure 3-7. Interrupt Stacking Order

3.8 CENTRAL PROCESSING UNIT

The central processing unit for the M6805 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 address, data, and control buses.

SECTION 4
PROGRAMMABLE REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 4-1 and are explained in the following paragraphs.

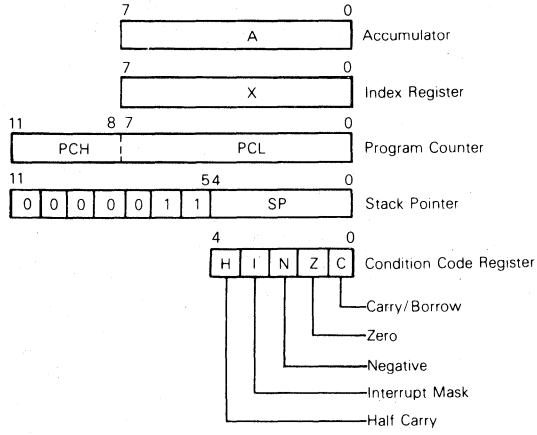


Figure 4-1. Programming Model

4.1 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

4.2 INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.

4.3 PROGRAM COUNTER (PC)

The program counter is a 12-bit register that contains the address of the next bye to be fetched.

4.4 STACK POINTER (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

4.5 CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

4.5.1 Half Carry (H)

Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

4.5.2 Interrupt (I)

When this bit is set, the timer and external interrupts ($\overline{INT}$ and $\overline{INT2}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

4.5.3 Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

4.5.4 Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

4.5.5 Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

SECTION 5
TIMER

The following paragraphs describe the timer circuitry for the eight versions of the M6805 Family found in this document. Note that while each timer consists of an 8-bit software programmable counter driven by a 7-bit prescaler there are three distinctly different configurations (Figures 5-1, 5-2, and 5-3).

5.1 MC6805R2/MC6805U2 TIMER CIRCUITRY

The timer circuitry for the MC6805R2 and MC6805U2 microcomputers is shown in Figure 5-1. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (or prescaler output). When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine (see **SECTION 7 RESET, CLOCK, AND INTERRUPT STRUCTURE**). *The timer interrupt request bit must be cleared by software.* The TIMER and $\overline{INT2}$ share the same interrupt vector. *The interrupt routine must check the request bits to determine the source of the interrupt.*

The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal phase two signal. Three machine cycles are required for a change in state of the TIMER pin to

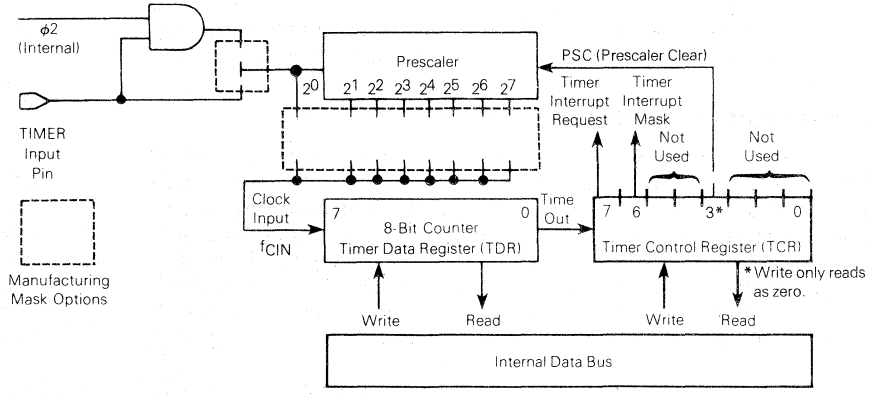


Figure 5-1. MC6805R2/MC6805U2 Timer Block Diagram

decrement the timer prescaler. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows (assumes 50/50 duty cycle for a given period):

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 nanoseconds times two).

When the phase two signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. The source of the clock input is one of the mask options that is specified before manufacture of the MCU.

NOTE

For ungated phase two clock input to the timer prescaler, the TIMER pin should be tied to V_{CC}.

A prescaler option, divide by 2^n, can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture. To avoid truncation errors, the prescaler is cleared when bit 3 of the timer control register is written to a logic one (this bit always reads a logic zero).

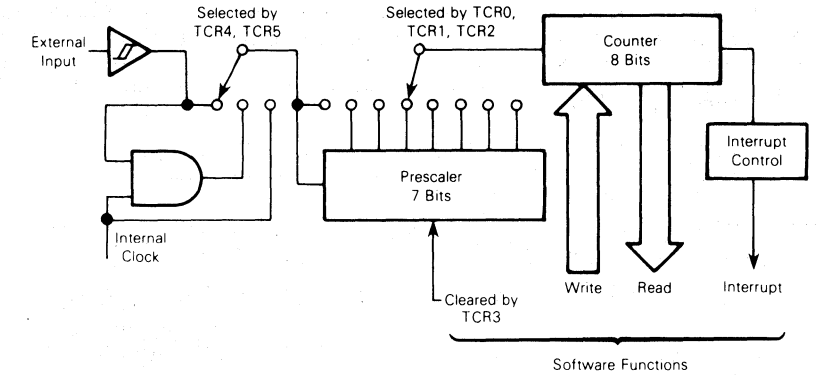
The timer continues to count past zero, falling through to $FF from $00 and then continuing the countdown. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred, and not disturb the counting process.

At power up or reset, the prescaler and counter are initialized with all logic ones; the timer interrupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set.

5.2 MC6805R3/MC6805U3 TIMER CIRCUITRY

The timer circuitry for the MC6805R3/MC6805U3 microcomputers is shown in Figure 5-2. The timer contains a single 8-bit software programmable counter with 7-bit software selectable prescaler. The counter may be preset under program control and decrements toward zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer interrupt vector from locations $FF8 and $FF9 in order to begin servicing the interrupt.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt request bit remains set until cleared by the software. If a write occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR = 1).



NOTES:
1. The prescaler and 8-bit counter are clocked on the rising edge of the internal clock (phase two) or external input.
2. The counter is written to during data strobe (DS) and counts down continuously.

Figure 5-2. MC6805R3/MC6805U3 Timer Block Diagram

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals one, which allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode, depending on the value written to the TCR4 and TCR5 control bits. Refer to **5.2.5 Timer Control Register (TCR)** for further information.

5.2.1 Timer Input Mode 1

If TCR5 and TCR4 are both programmed to a zero, the input to the timer is from an internal clock and the external TIMER input is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock.

5.2.2 Timer Input Mode 2

With TCR5=0 and TCR4=1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse widths.

5.2.3 Timer Input Mode 3

If TCR5=1 and TCR4=0, then all inputs to the timer are disabled.

5.2.4 Timer Input Mode 4

If TCR5=1 and TCR4=1, the internal clock input to the timer is disabled and the TIMER input pin

becomes the input to the timer. The external TIMER pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

5.2.5 Timer Control Register (TCR)

TCR7	TCR6	TCR5	TCR4	TCR3*	TCR2	TCR1	TCR0	$009

*Write only (read as zero)

TCR7 — Timer interrupt request bit: indicates the timer interrupt when it is a logic one.
1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, or program control (write).

TCR6 — Timer interrupt mask bit: inhibits the timer interrupt to the processor, when this bit is a logic one.
1 — Set on external reset, power-on reset, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock (unaffected by $\overline{\text{RESET}}$).
1 — Select external clock source. Set to a logic one on external reset, power-on reset, or program control.
0 — Select internal clock source (phase two). Cleared under program control.

TCR4 — External enable bit: control bit used to enable the external TIMER pin (unaffected by $\overline{\text{RESET}}$).
1 — Enable external TIMER pin. Set on external reset, power-on reset, or program control.
0 — Disable external TIMER pin. Cleared under program control.

TCR5	TCR4	
0	0	Internal clock to timer
1	1	AND of internal clock and TIMER pin to timer
1	0	Input to timer disabled
1	1	TIMER pin to timer

TCR3 — Timer prescaler reset bit: writing a one to this bit resets the prescaler to zero. A read of this location always indicates a zero (unaffected by $\overline{\text{RESET}}$).

TCR2, TCR1, and TCR0 — Prescaler address bits: decoded to select one of eight outputs of the prescaler (set to all ones by $\overline{\text{RESET}}$).

PRESCALER

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8

TCR2	TCR1	TCR0	Result
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

5.3 MC68705R3/MC68705U3 AND MC68705R5/MC68705U5 TIMER CIRCUITRY

The timer for the MC68705R3 and MC68705U3 microcomputers is shown in Figure 5-3 and the timer for the MC68705R5 and MC68705U5 microcomputers is shown in Figure 5-4. The timer for all four devices contains an 8-bit software programmable counter which is driven by a 7-bit prescaler with one-of-eight selectable outputs. Various timer clock sources may be selected ahead of the prescaler and counter. The timer selections are made via the timer control register (TCR) and/or the mask option register (MOR). The TCR also contains the interrupt control bits. Note that the MC68705R5 and MC68705U5 offer a secure/non-secure mode option which is implemented through bit 3 of the mask option register (refer to **SECTION 9 MASK OPTIONS AND PROGRAMMING** for further information regarding the secure/non-secure mode option).

The 8-bit counter may be loaded under program control and is decremented toward zero by the counter input frequency (f_{CIN}) input (output of the prescaler selector). Once the 8-bit counter has decremented to zero, it sets the TIR (timer interrupt request) bit 7 (b7 of TCR). The TIM (timer interrupt mask) bit (b6) can be software set to inhibit the interrupt request, or software cleared to pass the interrupt request to the processor. When the I bit in the condition code register is cleared, the processor receives the timer interrupt. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9, and executing the interrupt routine. The processor is sensitive to the level of the timer interrupt request line; therefore if the interrupt is masked, the TIR bit may be cleared by software (e.g., BCLR) without generating an interrupt. The TIR bit *must* be cleared by the timer interrupt service routine to clear the timer interrupt request.

The timer interrupt and $\overline{INT2}$ share the same interrupt vector. The interrupt routine thus must check the two request bits to determine the source of the interrupt.

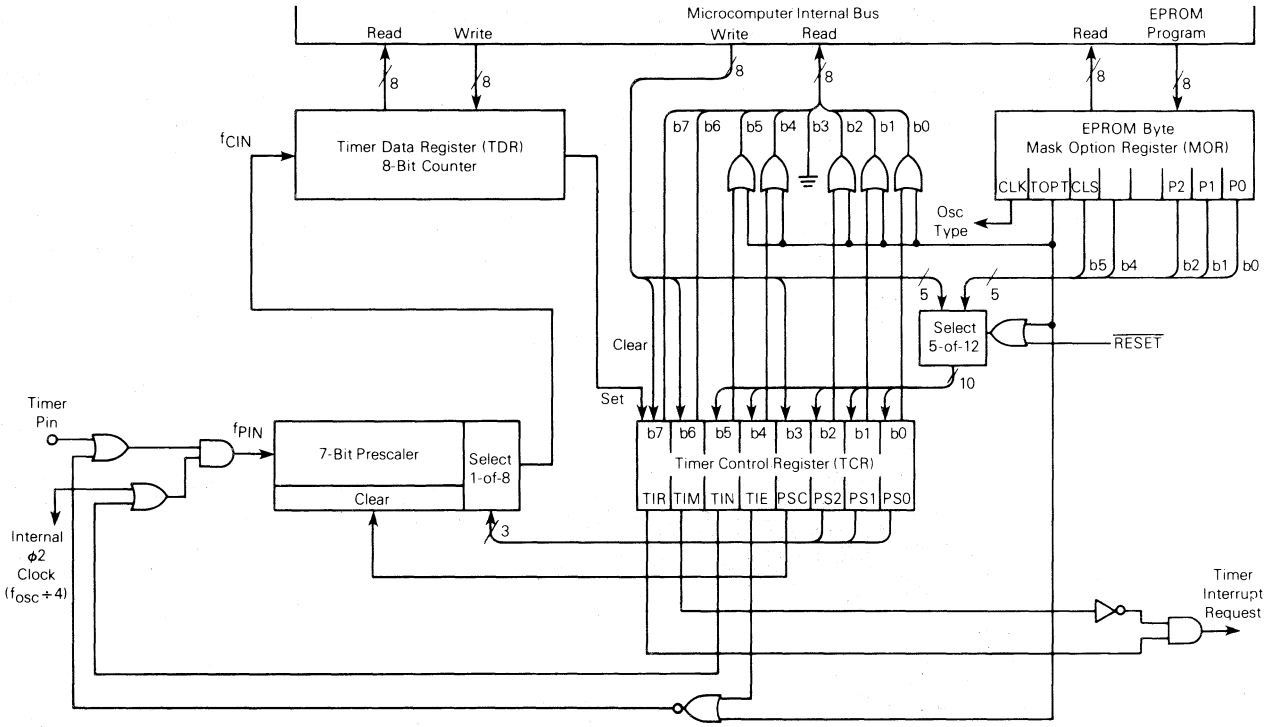
The counter continues to count (decrement) by falling through to $FF from zero. Thus, the counter can be read at any time by the processor without disturbing the count. This allows a program to determine the length of time since the occurrence of a timer interrupt and does not disturb the counting process.

The clock input to the timer can be from an external source (decrementing the counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal phase two signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows (assumes 50/50 duty cycle for a given period):

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

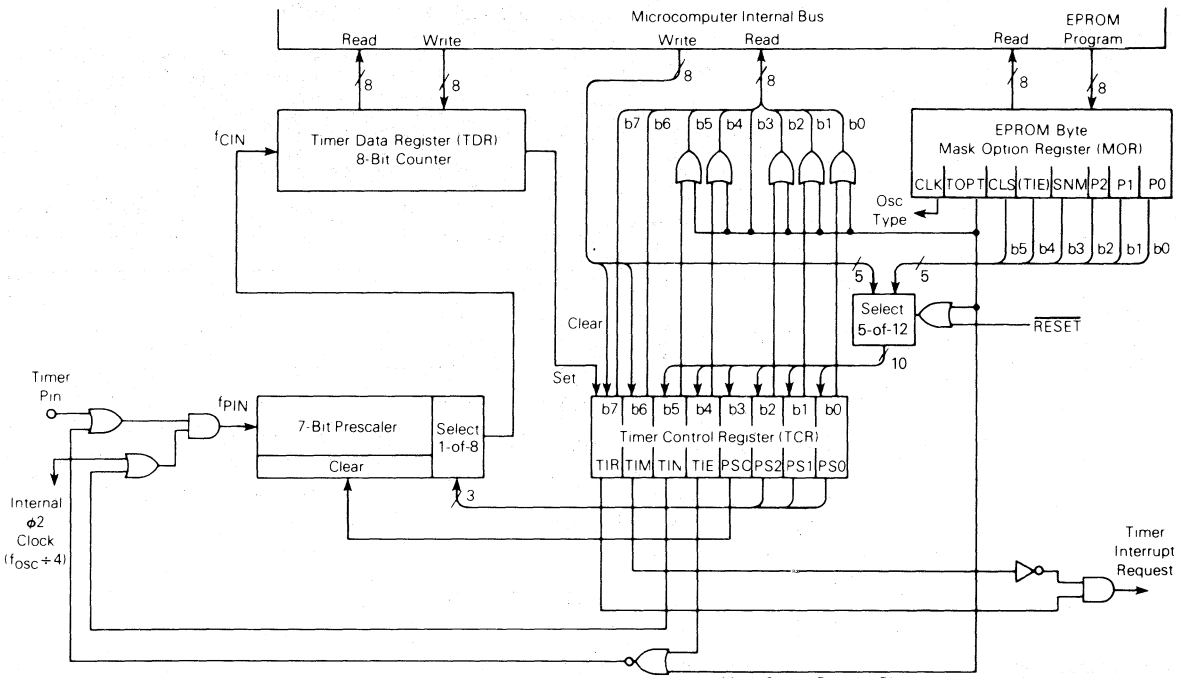
The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 nanoseconds twice).

When the phase two signal is used as the source, it can be gated by an input applied to the TIMER pin allowing the user to easily perform pulse width measurements. The source of the clock input is selected via the TCR or the MOR as described later.



NOTE: The TOPT bit in the mask option register selects whether the timer is software programmable via the timer control register or emulates the mask programmable parts via the MOR EPROM byte.

Figure 5-3. MC68705R3/MC68705U3 Timer Block Diagram



f_{PIN} – Prescaler Input Frequency
f_{CIN} – Counter Input Frequency

Timer Control Register Bits:
- TIR – Timer Interrupt Request Status
- TIM – Timer Interrupt Mask
- TIN – Timer Input Select
- TIE – Timer External Input Enable
- PSC – Prescaler Clear
- PS2, PS1, PS0 – Prescaler Select

Mask Option Register Bits:
- CLK – Clock Oscillator Type
- TOPT – Timer Mask/Programmable Option
- CLS – Timer Clock Source
- (TIE) – (Timer External Input Enable)
- SNM – Secure/Non-Secure Mode Option
- P2, P1, P0 – Prescaler Option

NOTES: The TOPT bit in the mask option register selects whether the timer is software programmable via the timer control register or emulates the mask programmable parts via the MOR PROM byte.

The TIE bit in the mask option register is not used if MOR TOPT = 1 (MC6805P2 emulation). It sets the intial value of TCR TIE if MOR TOPT = 0.

Figure 5-4. MC68705R5/MC68705U5 Timer Block Diagram

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling TCR or MOR option selects one of eight outputs on the 7-bit binary divider; one output bypasses prescaling. To avoid truncation errors, the prescaler is cleared when bit 3 (b3) of the TCR is written to a logic one; however, TCR bit 3 always reads as a logic zero to ensure proper operation with read-modify-write instructions (bit set and clear for example).

At reset, the prescaler and counter are initialized to an all ones condition; the timer interupt request bit (TCR, b7) is cleared and the timer interrupt request mask (TCR, b6) is set. TCR bits b0, b1, b2, b4, and b5 are initialized by the corresponding mask option register (MOR) bits at reset. They are then software selectable after reset (if the TOPT bit (b6) in the MORE is equal to zero).

Note that the timer block diagrams in Figures 5-3 and 5-4 reflect two separate timer control configurations: a) software controlled mode via the timer control register (TCR), and b) MOR controlled mode to emulate a mask ROM version with the mask option register. In the software controlled mode, all TCR bits are read/write, except bit b3 which is write-only (always reads as a logic zero). In the MOR controlled mode, for all four devices, TCR bit b7 and b6 are read/write and bits b5, b4, b2, b1, and b0 have no effect on a write (always read as logic ones). For the MC68705R3/MC68705U3, bit b3 is write-only (reads as logic zero), and for the MC68705R5/MC68705U5, bit b3 has no effect on a write (reads as a logic one).

5.3.1 Software Controlled Mode

The TOPT (timer option) bit (b6) in the mask option register is EPROM programmed to a logic zero to select the software controlled mode, which is described first. TCR bits b5, b4, b3, b2, b1, and b0 give the program direct control of the prescaler and input select options.

The timer prescaler input frequency (f_{PIN}) can be configured for three different operating modes plus a disable mode, depending upon the value written to TCR control bits b4 and b5 (TIE and TIN).

When the TIE and TIN bits are programmed to zero the timer input is from the internal clock (phase two) and TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

When TIE = 1 and TIN = 0, the internal clock and the TIMER input pin signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The external pulse simply gates in the internal clock for the duration of the pulse. The accuracy of the count in this mode is plus or minus one count.

When TIE = 0 and TIN = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

When TIE and TIN are both programmed to a one, the timer is from the external clock. The external clock can be used to count external events as well as provide an external frequency for generating periodic interrupts.

Bits b0, b1, and b2 in the TCR are program controlled to choose the appropriate prescaler output. The prescaling divides the prescaler input frequency by 1, 2, 4, etc. in binary multiplex to 128 producing counter input frequency to the counter. The processor cannot write into or read from the

prescaler; however, the prescaler is set to all ones by a write operation to TCR, b3 (when bit 3 of the written data equals one), which allows for truncation-free counting.

5.3.2 MOR Controlled Mode

The MOR controlled mode of the timer is selected when the TOPT (timer option) bit (b6) in the MOR is programmed to a logic one to emulate the MC6805R2 mask-programmable prescaler and timer clock source. The timer circuits are the same as described above, however, the timer control register (TCR) is configured differently, as discussed below.

The logic level for the functions of bits b0, b1, b2, and b5 in the TCR are all determined at the time of EPROM programming. They are controlled by corresponding bits within the mask option register (MOR, $F38). The value programmed into MOR bits b0, b1, b2, and b5 controls the prescaler division and the timer clock selection. Bit b4 (TIE) is set to a logic one in the MOR controlled mode (when read by software, these five TCR bits always read as logic ones). As in the software programmable configuration, the TIM (b6) and TIR (b7) bits of the TCR are controlled by the counter and software as described above. Bit b3 of the TCR (in the MOR controlled mode) for the MC68705R3/MC68705U3 always reads as a logic zero and can be written to a logic one to clear the prescaler; however, for the MC68705R5/MC68705U5 bit b3 is set to a logic one and when read by software always reads as a logic one. The MOR controlled mode is designed to exactly emulate the MC6805R2 which has only TIM, TIR, and PSC in the TCR and has the prescaler options defined as manufacturing mask options.

5.3.3 Timer Control Register (TCR)

The configuration of the TCR is determined by the logic level of bit 6 (timer option, TOPT) in the mask option register (MOR). Two configurations of the TCR are shown below, one for TOPT = 1 and the other for TOPT = 0. TOPT = 1 configures the TCR to emulate the MC6805R2. When TOPT = 0, it provides software control of the TCR. When TOPT = 1, the prescaler "mask" options are user programmable via the MOR. A description of each TCR bit is provided below (also see Figures 5-3 and 5-4).

TCR with MOR TOPT = 1 (MC6805R2 Emulation)

b7	b6	b5	b4	b3	b2	b1	b0	Timer Control
TIR	TIM	1	1	PSC*	1	1	1	Register $009

*For the MC68705R3/MC68705U3 write only, reads as a zero – for the MC68705R5/MC68705U5 reads as a one and has no effect on the prescaler.

TCR with MOR TOPT = 0 (Software Programmable Timer)

b7	b6	b5	b4	b3	b2	b1	b0	Timer Control
TIR	TIM	TIN	TIE	PSC*	PS2	PS1	PS0	Register $009

*Write only, reads as a zero.

b7, TIR Timer Interrupt Request – Used to initiate the timer interrupt or signal a timer data register underflow when it is a logic one.
1 = Set when the timer data register changes to all zeros.
0 = Cleared by external reset or under program control

b6, TIM Timer Interrupt Mask — Used to inhibit the timer interrupt to the processor when it is a logic one.
1 = Set by an external reset or under program control.
0 = Cleared under program control.

b5, TIN External or Internal — Selects the input clock source to be either the external TIMER (pin 8) or the internal phase two.
1 = Selects the external clock source.
0 = Selects the internal phase two ($f_{OSC} \div 4$) clock source.

b4, TIE External Enable — Used to enable the external TIMER (pin 8) or to enable the internal clock (if TIN = 0) regardless of the external TIMER pin state (disables gated clock feature). When TOPT = 1, TIE is always a logic one.
1 = Enables external TIMER pin.
0 = Disables external TIMER pin.

TIN-TIE MODES

TIN	TIE	CLOCK
0	0	Internal Clock (phase two)
0	1	Gated (AND) of External and Internal Clocks
1	0	No Clock
1	1	External Clock

b3, PSC Prescaler Clear — When TOPT = 0, this is a write-only bit. It reads as a logic zero so the BSET and BCLR on the TCR function correctly. Writing a one into PSC generates a pulse which clears the prescaler. When TOPT = 1, operation remains the same for the MC68705R3/MC68705U3; however, for the MC68705R5/MC68705U5 this bit is always read as a logic one and has no effect on the prescaler.

b2, PS2
b1, PS1
b0, PS0 Prescaler Select — These bits are decoded to select one of eight outputs on the timer prescaler division resulting from decoding these bits.

PS2	PS1	PS0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

NOTE

When changing the PS2-PS0 bits in software, the PSC bit should be written to a one in the same write cycle to clear the prescaler. Changing the PS bits without clearing the prescaler may cause an extraneous toggle of the timer data register.

SECTION 6
SELF-CHECK

The self-check capability of the MC6805R2, MC6805U2, MC6805R3, and MC6805U3 microcomputers provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 6-1 and monitor the output of port C bit 3 for an oscillation of approximately 7 hertz. A 10-volt level (through a 10k resistor) on the timer input, pin 8, and pressing then releasing the $\overline{\text{RESET}}$ button, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, TIMER, interrupts, and I/O ports, as well as the A/D for the MC6805R2 and MC6805R3.

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM, and four-channel A/D tests. The timer routine may also be called if the timer input is the internal phase two clock.

6.1 RAM SELF-CHECK SUBROUTINE

The RAM self-check is called at location $F6F for the MC6805R2/MC6805U2 and at location $F84 for the MC6805R3/MC6805U3. If any error is detected, it returns with the Z bit cleared; otherwise the Z bit is set. The walking diagnostic pattern method is used.

The RAM test must be called with the stack pointer at $07F. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except $07F and $07E are modified.

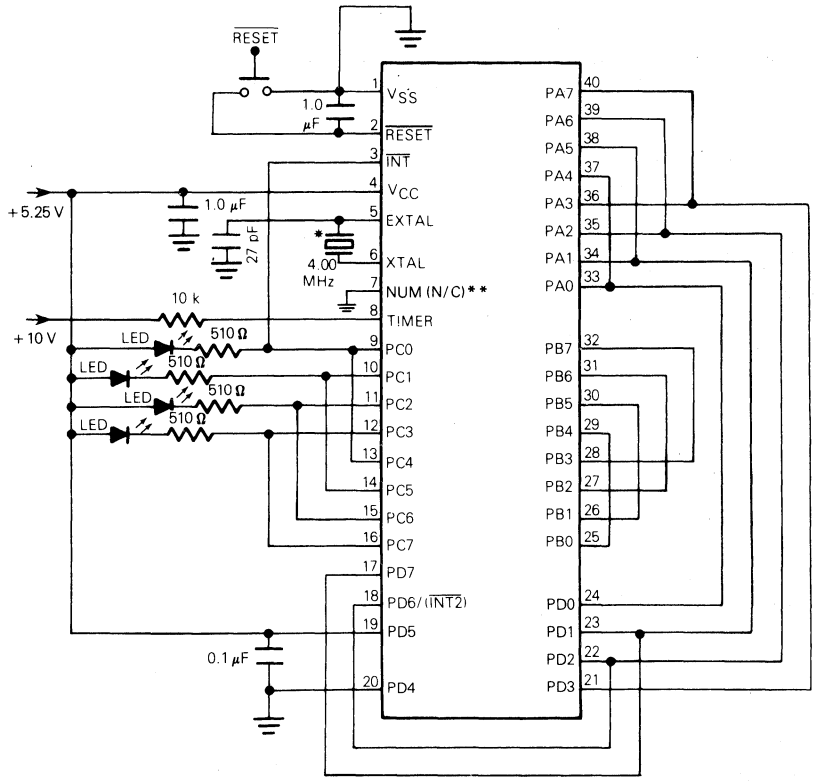
6.2 ROM CHECKSUM SUBROUTINE

The ROM self-check is called at location $F8A for the MC6805R2/MC6805U2 and at location $F95 for the MC6805R3/MC6805U3. If any error is detected, it returns with the Z bit cleared; otherwise Z = 1, X = 0 on return, and A is zero if the test passes. RAM locations $040 are overwritten.

6.3 ANALOG-TO-DIGITAL CONVERTER SELF-CHECK

The analog-to-digital self-check for the MC6805R2 is called at location $FA4 and for the MC6805R3 at $FAE. For both devices, it returns with the Z bit cleared if any error was found; otherwise the Z bit is set.

The A and X register contents are lost. The X register must be set to four before the call. On return, X = 8 and A/D channel 7 is selected. The A/D test uses the internal voltage references and confirms port connections.



* This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected.
** For the MC6805R2/MC6805U2 pin 7 is not for user application and must be connected to V_{SS}. For the MC6805R3/MC6805U3 pin 7 is not connected.

LED Meanings

PC0	PC1	PC2	PC3	Remarks [1:LED ON; 0:LED OFF]
1	0	1	0	Bad I/O
0	0	1	0	Bad Timer
1	1	0	0	Bad RAM
0	1	0	0	Bad ROM
1	0	0	0	Bad A/D
0	0	0	0	Bad Interrupts or Request Flag
All Flashing				Good Device

Anything else bad Part, Bad Port C, etc.

Figure 6-1. Self-Check Connections

6.4 TIMER SELF-CHECK SUBROUTINE

The timer self-check is called at location $FCF for the MC6805R2/MC6805U2 and at location $F6D for the MC6805R3/MC6805U3. If any error was found, it returns with the Z bit cleared; otherwise the Z bit is set.

In order to work correctly as a user subroutine, the internal phase two clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. The timer self-check routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects a timer which is not running.

SECTION 7
RESET, CLOCK, AND INTERRUPT STRUCTURE

7.1 RESET

The MCU can be reset three ways: by initial powerup, by the external reset input ($\overline{\text{RESET}}$) and by an optional internal low-voltage detect circuit (not available on the MC68705U3 or MC68705R3 EPROM versions). The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. A typical reset Schmitt trigger hysteresis curve is shown in Figure 7-1. The Schmitt trigger provides an internal reset voltage if it senses a logical zero on the $\overline{\text{RESET}}$ pin.

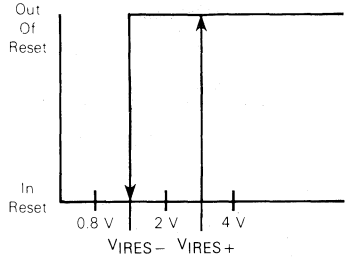


Figure 7-1. Typical Reset Schmitt Trigger Hysteresis

7.1.1 Power-On Reset (POR)

An internal reset is generated upon powerup that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the $\overline{\text{RESET}}$ input to go high. Refer to the power and reset timing diagram of Figure 7-2. Connecting a capacitor to the $\overline{\text{RESET}}$ input (as illustrated in Figure 7-3) typically provides sufficient delay. During powerup, the Schmitt trigger switches on (removes reset) when $\overline{\text{RESET}}$ rises to V_{IRES+}.

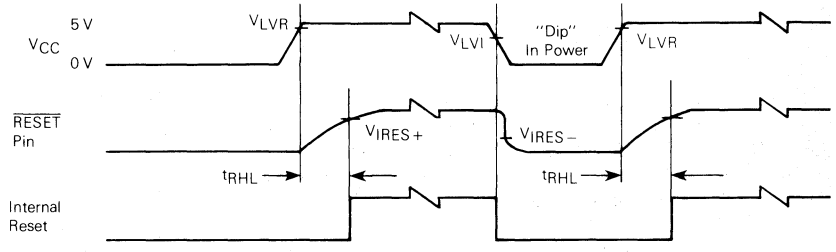


Figure 7-2. Power and Reset Timing

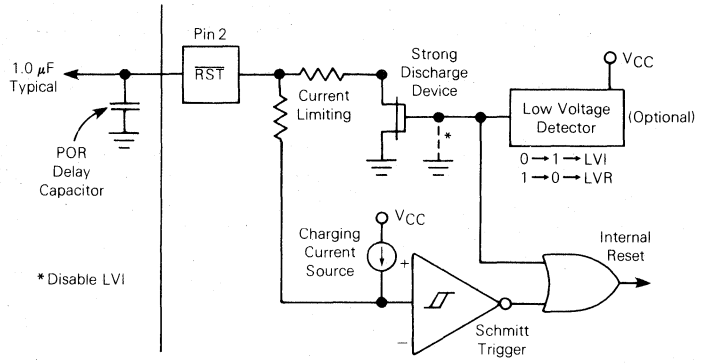


Figure 7-3. $\overline{\text{RESET}}$ Configuration

7.1.2 External Reset Input

The MCU will be reset if a logical zero is applied to the $\overline{\text{RESET}}$ input for a period longer than one machine cycle (t_{CYC}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

7.1.3 Low-Voltage Inhibit (LVI)

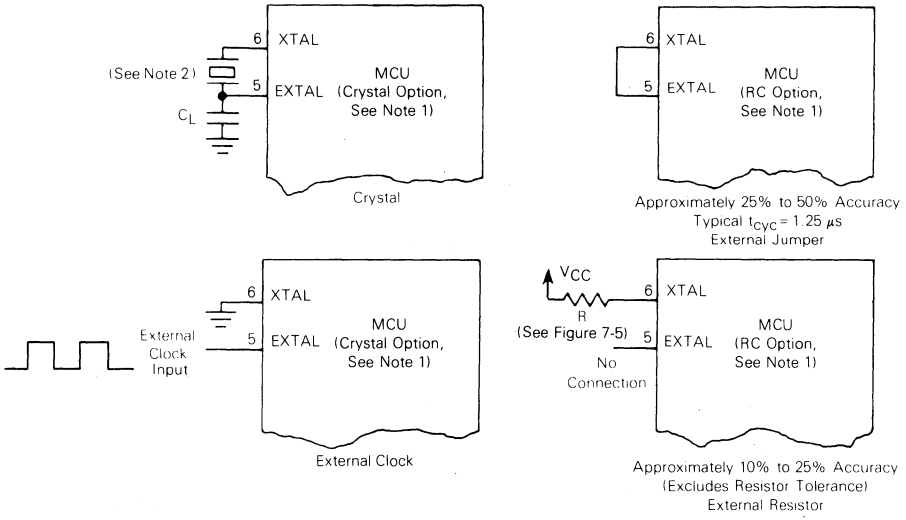
The optional low-voltage detection circuit (not available on the MC68705R3, MC68705R5, MC68705U3, and MC68705U5) causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{CYC} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{CYC}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{\text{RESET}}$ pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

7.2 INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The mask option register (EPROM) is programmed to select crystal or resistor operation. The oscillator frequency is internally divided by four to produce the internal system clocks. For MC6805R2, MC6805U2, MC6805R3, and MC6805U3 a manufacturing mask option is used to select crystal or resistor operation.

The different connection methods are shown in Figure 7-4. Crystal specifications and suggested PC board layouts are given in Figure 7-5. A resistor selection graph is given in Figure 7-6.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start up, neither the crystal characteristics nor the load capacitances should exceed recommendations.



NOTES:
1. For the MC68705R3, MC68705U3, MC68705R5, and MC68705U5 MOR b7 = 0 for the crystal option and MOR b7 = 1 for the RC option. When the TIMER input pin is in the V_{IHTP} range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V_{CC}, the clock generator option is determined by bit 7 of the Mask Option Register (CLK).
2. The recommended C_L value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

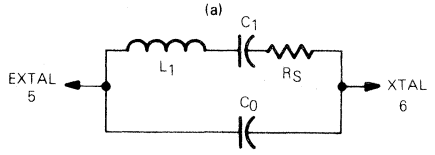
Figure 7-4. Clock Generator Options

When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition: the oscillator start-up voltage, the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{RESET}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{RESET}$ pin will charge at approximately:

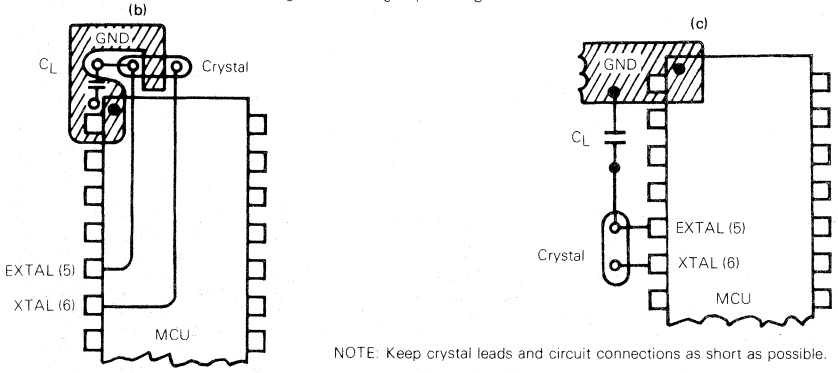
$$(V_{IRES+}) \bullet C_{ext} = I_{RES} \bullet t_{RHL}$$

Assuming the external capacitor is initially discharged.



AT – Cut Parallel Resonance Crystal
$C_0 = 7$ pF Max
Freq. = 4.0 MHz @ $C_L = 24$ pF
$R_S = 50$ ohms Max.

Piezoelectric ceramic resonators which have the equivalent specifications may be used instead of crystal oscillators. Follow ceramic resonator manufacturer's suggestions for C_0, C_1, and R_S values.



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

Figure 7-5. Crystal Motional-Arm Parameters and Suggested PC Board Layout

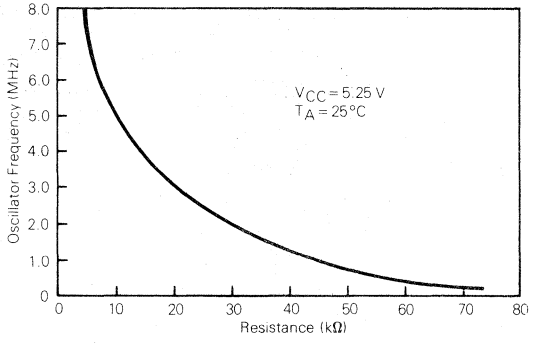


Figure 7-6. Typical Frequency Selection for Resistor (RC Oscillator Option)

7.3 INTERRUPTS

The microcomputers can be interrupted four different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, the external port D bit 6 ($\overline{INT2}$) input pin, or the software interrupt instruction (SWI). When any interrupt occurs: the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU register, setting the I bit, and vector fetching require a total of 11 t_{CYC} periods for completion. A flowchart of the interrupt sequence is shown in Figure 7-7. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the

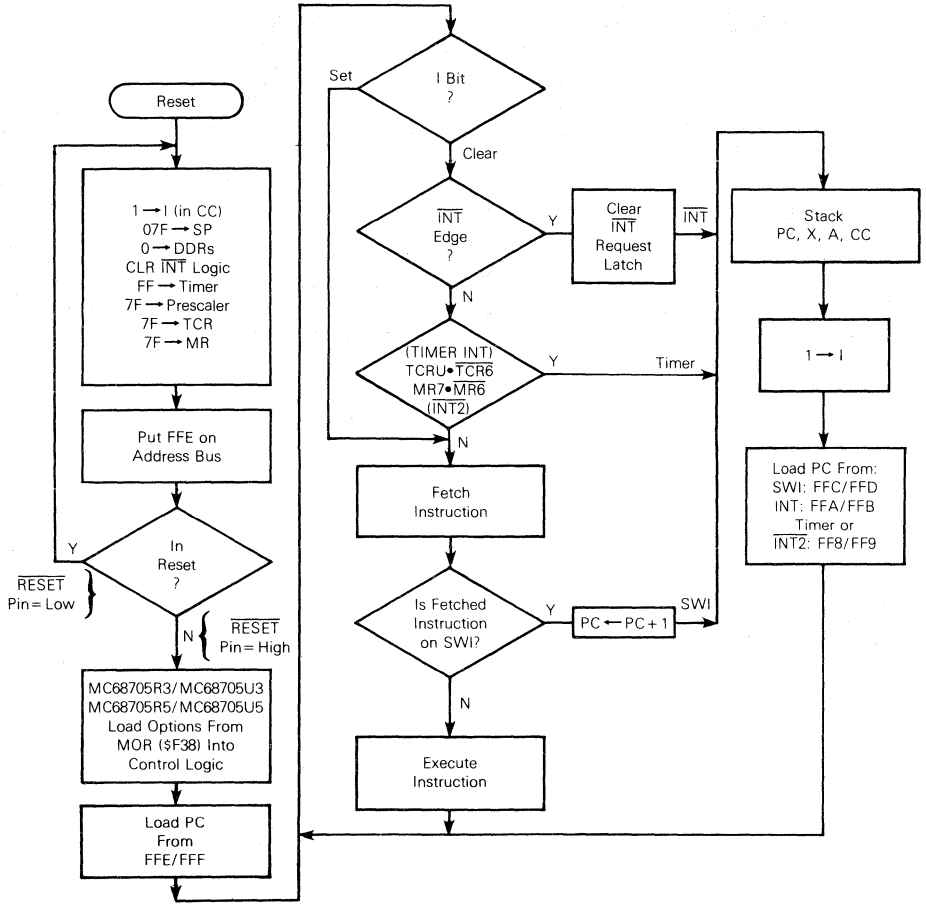


Figure 7-7. Reset and Interrupt Processing Flowchart

program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

NOTE

The timer and $\overline{\text{INT2}}$ interrupts share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR b7 and MR b7). Both TCR b7 and MR b7 can only be written to zero by software.

The external interrupt, $\overline{\text{INT}}$ and $\overline{\text{INT2}}$, are synchronized and then latched on the falling edge of the input signal. The $\overline{\text{INT2}}$ interrupt has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the miscellaneous register (MR). The $\overline{\text{INT2}}$ interrupt is inhibited when the mask bit is set. The $\overline{\text{INT2}}$ is always read as a digital input on port D. The $\overline{\text{INT2}}$ and timer interrupt request bits, if set, cause the MCU to process an interrupt when the condition code I bit is clear.

A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock. See Figure 7-8.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as breakpoints for debugging or as system calls.

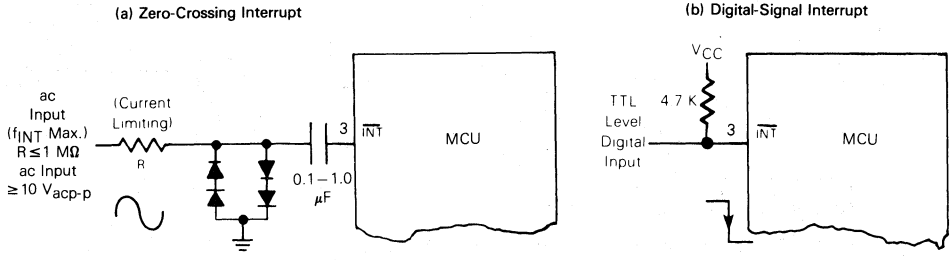
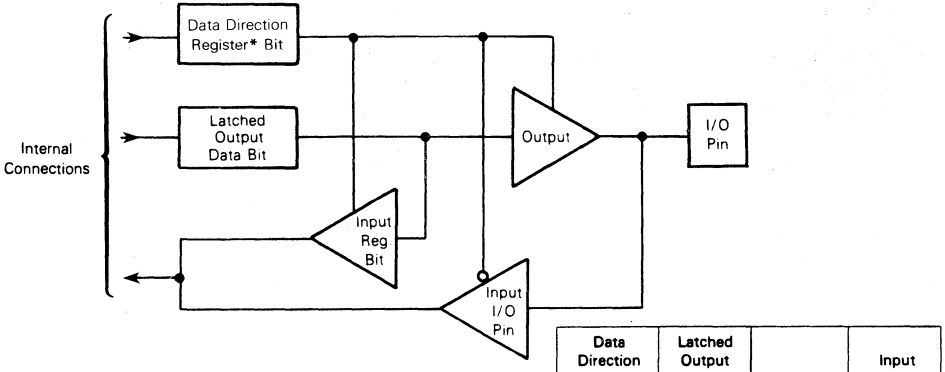


Figure 7-8. Typical Interrupt Circuits

SECTION 8
INPUT/OUTPUT CIRCUITRY AND ANALOG-TO-DIGITAL CONVERTER

8.1 INPUT/OUTPUT CIRCUITRY

There are 32 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on ports 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, placing the ports in the input mode. The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. A read operation on a port programmed as an output will read the contents of the output latch regardless of the logic levels at the output pin, due to output loading. Refer to Figure 8-1.



Data Direction Register Bit	Latched Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

*DDR is a write-only register and reads as all ones.
Port B and C are three-state ports. Port A has optional internal pullup devices to provide CMOS data drive capability. See **SECTION 11 ELECTRICAL CHARACTERISTICS.

Figure 8-1. Typical Port I/O Circuitry

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs (mask option on the MC6805R2, MC6805U2, MC6805R3, and MC6805U3) while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only; thus, there is no corresponding DDR. When programmed as outputs, port b is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin.

Port D provides the reference voltage, $\overline{INT2}$, and multiplexed analog inputs for the MC6805R2, MC6805R3, MC68705R3, and MC68705R5. All of these lines are shared with the port D digital inputs. Port D may always be used as digital inputs and may also be used as analog inputs providing V_{RH} and V_{RL} are connected to the appropriate reference voltages. The V_{RL} and V_{RH} lines (PD4 and PD5) are internally connected to the A/D resistor. Analog inputs may be prescaled to attain the V_{RL} and V_{RH} recommended input voltage range.

The address maps in Section 3 give the addresses of data registers and data direction registers. The register configuration is provided in Figure 7-6. Figure 8-2 provides some examples of port connections.

3

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modfiy-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

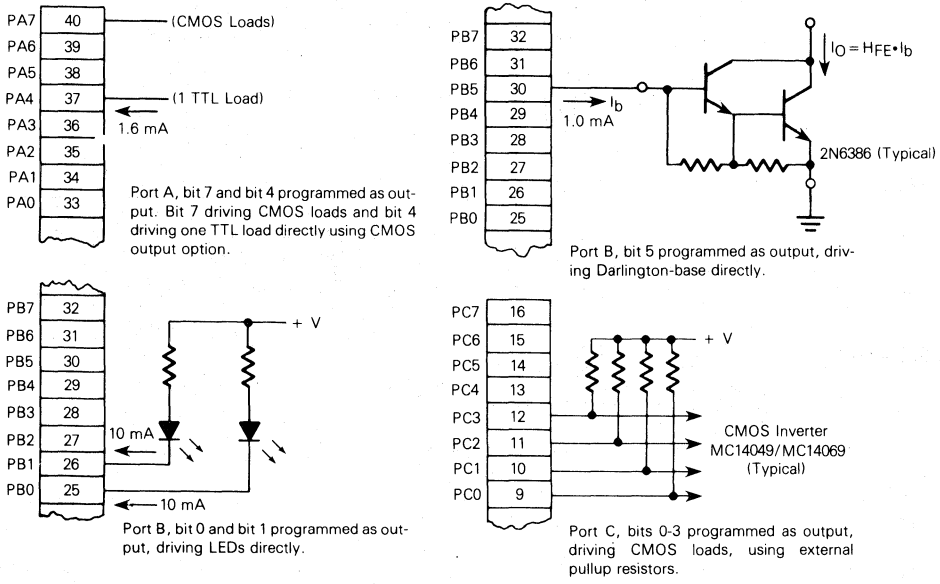


Figure 8-2a. Typical Port Connections—Output Modes

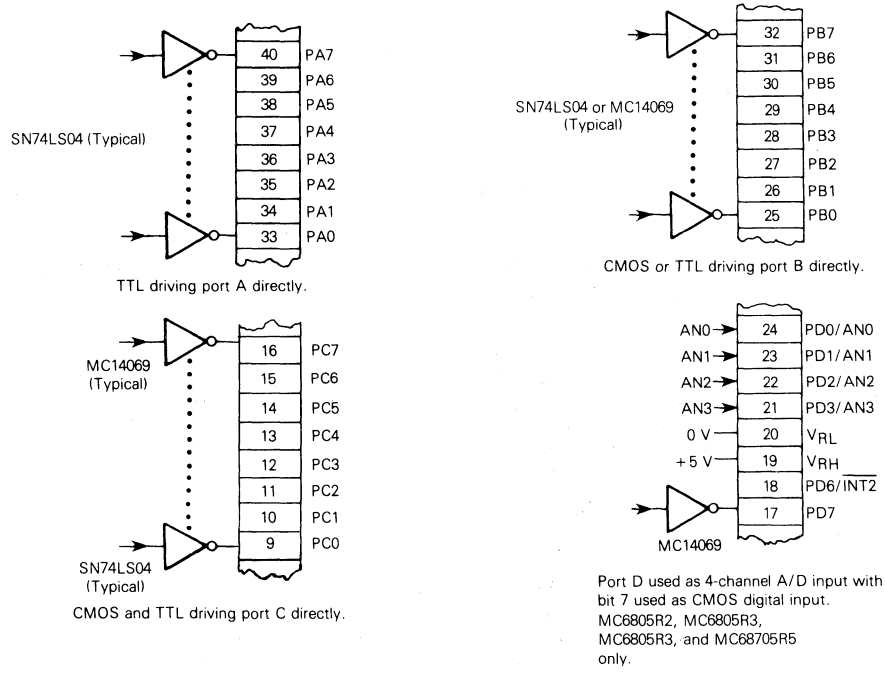


Figure 8-2b. Typical Port Connections—Input Modes

The latched output data bit (see Figure 8-1) must always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data register 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 (zero) and corresponds to the latched output data when the DDR is an output (one).

8.2 ANALOG-TO-DIGITAL CONVERTER

The MC6805R2, MC6805R3, MC68705R3 and MC68705R5 microcomputers have an 8-bit analog-to-digital (A/D) converter implemented on the chip using a successive approximation technique, as shown in Figure 8-3. Up to four external analog inputs, via port D, are connected to the A/D through a multiplexer. Four internal analog channels may be selected for calibration purposes (V_{RH}-V_{RL}, V_{RH}-$V_{RL}/2$, V_{RH}-$V_{RL}/4$, and V_{RL}). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2; see Table 8-1. This register is cleared during any reset condition. Refer to Figure 7-6 for the register configuration.

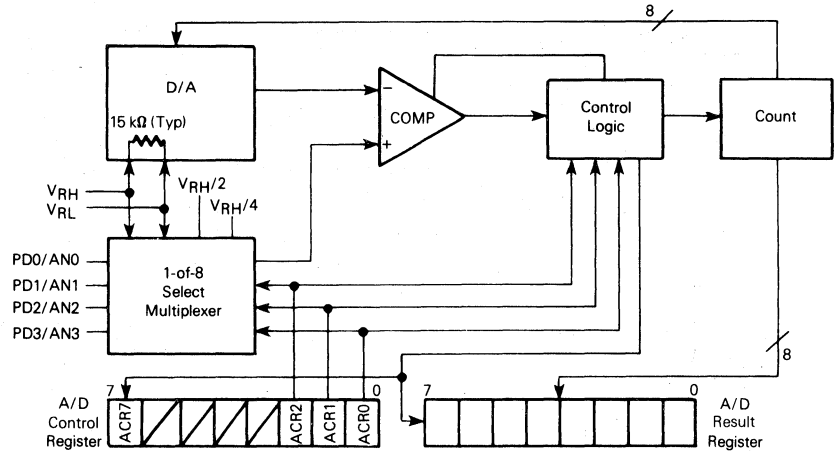


Figure 8-3. A/D Block Diagram

Table 8-1. A/D Input MUX Selection

A/D Control Register			Input Selected	A/D Output (Hex)		
ACR2	ACR1	ACR0		Min	Typ	Max
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}*	FE	FF	FF
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

*Internal (Calibration) Levels

Whenever the ACR is written, the conversion in progress is aborted, the conversion complete flag (ACR bit 7) is cleared, and the selected input is sampled for five machine cycles and held internally. During these five cycles, the analog input will appear approximately like a 25 picofarad (maximum) capacitor (plus approximately 10 pF for packaging) charging through a 2.6 kilohm resistor (typical). Refer to Figure 8-4.

The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When the conversion is complete, the digitized sample of digital value is placed in the A/D result register (ARR), the conversion complete flag is set, the selected input is sampled again, and a new conversion is started.

The A/D is ratiometric. Two reference voltages (V_{RH} and V_{RL}) are supplied to the converter via port D pins. An input voltage equal to V_{RH} converts to $FF (full scale) and an input voltage equal to V_{RL} converts to $00. An input voltage greater than V_{RH} converts to $FF and no overflow indication is provided. Similarly, an input voltage less than V_{RL}, but greater than V_{SS} converts to $00. Maximum and minimum ratings must not be exceeded. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}. To maintain the full

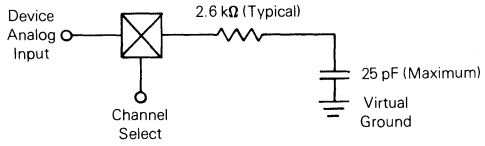


Figure 8-4. Effective Analog Input Inpedance (During Sampling Only)

accuracy on the A/D, V_{RH} should be equal to or less than V_{DD}, V_{RL} should be equal to or greater than V_{SS} but less than the maximum specification and (V_{RH}-V_{RL}) should be equal to or greater than 4 volts.

The A/D has a built-in ½ LSB offset intended to reduce the magnitude of the quantizing error to ± ½ LSB, rather than +0, −1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at ½ LSB above V_{RL}. Similarly, the transition from $FE to $FF occurs 1½ LSB below V_{RH}, ideally. Refer to Figures 8-5 and 8-6.

On release of reset, the A/D control register (ACR) is cleared therefore after reset, channel zero will be selected and the conversion complete flag will be clear.

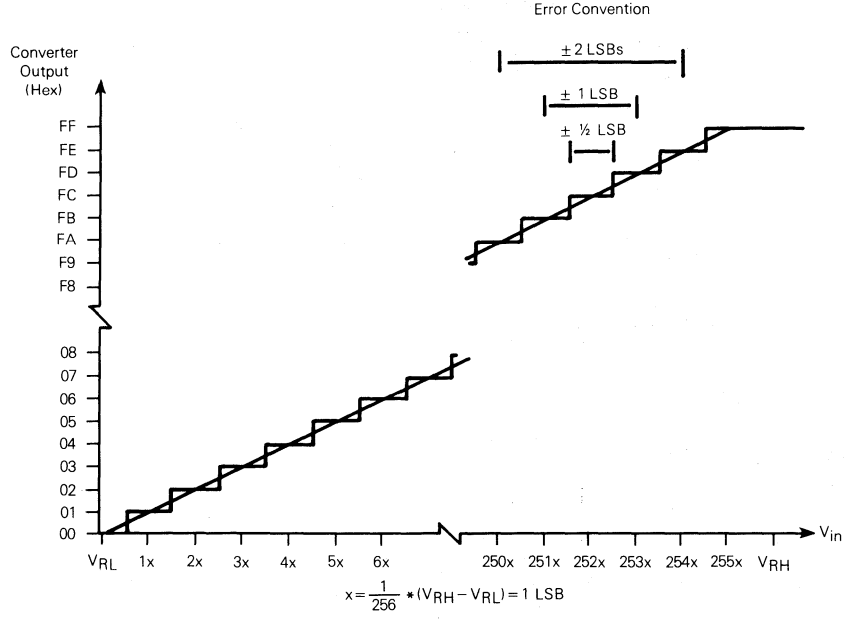


Figure 8-5. Ideal Converter Transfer Characteristic

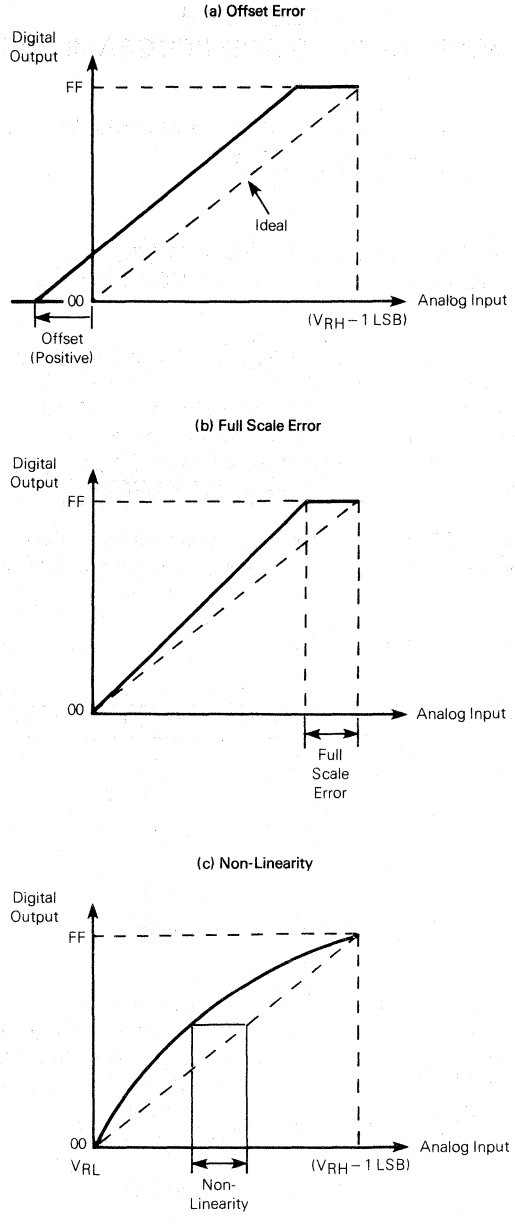


Figure 8-6. Types of Conversion Errors

SECTION 9
MASK OPTIONS AND PROGRAMMING

The information in this section pertains only to the MC68705R3, MC68705U3, MC68705R5, and MC68705U5 EPROM versions of the M6805 Family.

9.1 MASK OPTIONS

The MC68705R3, MC68705U3, MC68705R5, and MC68705U5 mask option registers are implemented in EPROM. Like all other EPROM bytes, the MOR contains all zeros prior to programming (if erased).

When used to emulate the MC6805R2 and MC6805U2, five of the eight MOR bits are used in conjunction with the prescaler. Of the remaining, the b7 bit is used to select the type of clock oscillator, b3 is the secure/non-secure mode option for the MC68705R5/MC68705U5 only (b3 is not used by the MC68705R3/MC68705U3), and b4 is not used. Bits b0, b1, and b2 determine the division of the timer prescaler. Bit b6 determines the timer option selection. The value of the TOPT bit (b6) is programmed to configure the TCR (a logic one for MC6805R2/MC6805U2 emulation).

If the MOR timer option (TOPT) bit is a zero, the MC6805R3 and MC6805U3 are emulated. Here, bits b5, b4, b2, b1, and b0 set the initial value of their respective TCR bits during reset. After initialization the TCR is software controllable.

A description of the MOR bits is as follows:

b7	b6	b5	b4	b3	b2	b1	b0	Mask Option
CLK	TOPT	CLS		SNM*	P2	P1	P0	Register $F38

*MC68705R5 and MC68705U5 only.

b7, CLK Clock Oscillator Type
1 = RC
0 = Crystal

NOTE

V_{IHTP} on the TIMER (pin 8) forces the crystal mode.

b6, TOPT Timer Option
1 = MC6805R2/MC6805U2 type timer/prescaler. All bits, except b6 and b7, of the timer control register (TCR) are invisible to the user. Bits b5, b2, b1, and b0 of the mask option register determine the equivalent MC6805R2/MC6805U2 mask options.
0 = MC6805R3/MC6805U3 type timer/prescaler. All TCR bits are implemented as a software programmable timer. The state of bits b5, b4, b2, b1, and b0 set the initial values of their respective TCR bits (TCR is then software controlled after initialization).

b5, CLS Timer/Prescaler Clock Source
1 = External TIMER pin.
0 = Intenal phase two.

b4 Not used if MOR TOPT = 1 (MC6805R2/MC6805U2 emulation). Sets initial value of TCR TIE if MOR TOPT = 0 (MC6805R3/MC6805U3 emulation).

b3, SNM* When this bit is set, i.e., programmed to a one, it is not possible to access the EPROM contents of the MC68705R5 and MC68705U5 externally. For more information refer to **9.4 PROGRAMMING FIRMWARE.**

Note, For MC68705R3 and MC68705U3 operation, b3 is not used.

b2, P2
b1, P1
b0, P0 Prescaler Option — the logic levels of these bits, when decoded, select one of eight outputs on the timer prescaler. The table below shows the division resulting from decoding combinations of these three bits.

P2	P1	P0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Two examples for programming the mask option register are discussed below.

Example 1

To emulate an MC6805R2/MC6805U2 to verify your program with an RC oscillator and an event count input for the timer with no prescaling, the mask option register would be set to 11110000. To write the mask option register, it is simply programmed as any other EPROM byte.

Example 2

Suppose you wish to use programmable prescaler functions (MC6805R3/MC6805U3 emulation), and you wish the initial condition of the prescaler to be divided by 64, with the input disabled and an internal clock source. If the clock oscillator was to be in the crystal mode, the mask option register would be set to 00000110.

9.2 ON-CHIP PROGRAMMING HARDWARE

The programming control register (PCR) at location $00B is an 8-bit register which utilizes the three least significant bits (the five most significant bits are set to logic ones). This register provides the

*MC68705R5 and MC68705U5 only.

necessary control bits to allow programming the EPROM. The bootstrap program manipulates the PCR when programming so that users need not be concerned with the PCR in most applications. A description of each bit follows.

b7	b6	b5	b4	b3	b2	b1	b0	Programming
1	1	1	1	1	$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Control Register $00B

b0, $\overline{PLE}$ Programming Latch Enable — When cleared, this bit allows the address and data to be latched into the EPROM. When this bit is set, data can be read from the EPROM.

1 = (set) read EPROM

0 = (clear) latch address and data into EPROM (read disable)

PLE is set during a reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{VPON}$ is a logic one.

b1, $\overline{PGE}$ Program Enable — When cleared, $\overline{PGE}$ enables programming of the EPROM. $\overline{PGE}$ can only be cleared if $\overline{PLE}$ is cleared. $\overline{PGE}$ must be set when changing the address and data; i.e., setting up the byte to be programmed.

1 = (set) inhibit EPROM programming

0 = (clear) enable EPROM programming (if $\overline{PLE}$ is low)

$\overline{PGE}$ is set during a reset; however, it has no effect on EPROM circuits if $\overline{VPON}$ is a logic one.

b2, $\overline{VPON}$ (V_{PP} ON) — $\overline{VPON}$ is a read-only bit and when at a logic zero it indicates that a "high voltage" is present at the V_{PP} pin.

1 = no "high voltage" on V_{PP} pin

0 = "high voltage" on V_{PP} pin

$\overline{VPON}$ being one "disconnects" $\overline{PGE}$ and $\overline{PLE}$ from the rest of the chip, preventing accidental clearing of these bits from affecting the normal operating mode.

NOTE

$\overline{VPON}$ being zero does not indicate that the V_{PP} level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

The programming control register functions are shown below.

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	0	1	Invalid state; $\overline{PGE}=0$ iff $\overline{PLE}=0$
1	0	1	Invalid state; $\overline{PGE}=0$ iff $\overline{PLE}=0$
0	1	1	"High voltage" on V_{PP}
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled from system (operating mode)

9.3 ERASING THE EPROM

The EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 angstroms. The recommended integrated dose (UV intensity × exposure time) is 25 Ws/cm^2. The lamps should be used without shortwave filters and the EPROM should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the EPROM to the zero state. Data is the entered by programming ones into the desired bit locations.

CAUTION

Be sure that the EPROM window is shielded from light except when erasing. This protects both the EPROM and light-sensitive nodes.

9.4 PROGRAMMING FIRMWARE

The MC68705R3, MC68705U3, MC68705R5, and MC68705U5 have 120 bytes of mask ROM containing a bootstrap program which can be used to program the EPROM. The vector at addresses $FF6 and $FF7 is used to start executing the program. This vector is fetched when V_{IHTP} is applied to pin 8 (TIMER pin) and the $\overline{RESET}$ pin is allowed to rise above V_{IRES+}. Figure 9-1 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM. It is possible to program the EPROM via the bootstrap software and validate its contents in the secure mode. The only way to go from the secure mode to the non-secure mode is by erasing the entire EPROM.

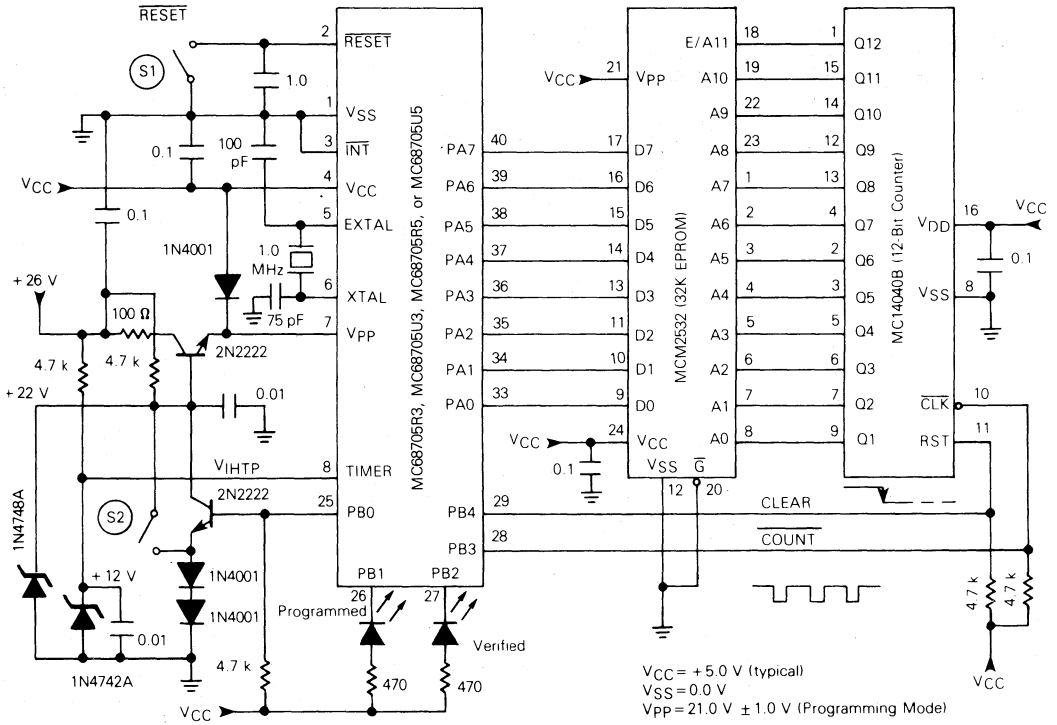
9.5 PROGRAMMING STEPS

The MCM2532 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705R3, MC68705U3, MC68705R5, or MC68705U5. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705R3, MC68705U3, MC68705R5, or MC68705U5 and the MCM2532 are to be inserted and removed from the circuit, they should be mounted in sockets. In addition, the precaution below must be observed (refer to Figure 9-1).

CAUTION

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705R3, MC68705U3, MC68705R5, or MC68705U5 and MCM2532 into their respective sockets. This ensures that $\overline{RESET}$ is held low while inserting the devices.

When ready to program the MC68705R3, MC68705U3, MC68705R5, or MC68705U5 it is only necessary to provide V_{CC} and +26 volts, open switch S2 (to apply V_{PP} and V_{IHTP}), and then open S1 (to remove reset). Once the voltages are applied and both S2 and S1 are open, the CLEAR output control line (PB4) goes high and then low, then the 12-bit counter (MC14040B) is clocked by the PB3 output (COUNT). The counter selects the MCM2532 EPROM byte which is to load the equivalent EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, COUNT clocks the counter to the next EPROM location. This continues until the EPROM is completely programmed at which time the programmed indicator LED is lit. The counter is cleared and the loop is repeated to verify the programmed data. The verified indicator LED lights if the programming is correct.



Summary of Programming Steps:

1. When plugging in the MC68705R3, MC68705U3, MC68705R5, or MC68705U5 or the MCM2532 be sure that S1 and S2 are closed and that V_{CC} and + 26 V are not applied.
2. To initiate programming, be sure S1 is closed, S2 is closed and V_{CC} and + 25 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705R3, MC68705U3, MC68705R5, or MC68705U5, first close S2 and then close S1. Disconnect V_{CC} and + 26 V then remove the MC68705R3, MC68705U3, MC68705R5, or MC68705U5.

Figure 9-1. Programming Connections Schematic Diagram

Once the EPROM has been programmed and verified, close switch S2 (to remove V_{PP} and V_{IHTP}) and close switch S1 (to reset). Disconnect +26 volts and V_{CC}, then remove the EPROM from its socket.

9.6 EMULATION

The MC68705R3 and MC68705R5 emulate the MC6805R2 and MC6805R3 while the MC68705U3 and MC68705U5 emulate the MC685U2 and MC6805U3 "exactly." MC6805R2, MC6805U2, MC6805R3, and MC6805U3 mask features are implemented in the mask option register (MOR) EPROM byte. There are a few minor exceptions to the exactness of emulation which are listed below.

1. The MC6805R2/MC6805U2 and MC6805R3/MC6805U3 "future ROM" areas are implemented in the MC68705R3/MC68705U3 and MC68705R5/MC68705U5 and these 1728 bytes must be left unprogrammed to accurately simulate the MC6805R2/MC6805U2 and MC6805R3/MC6805U3. (The MC6805R2/MC6805U2 and MC6805R3/MC6805U3 read all zeros from this area.)

2. The reserved ROM areas have different data stored in them and this data is subject to change without notice. The MC6805R2, MC6805U2, MC6805R3, and MC6805U3 use the reserved ROM for the self-check feature while the MC68705R3, MC68705U3, MC68705R5, and MC68705U5 use this area for the bootstrap program.

3. The MC6805R2/MC6805U2 and MC6805R3/MC6805U3 read all ones in their 48 byte "future RAM" area. This RAM is not implemented in the MC6805R2/MC6805U2 and MC6805R3/MC6805U3 mask ROM version, but is implemented in the EPROM version (MC68705R3/MC68705U3 and MC68705R5/MC68705U5).

4. The V_{PP} line (pin 7) in the MC68705R3/MC68705U3 and MC68705R5/MC68705U5 EPROM versions is tied to V_{CC} for normal operation. In the MC6805R2/MC6805U2 and MC6805R3/MC6805U3 mask ROM versions, pin 7 is grounded in normal operation.

5. The LVI feature is not available in the EPROM version.

6. The MC68705R3/MC68705U3 and MC68705R5/MC68705U5 EPROM versions do not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805R2/MC6805U2 and MC6805R3/MC6805U3 mask ROM versions, except for pin 7 as previously noted.

The operation of all other circuitry has been exactly duplicated or designed to function exactly the same way in all devices including interrupts, timer, data ports, and data direction registers (DDRs).

SECTION 10
SOFTWARE

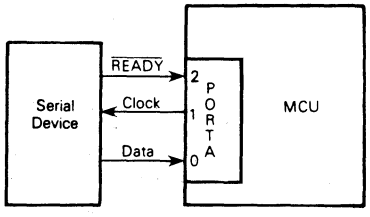
10.1 BIT MANIPULATION

The microcomputers have the ability to set or clear any single RAM or I/O bit (except the data direction registers) with a single instruction (BSET, BCLR) (see Caution below). Any bit in page zero can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit equals the value of the bit references by BRSET or BRCLR. The capability of working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

CAUTION

The corresponding data direction registers for ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a data direction regiser bit (all "unaffected" bits would be set). It is recommended that all data direction register bits in a port be written using a single-store instruction.

The coding examples shown in Figure 10-1 illustrate the usefulness of the bit manipulation and test instruction. Assume that the microcomputer 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, least significant bit first out of the device. The microcomputer waits until the data is ready, clocks the external device, picks up the data in the carry flag, clears the clock line, and finally accumulates the data bit in a random-access memory location.




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

Figure 10-1. Bit Manipulation Examples

10.2 ADDRESSING MODES

The microcomputers have ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and illustrations, refer to the *M6805 HMOS/M146805 CMOS Family Microcomputer/Microprocessor User's Manual.*

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

10.2.1 Immediate

In the immediate addressing mode, the operand is contained in the byte immediately 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).

10.2.2 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 lowest 256 bytes in memory with a single two-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

10.2.3 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

10.2.4 Relative

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not worry about calculating the correct offset if he uses the Motorola assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

10.2.5 Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only on byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

10.2.6 Indexed, 8-Bit Offset

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is

useful in selecting the kth element in an n element table. With this two-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

10.2.7 Index, 16-Bit Offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended, the Motorola determines the shortest form of indexed addressing.

10.2.8 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 addressing of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

CAUTION

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined since BSET and BCLR are read-modify-write in function, they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

10.2.9 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 and condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the condition code registers. See Caution under paragraph 10.2.8.

10.2.10 Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other agruments, are included in this mode. These instructions are one byte long.

10.3 INSTRUCTION SET

The MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 usable opcodes. They can be divided into five 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.

10.3.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from meory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 10-1.

10.3.2 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 (see Caution under paragraph 10.2.8). The test for negative or zero (TST) instructions is included in the read-modify-write instructions, though it does not perform the write. Refer to Table 10-2.

10.3.3 Branch Instructions

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 10-3.

10.3.4 Bit Manipulation Instructions

These instructions are used on any bit in the first 256 bytes of memory; see Caution under paragraph 10.2.8. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 10-4.

10.3.5 Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 10-5.

10.3.6 Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 10-6.

10.3.7 Opcode Map Summary

Table 10-7 is an opcode made for the instructions used on the MCU.

Table 10-1. Register Memory Instructions

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—	—	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—	—	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 10-2. Read-Modify-Write Instructions

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 10-3. Branch Instructions

		Relative Addressing Mode		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 10-4. Bit Manipulation Instructions

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n(n=0. . .7)	–	–	–	2•n	3	10
Branch IFF Bit n is Clear	BRCLR n(n=0. . .7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n(n=0. . .7)	10+2•n	2	7	–	–	–
Clear Bit n	BCLR n(n=0. . .7)	11+2•n	2	7	–	–	–

Table 10-5. Control Instructions

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

3

Table 10-6. Instruction Set (Sheet 1 of 2)

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			^	•	^	^	^
ADD		X	X	X		X	X	X			^	•	^	^	^
AND		X	X	X		X	X	X			•	•	^	^	•
ASL	X		X			X	X				•	•	^	^	^
ASR	X		X			X	X				•	•	^	^	^
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	^	^	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected

Table 10-6. Instruction Set (Sheet 2 of 2)

	Addressing Modes										Condition Code				
Mnem	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	∧
BRSET										X	●	●	●	●	∧
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	∧	∧	∧
COM	X		X			X	X				●	●	∧	∧	1
CPX		X	X	X		X	X	X			●	●	∧	∧	∧
DEC	X		X			X	X				●	●	∧	∧	●
EOR		X	X	X		X	X	X			●	●	∧	∧	●
INC	X		X			X	X				●	●	∧	∧	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	∧	∧	●
LDX		X	X	X		X	X	X			●	●	∧	∧	●
LSL	X		X			X	X				●	●	∧	∧	∧
LSR	X		X			X	X				●	●	0	∧	∧
NEQ			X			X	X				●	●	∧	∧	∧
NOP	X										●	●	●	●	●
ORA	X	X	X	X		X	X	X			●	●	∧	∧	●
ROL	X		X			X	X				●	●	∧	∧	∧
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	∧	∧	∧
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	∧	∧	●
STX			X	X		X	X	X			●	●	∧	∧	●
SUB		X	X	X		X	X	X			●	●	∧	∧	∧
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	∧	∧	●
TXA	X										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10-7. M6805 HMOS Family Instruction Set Opcode Map

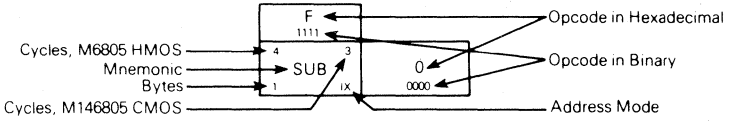
	Bit Manipulation		Branch	Read/Modify/Write				
	BTB	BSC	REL	DIR	A	X	IX1	IX
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	10 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEG 1 A	4 3 NEG 1 X	7 6 NEG 2 IX1	6 5 NEG 1 IX
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL					
2 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL					
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COM 1 A	4 3 COM 1 X	7 6 COM 2 IX1	6 5 COM 1 IX
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSR 1 A	4 3 LSR 1 X	7 6 LSR 2 IX1	6 5 LSR 1 IX
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL					
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 ROR 1 A	4 3 ROR 1 X	7 6 ROR 2 IX1	6 5 ROR 1 IX
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASR 1 A	4 3 ASR 1 X	7 6 ASR 2 IX1	6 5 ASR 1 IX
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSL 1 A	4 3 LSL 1 X	7 6 LSL 2 IX1	6 5 LSL 1 IX
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROL 1 A	4 3 ROL 1 X	7 6 ROL 2 IX1	6 5 ROL 1 IX
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DEC 1 A	4 3 DEC 1 X	7 6 DEC 2 IX1	6 5 DEC 1 IX
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL					
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INC 1 A	4 3 INC 1 X	7 6 INC 2 IX1	6 5 INC 1 IX
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TST 1 A	4 3 TST 1 X	7 5 TST 2 IX1	6 4 TST 1 IX
E 1110	10 7 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL					
F 1111	10 7 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLR 1 A	4 3 CLR 1 X	7 6 CLR 2 IX1	6 5 CLR 1 IX

Abbreviations for Address Modes

INH	Inherent	EXT	Extended	IX	Indexed (No Offset)
A	Accumulator	REL	Relative	IX1	Indexed, 1 Byte (8-Bit) Offset
X	Index Register	BSC	Bit Set/Clear	IX2	Indexed, 2 Byte (16-Bit) Offset
IMM	Immediate	BTB	Bit Test and Branch	*	M146805 CMOS Family Only
DIR	Direct				

Control		Register/Memory						
INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
9 9 RTI 1 INH		2 2 SUB 2 IMM	4 3 SUB 2 DIR	5 4 SUB 3 EXT	6 5 SUB 3 IX2	5 4 SUB 2 IX1	4 3 SUB 1 IX	0 0000
6 6 RTS 1 INH		2 2 CMP 2 IMM	4 3 CMP 2 DIR	5 4 CMP 3 EXT	6 5 CMP 3 IX2	5 4 CMP 2 IX1	4 3 CMP 1 IX	1 0001
		2 2 SBC 2 IMM	4 3 SBC 2 DIR	5 4 SBC 3 EXT	6 5 SBC 3 IX2	5 4 SBC 2 IX1	4 3 SBC 1 IX	2 0010
11 10 SWI 1 INH		2 2 CPX 2 IMM	4 3 CPX 2 DIR	5 4 CPX 3 EXT	6 5 CPX 3 IX2	5 4 CPX 2 IX1	4 3 CPX 1 IX	3 0011
		2 2 AND 2 IMM	4 3 AND 2 DIR	5 4 AND 3 EXT	6 5 AND 3 IX2	5 4 AND 2 IX1	4 3 AND 1 IX	4 0100
		2 2 BIT 2 IMM	4 3 BIT 2 DIR	5 4 BIT 3 EXT	6 5 BIT 3 IX2	5 4 BIT 2 IX1	4 3 BIT 1 IX	5 0101
		2 2 LDA 2 IMM	4 3 LDA 2 Dir	5 4 LDA 3 EXT	6 5 LDA 3 IX2	5 4 LDA 2 IX1	4 3 LDA 1 IX	6 0110
	2 2 TAX 1 INH		5 4 STA 2 DIR	6 5 STA 3 EXT	7 6 STA 3 IX2	6 5 STA 2 IX1	5 4 STA 1 IX	7 0111
	2 2 CLC 1 INH	2 2 EOR 2 IMM	4 3 EOR 2 DIR	5 4 EOR 3 EXT	6 5 EOR 3 IX2	5 4 EOR 2 IX1	4 3 EOR 1 IX	8 1000
	2 2 SEC 1 INH	2 2 ADC 2 IMM	4 3 ADC 2 DIR	5 4 ADC 3 EXT	6 5 ADC 3 IX2	5 4 ADC 2 IX1	4 3 ADC 1 IX	9 1001
	2 2 CLI 1 INH	2 2 ORA 2 IMM	4 3 ORA 2 DIR	5 4 ORA 3 EXT	6 5 ORA 3 IX2	5 4 ORA 2 IX1	4 3 ORA 1 IX	A 1010
	2 2 SEI 1 INH	2 2 ADD 2 IMM	4 3 ADD 2 DIR	5 4 ADD 3 EXT	6 5 ADD 3 IX2	5 4 ADD 2 IX1	4 3 ADD 1 IX	B 1011
	2 2 RSP 1 INH		3 2 JMP 2 DIR	4 3 JMP 3 EXT	5 4 JMP 3 IX2	4 3 JMP 2 IX1	3 2 JMP 1 IX	C 1100
	2 2 NOP 1 INH	8 6 BSR 2 REL	7 5 JSR 2 DIR	8 6 JSR 3 EXT	9 7 JSR 3 IX2	8 6 JSR 2 IX1	7 5 JSR 1 IX	D 1101
* 2 STOP 1 INH		2 2 LDX 2 IMM	4 3 LDX 2 DIR	5 4 LDX 3 EXT	6 5 LDX 3 IX2	5 4 LDX 2 IX1	4 3 LDX 1 IX	E 1110
* 2 WAIT 1 INH	2 2 TXA 1 INH		5 4 STX 2 DIR	6 5 STX 3 EXT	7 6 STX 3 IX2	6 5 STX 2 IX1	5 4 STX 1 IX	F 1111

LEGEND



SECTION 11
ELECTRICAL CHARACTERISTICS

11.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage			
MC6805R2, MC6805R3, MC6805U2, and MC6805U3 (Except TIMER in Self-Check Mode and Open-Drain Inputs)	V_{in}	−0.3 to +7.0	V
Self-Check Mode (TIMER Pin Only)	V_{in}	−0.3 to +15.0	V
MC68705R3/MC68705U3			
EPROM Programming Voltage (V_{PP} Pin)	V_{PP}	−0.3 to +22.0	V
TIMER Pin — Normal Mode	V_{in}	−0.3 to +7.0	V
TIMER Pin — Bootstrap Programming Mode	V_{in}	−0.3 to +15.0	V
All Others	V_{in}	−0.3 to +7.0	V
Operating Temperature Range		T_L to T_H	
MC6805R2, MC6805U2, MC6805R3, MC6805U3, MC68705R3, MC68705U3, MC68705R5, MC68705U5	T_A	0 to +70	°C
MC6805R2C, MC6805U2C, MC6805R3C, MC6805U3C, MC68705R3C, MC68705U3C, MC68705R5C, MC68705U5C		−40 to 85	
MC6805R2V, MC6805U2V, MC6805R3V, MC6805U3V		−40 to 105	
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature			
Plastic		150	
Ceramic	T_J	175	°C/W
Cerdip		175	

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

11.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic (P Suffix) — MC6805R2, MC6805U2, MC6805R3, MC6805U3	θ_{JA}	60	°C/W
Ceramic — MC6805R2, MC6805U2, MC6805R3, MC6805U3, MC68705R3, MC68705U3, MC68705R5, MC68705U5		50	
Cerdip — MC6805R2, MC6805U2, MC6805R3 MC6805U3, MC68705R5, MC68705U5		60	

11.3 POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

Where K is a constant pertaining to the particuiar 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.

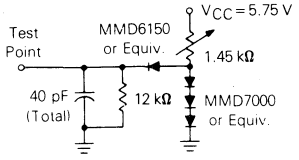


Figure 11-1. TTL Equivalent Test Load (Port B)

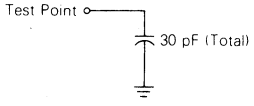


Figure 11-2. CMOS Equivalent Test Load (Port A)

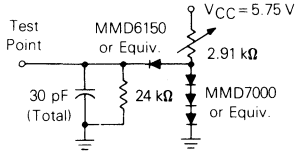


Figure 11-3. TTL Equivalent Test Load (Ports A and C)

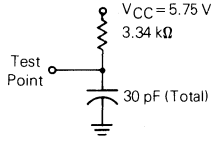


Figure 11-4. Open-Drain Equivalent Test Load (Port C)

11.4 MC6805R2 AND MC6805R3

11.4.1 Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} – 0.5	–	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} < 4.75)		V_{CC} – 0.5	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer					
Timer Mode	V_{IH}	2.0	–	V_{CC} + 1.0	V
Self-Check Mode		9.0	10.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other (Except A/D Inputs)		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltages (See Figures 7-1, 7-2, and 7-3)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{INT}$ Zero Crossing Input Voltage, Through a Capacitor	V_{INT}	2	–	4	$V_{ac\ p\text{-}p}$
Power Dissipation – (No Port Loading, V_{CC} = 5.75 V T_A = 0°C	P_D	–	520	740	mW
for Steady-State Operation) T_A = –40°C		–	580	800	
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other Except Analog Inputs (See Note)		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	2.75	3.75	4.70	V
Input Current					
TIMER (V_{in} = 0.4)	I_{in}	–	–	20	μA
INT (V_{in} = 2.4 V to V_{CC})		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option)		–	–	10	
(V_{in} = 0.4 V Crystal Option)		–	–	– 1600	
$\overline{RESET}$ (V_{in} = 0.8 V)	I_{RES}	–4.0	–	–40	
(External Capacitor Charging Current)					

NOTE: Port D Analog Inputs, when selected, C_{in} = 25 pF for the first 5 out of 30 cycles.
*Due to internal biasing this input (when unused) floats to approximately 2.0 V.

MC6805R2 AND MC6805R3

11.4.2 Switching Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time ($4/f_{osc}$)	t_{cyc}	0.95	–	10	μs
$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{INT}$ Zero-Crossing Detection Input Frequency	f_{INT}	0.03	–	1	kHz
External Clock Input Duty Cycle (EXTAL)	–	40	50	60	%
Crystal Oscillator Start-Up Time*	–	–	–	100	ms

*See Figure 7-5 for typical crystal parameters.

11.4.3 A/D Converter Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Non-Linearity	–	–	±1/2	LSB	For V_{RH} = 4.0 to 5.0 V and V_{RL} = 0 V
Quantizing Error	–	–	±1/2	LSB	
Conversion Range	V_{RL}	–	V_{RH}	V	
V_{RH}	–	–	V_{CC}	V	A/D accuracy may decrease proportionately as
V_{RL}	V_{SS}	–	0.2	V	V_{RH} is reduced below 4.0 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.
Conversion Time	30	30	30	t_{cyc}	Includes sampling time
Monotonicity	Inherent (within total error)				
Zero Input Reading	00	00	01	hexadecimal	V_{in} = 0
Ratiometric Reading	FE	FF	FF	hexadecimal	$V_{in} = V_{RH}$
Sample Time	5	5	5	t_{cyc}	
Sample/Hold Capacitance, Input	–	–	25	pF	
Analog Input Voltage	V_{RL}	–	V_{RH}	V	Negative transients on any analog lines (Pins 19-24) are not allowed at any time during conversion

MC6805R2 AND MC6805R3

11.4.4 Port Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Output High Voltage I_{Load} = – 10 μA	V_{OH}	V_{CC} – 1	–	–	V
Input High Voltage I_{Load} = – 300 μA (max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage I_{Load} = – 500 μA (max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	– 300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	– 500	μA
Port B					
Output Low Voltage I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage I_{Load} = – 200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source) V_O = 1.5 V	I_{OH}	– 1.0	–	– 10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	10	μA
Port C and Port A with CMOS Device Disabled					
Ouput Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	10	μA
Port C (Open-Drain Option)					
Input High Voltage	V_{IH}	2.0	–	13.0	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Leakage Current (V_{in} = 13.0 V)	I_{LOD}	–	<3	15	μA
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Port D (Digital Inputs Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Current*	I_{in}	–	<1	5	μA

*PD4/V_{RL} – PD5/V_{RH}: The A/D conversion resistor (15 kΩ typical) is connected internally between these two lines, impacting their use as digital inputs in some applications.

11.5 MC6805U2 AND MC6805U3

11.5.1 Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} – 0.5	–	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} < 4.75)		V_{CC} – 0.5	*	V_{CC}	
All Other (Except Timer)		2.0	–	V_{CC}	
Input High Voltage Timer					
Timer Mode	V_{IH}	2.0	–	V_{CC} + 1.0	V
Self-Check Mode		9.0	10.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltages (See Figures 7-1, 7-2, and 7-3)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{INT}$ Zero Crossing Voltage, Through a Capacitor	V_{INT}	2	–	4	$V_{ac\ p\text{-}p}$
Internal Power Dissipation (No Port Loading, V_{CC} = 5.75 V for Steady-State Operation) T_A = 0°C	P_{INT}	–	520	740	mW
T_A = –40°C		–	580	800	
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	2.75	3.75	4.70	V
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{INT}$ (V_{in} = 2.4 V to V_{CC})		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option)		–	–	10	
(V_{in} = 0.4 V Crystal Option)		–	–	–1600	
$\overline{RESET}$ (V_{in} = 0.8 V)	I_{RES}	–4.0	–	–40	
(External Capacitor Charging Current)					

* Due to internal biasing, this input (when unused) floats to approximately 2.0 V.

11.5.2 Switching Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time (4/f_{osc})	t_{cyc}	0.95	–	10	μs
$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Cap = 1 μF)	t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)	–	40	50	60	%
Crystal Oscillator Start-Up Time*	–	–	–	100	ms

* See Figure 7-5 for typical crystal parameters.

MC6805U2 AND MC6805U3

11.5.3 Port Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS drive enabled					
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Output High Voltage I_{Load} = −10 μA	V_{OH}	V_{CC}−1.0	–	–	V
Input High Voltage I_{Load} = −300 μA (max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage I_{Load} = −500 μA (max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	−300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	−500	μA
Port B					
Output Low Voltage I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage I_{Load} = −200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source) V_O = 1.5 V	I_{OH}	−1.0	–	−10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	$\|I_{TSI}\|$	–	<2	10	μA
Port C and Port A with CMOS drive disabled					
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	$\|I_{TSI}\|$	–	<2	10	μA
Port C (Open-Drain Option)					
Input High Voltage	V_{IH}	2.0	–	13.0	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Leakage Current (V_{in} = 13.0 V)	I_{LOD}	–	<3	15	μA
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Current	I_{in}	–	<1	5	μA

11.6 MC68705R3 AND MC68705R5

11.6.1 Programming Operation Electrical Characteristics (V_{CC} = 5.25 Vdc ±0.5, V_{SS} = 0, T_A = 20° to 30°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 5.25 V V_{PP} = 21.0 V	I_{PP}	– –	– –	8 30	mA
Oscillator Frequency	$f_{osc(p)}$	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) @ I_{IHTP} = 100 µA Max	V_{IHTP}	9.0	12.0	15.0	V

11.6.2 Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage $\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) $\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) All Other	V_{IH}	4.0 V_{CC} – 0.5 4.0 V_{CC} – 0.5 2.0	– – ** ** –	V_{CC} V_{CC} V_{CC} V_{CC} V_{CC}	V V V
Input High Voltage (TIMER Pin) Timer Mode Bootstrap Programming Mode	V_{IH}	2.0 9.0	– 12.0	V_{CC} + 1.0 15.0	V V
Input Low Voltage $\overline{RESET}$ $\overline{INT}$ All Other	V_{IL}	V_{SS} V_{SS} V_{SS}	– ** –	0.8 1.5 0.8	V V V
$\overline{INT}$ Zero-Crossing Input Voltage – Through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p\text{-}p}$
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V T_A = 0°C for Steady-State Operation T_A = –40°C	P_{INT}	– –	520 580	740 800	mW
Input Capacitance EXTAL All Other (See Note)	C_{in}	– –	25 10	– –	pF pF
$\overline{RESET}$ Hysteresis Voltage (See Figure 7-1) Out of Reset Voltage Into Reset Voltage	V_{IRES+} V_{IRES-}	2.1 0.8	– –	4.0 2.0	V V
Programming Voltage (V_{PP} Pin) Programming EPROM Operating Mode	V_{PP}*	20.0 4.75	21.0 V_{CC}	22.0 5.75	V V
Input Current TIMER (V_{in} = 0.4 V) $\overline{INT}$ (V_{in} = 0.4 V) EXTAL (V_{in} = 2.4 V to V_{CC}) (V_{in} = 0.4 V) $\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	I_{in} I_{RES}	– – – – –4.0	– 20 – – –	20 50 10 –1600 –40	µA

* V_{PP} is pin 7 on the MC68705R3 and MC68705R5 and is connected to V_{CC} in the normal operating mode. In the MC6805R2, pin 7 is connected to V_{SS} in the normal operating mode. The user must allow for this difference when emulating the MC6805R2 ROM-based MCU.

** Due to internal biasing, this input (when not used) floats to approximately 2.0 V.

NOTE: Port D analog inputs, when selected, C_{in} = 25 pF for the first 5 out of 30 cycles.

MC68705R3 AND MC68705R5

11.6.3 Switching Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	f_{osc}	0.4	–	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	μs
$\overline{INT}$, $\overline{INT2}$, or Timer Pulse Width	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Cap = 1.0 μF)	t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Duty Cycle (EXTAL)	–	40	50	60	%
Crystal Oscillator Start-Up Time*	–	–	–	100	ms

*See Figure 7-5 for typical crystal parameters.

3

11.6.4 A/D Converter Characteristics (V_{CC} = +5.25 V ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Non-Linearity	–	–	±1/2	LSB	For V_{RH} = 4.0 to 5.0 V and V_{RL} = 0 V.
Quantitizing Error	–	–	±1/2	LSB	
Conversion Range	V_{RL}	–	V_{RH}	V	
V_{RH}	–	–	V_{CC}	V	A/D accuracy may decrease proportionately as V_{RH} is reduced below 4.0 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.
V_{RL}	V_{SS}	–	0.2	V	
Conversion Time	30	30	30	t_{cyc}	Includes sampling time
Monotonicity	Inherent (within total error)				
Zero Input Reading	00	00	01	hexadecimal	V_{in} = 0
Ratiometric Reading	FE	FF	FF	hexadecimal	V_{in} = V_{RH}
Sample Time	5	5	5	t_{cyc}	
Sample/Hold Capacitance, Input	–	–	25	pF	
Analog Input Voltage	V_{RL}	–	V_{RH}	V	Negative transients on any analog lines (pins 19-24) are not allowed at any time during conversion.

MC68705R3 AND MC68705R5

11.6.5 Port Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} −1.0	–	–	V
Input High Voltage, I_{Load} = −300 μA (Max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = −500 μA (Max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	−300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	−500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = −200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	−1.0	–	−10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	10	μA
Port C					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	10	μA
Port D (Input Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Current	I_{in}	–	<1	5	μA

11.7 MC68705U3 AND MC68705U5

11.7.1 Programming Operation Electrical Characteristics (V_{CC} = 5.25 Vdc ±0.5, V_{SS} = 0 Vdc, T_A = 20° to 30°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 5.25 V V_{PP} = 21.0 V	I_{PP}	 – –	 – –	 8 30	mA
Programming Oscillator Frequency	f_{OSCP}	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin)(@ I_{IHTP} = 100 μA Max)	V_{IHTP}	9.0	12.0	15.0	V

MC68705U3 AND MC68705U5

11.7.2 Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ ($4.99 \le V_{CC} \le 5.51$)	V_{IH}	4.0	–	V_{CC}	V
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{INT}$ ($4.99 \le V_{CC} \le 5.51$)		4.0	**	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	**	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage (TIMER Pin)					
Timer Mode	V_{IH}	2.0	–	$V_{CC}+1.0$	V
Bootstrap Programming Mode		9.0	12.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	**	1.5	
All Other		V_{SS}	–	0.8	
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V T_A = 0°C for Steady-State Operation)	P_{INT}	–	520	740	mW
T_A = −40°C		–	580	800	
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
$\overline{INT}$ Zero-Crossing Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p-p}$
$\overline{RESET}$ Hysteresis Voltage (See Figure 7-1)					
Out of Reset Voltage	V_{IRES+}	2.1	–	4.0	V
Into Reset Voltage	V_{IRES-}	0.8	–	2.0	
Programming Voltage (V_{PP} Pin)					
Programming EPROM	V_{PP}*	20.0	21.0	22.0	V
Operating Mode		4.75	V_{CC}	5.75	
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{INT}$ (V_{in} = 0.4 V)		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option)		–	–	10	
(V_{in} = 0.4 V Crystal Option)		–	–	−1600	
$\overline{RESET}$ (V_{in} = 0.8 V)	I_{RES}	−4.0	–	−40	
(External Capacitor Changing Current)					

*V_{PP} is Pin 7 on the MC68705U3 and MC68705U5 and is connected to V_{CC} in the Normal Operating Mode. In the MC6805U2, Pin 7 is NUM and is connected to V_{SS} in the Normal Operating Mode. The user must allow for this difference when emulating the MC6805U2 ROM-based MCU.

**Due to internal biasing, this input (when not used) floats to approximately 2.0 V.

11.7.3 Switching Characteristics (V_{CC} = +5.25 Vdc, ±0.5 V, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency					
Normal	f_{osc}	0.4	–	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	μs
$\overline{INT}$, $\overline{INT2}$, or Timer Pulse Width	t_{WL}, t_{WH}	$t_{cyc}+250$	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	$t_{cyc}+250$	–	–	ns
$\overline{RESET}$ Delay Time (External Cap = 1.0 μF)	t_{RHL}	100	–	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Duty Cycle (EXTAL)	–	40	50	60	%
Crystal Oscillator Start-Up Time*	–	–	–	100	ms

*See Figure 7-5 for typical crystal parameters.

MC68705U3 AND MC68705U5

11.7.4 Port Electrical Characteristics (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC} − 1.0	–	–	V
Input High Voltage, I_{Load} = −300 μA (Max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = −500 μA (Max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	−300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	−500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = −200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	−1.0	–	−10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	0	μA
Port C					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	<2	10	μA
Port D (Input Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Current	I_{in}	–	<1	5	μA

11.8 I/O CHARACTERISTICS

Figures 11-5 through 11-15 illustrate I/O characteristic data for HMOS M6805 Family devices. Simplified port logic diagrams are shown in Figures 11-16 and 11-17, typical input protection in Figure 11-18, and an I/O characteristic measurement circuit in Figure 11-19. The I/O characteristic curves and logic diagrams are intended to allow the system designer to interface the M6805 in a variety of applications where non-TTL loading conditions exist.

A minimum specification curve (included with V_{OH} vs I_{OH} charts only) is provided as a guaranteed limit of performance under the conditions shown. The expected minimum and maximum curves in each figure represent the anticipated performance window under normal manufacturing and operating conditions. A typical curve also is illustrated indicating performance under nominal conditions.

Figure 11-15 represents the variation of I_{DD} with temperature and V_{DD} for a typical M6805 Family device. As shown, I_{DD} varies directly with V_{DD} and inversely with temperature.

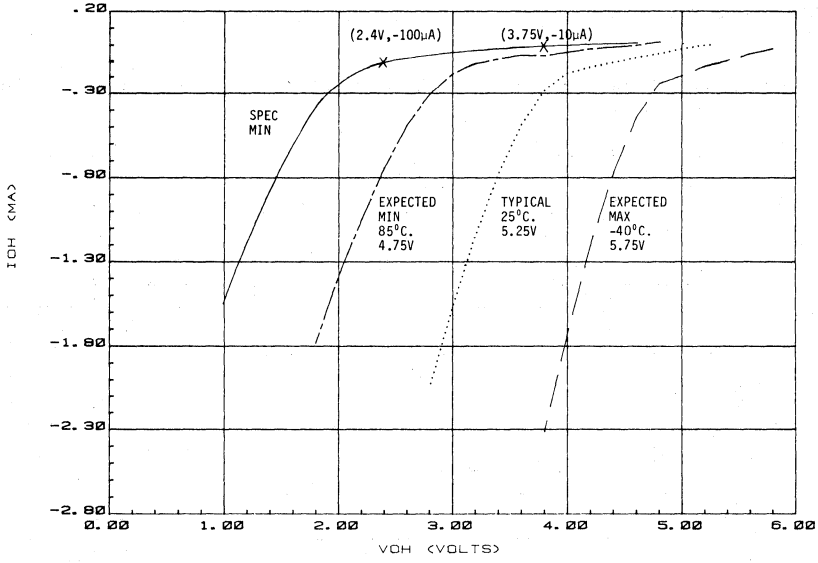


Figure 11-5. Port A V_{OH} vs I_{OH}
(with CMOS Pull-ups)

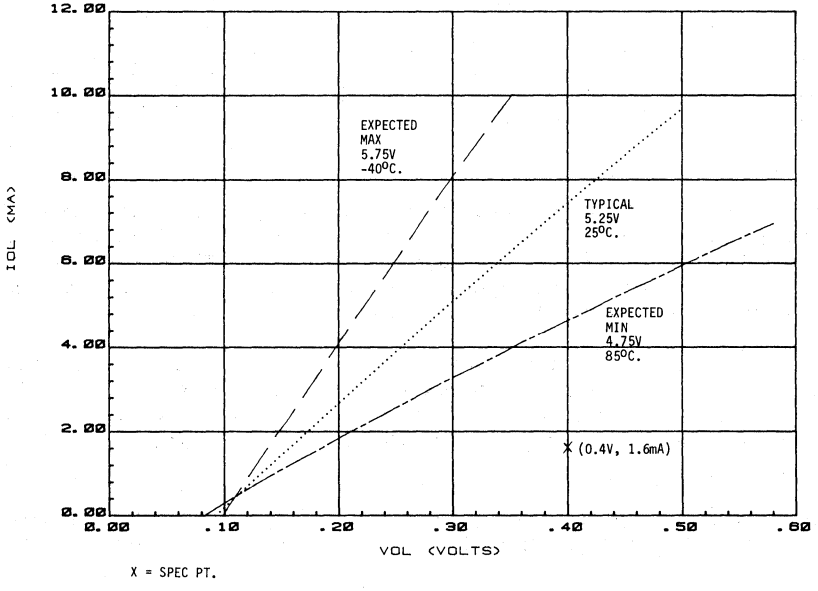


Figure 11-6. Port A V_{OL} vs I_{OL}
(with CMOS Pull-ups)

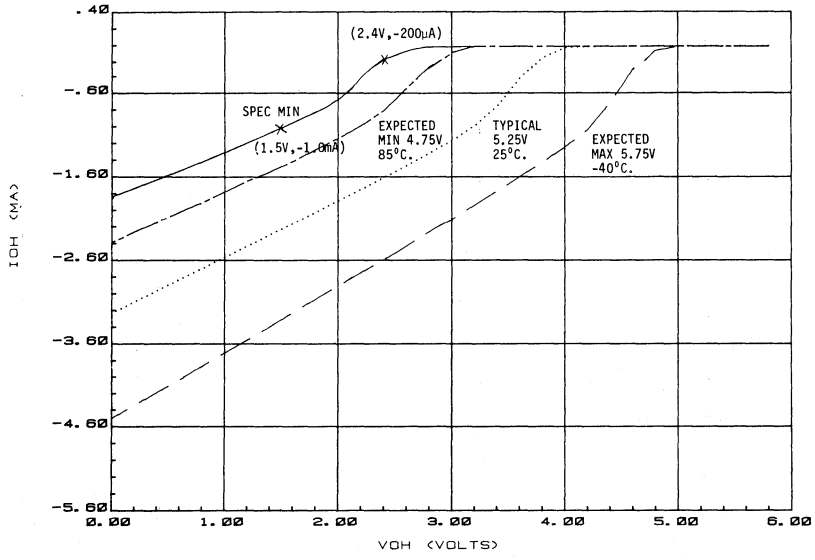


Figure 11-7. Port B V_{OH} vs I_{OH}

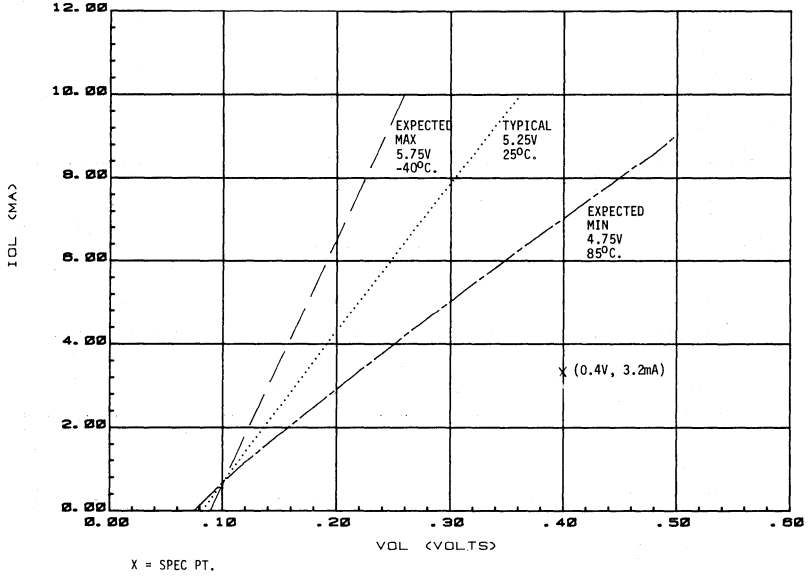


Figure 11-8. Port B V_{OL} vs I_{OL}

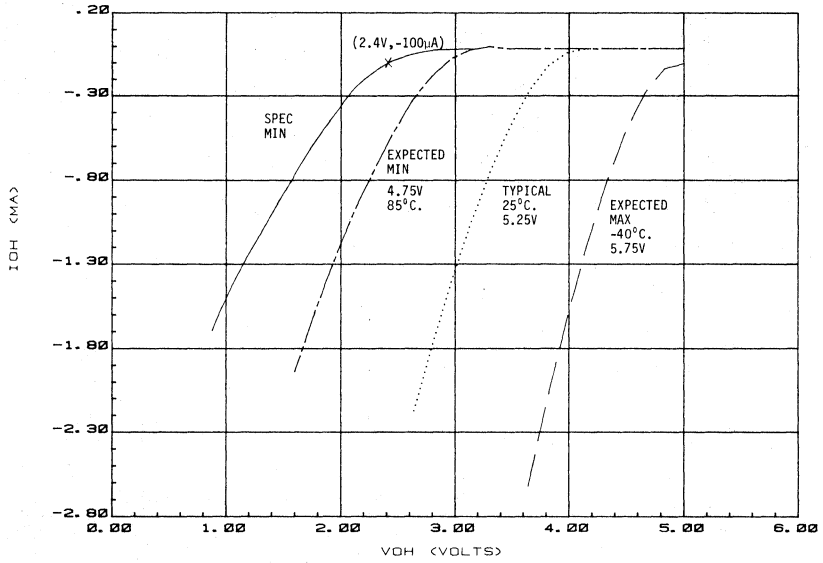


Figure 11-9. Port C V_{OH} vs I_{OH}
(Port A Without CMOS Pull-ups)

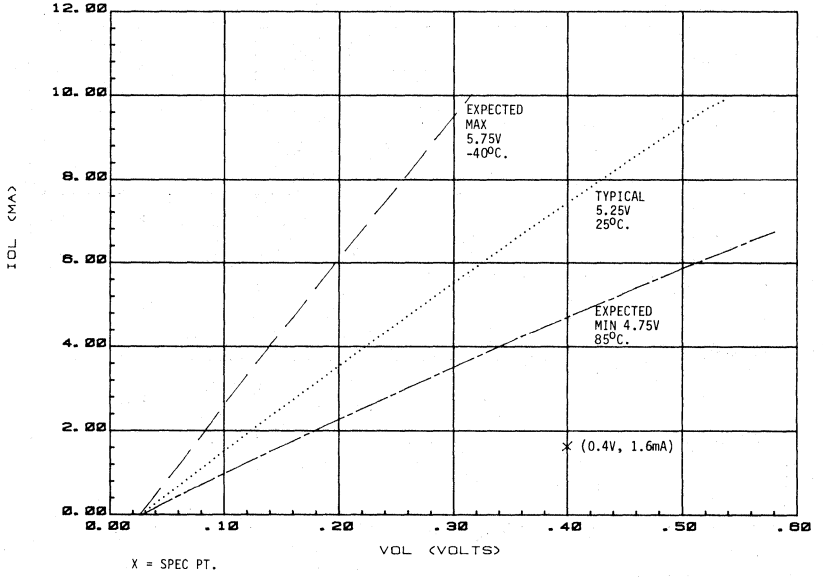


Figure 11-10. Port C V_{OL} vs I_{OL}
(Port A Without CMOS Pull-ups)

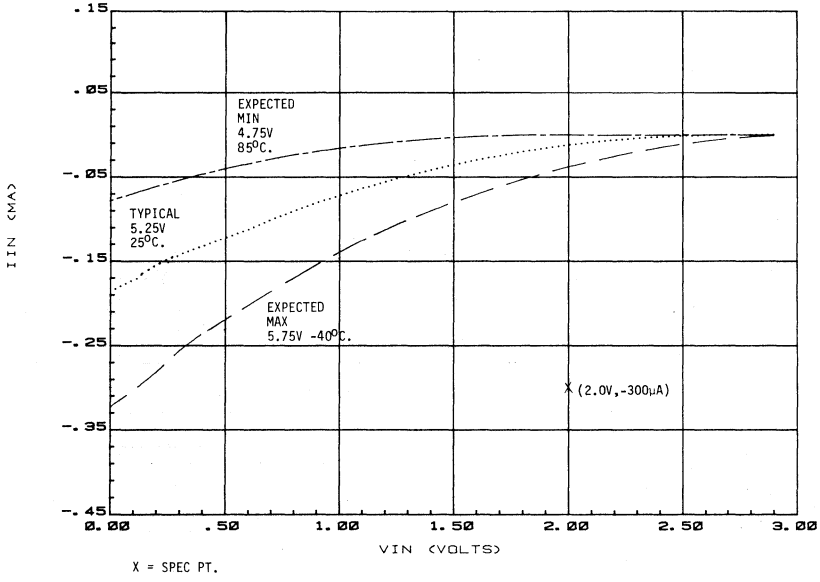


Figure 11-11. Port A V_{in} vs I_{in}
(with CMOS Pull-ups)

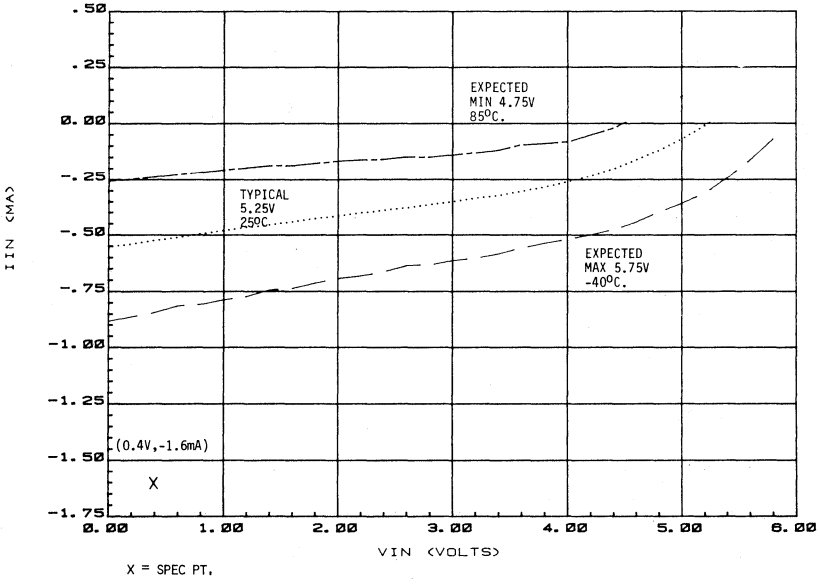


Figure 11-12. EXTAL V_{in} vs I_{in}

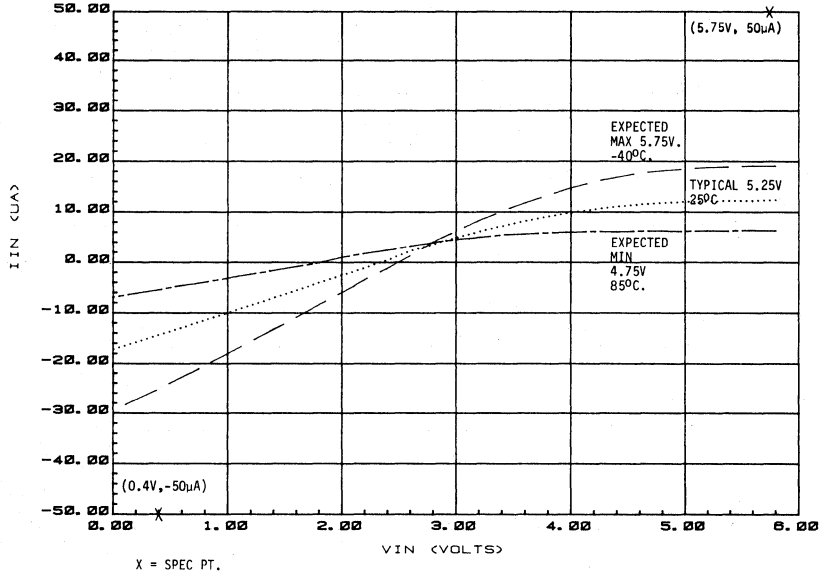


Figure 11-13. Interrupt V_{in} vs I_{in}

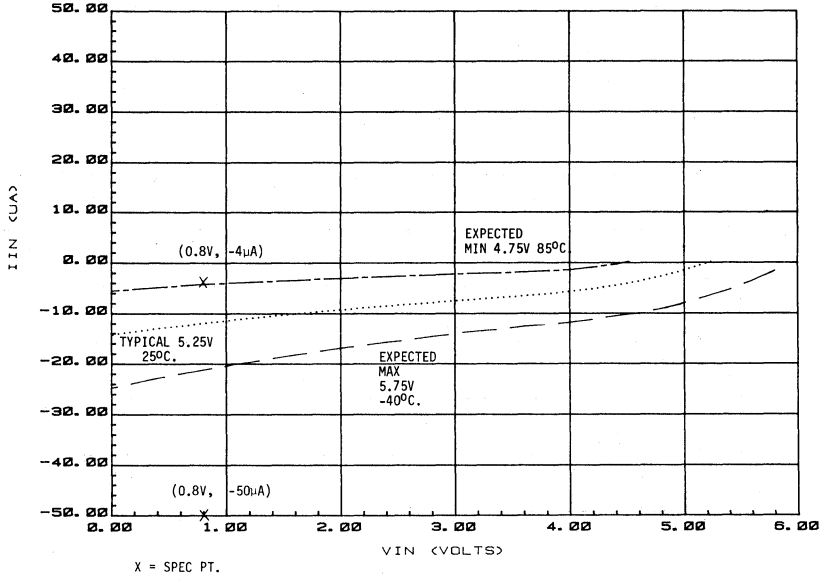


Figure 11-14. $\overline{\text{RESET}}$ V_{in} vs I_{in}

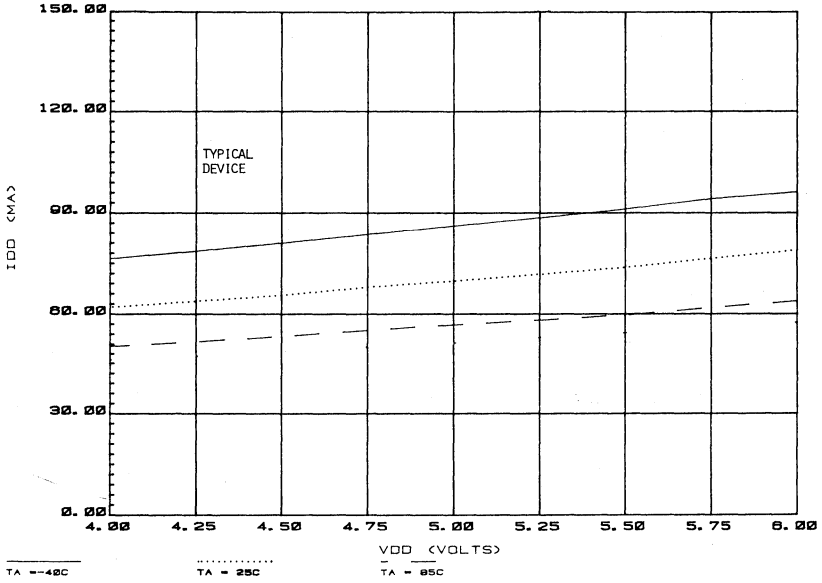


Figure 11-15. V_{DD} vs I_{DD}
(Variation with Temperature)

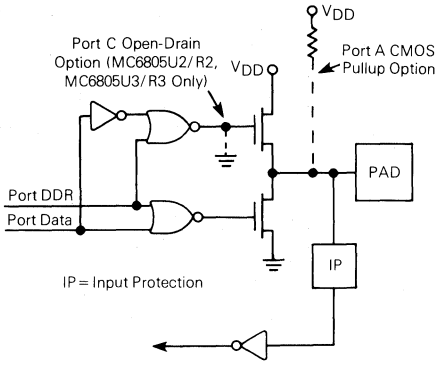


Figure 11-16. Ports A and C Logic Diagram

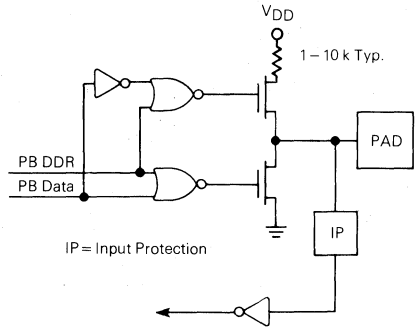


Figure 11-17. Port B Logic Diagram

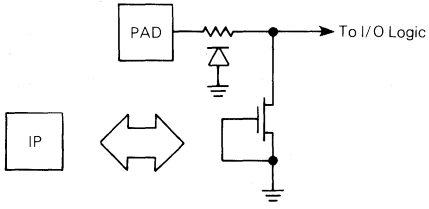


Figure 11-18. Typical Input Protection

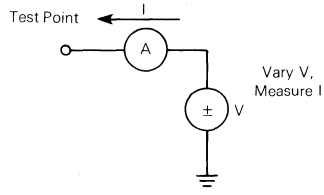


Figure 11-19. I/O Characteristic Measurement Circuit

SECTION 12
ORDERING INFORMATION

This section contains detailed information to be used as a guide when ordering an MC68(7)05 series device.

12.1 MC6805R2

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC6805R2L
L Suffix	−40°C to 85°C	MC6805R2CL
Plastic	0°C to 70°C	MC6805R2P
P Suffix	−40°C to 85°C	MC6805R2CP
Cerdip	0°C to 70°C	MC6805R2S
S Suffix	−40°C to 85°C	MC6805R2CS

12.2 MC6805R3

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC6805R3L
L Suffix	−40°C to 85°C	MC6805R3CL
Plastic	0°C to 70°C	MC6805R3P
P Suffix	−40°C to 85°C	MC6805R3CP
Cerdip	0°C to 70°C	MC6805R3S
S Suffix	−40°C to 85°C	MC6805R3CS

12.3 MC6805U2

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC6805U2L
L Suffix	−40°C to 85°C	MC6805U2CL
Plastic	0°C to 70°C	MC6805U2P
P Suffix	−40°C to 85°C	MC6805U2CP
Cerdip	0°C to 70°C	MC6805U2S
S Suffix	−40°C to 85°C	MC6805U2CS

12.4 MC6805U3

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC6805U3L
L Suffix	−40°C to 85°C	MC6805U3CL
Plastic	0°C to 70°C	MC6805U3P
P Suffix	−40°C to 85°C	MC6805U3CP
Cerdip	0°C to 70°C	MC6805U3S
S Suffix	−40°C to 85°C	MC6805U3CS

12.5 MC68705R3

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC68705R3L
L Suffix	−40°C to 85°C	MC68705R3CL

12.6 MC68705R5

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC68705R5L
L Suffix	−40°C to 85°C	MC68705R5CL
Cerdip	0°C to 70°C	MC68705R5S
S Suffix	−40°C to 85°C	MC68705R5CS

12.7 MC68705U3

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC68705U3L
L Suffix	−40°C to 85°C	MC68705U3CL

12.8 MC68705U5

Package Type	Temperature	Generic Number
Ceramic	0°C to 70°C	MC68705U5L
L Suffix	−40°C to 85°C	MC68705U5CL
Cerdip	0°C to 70°C	MC68705U5S
S Suffix	−40°C to 85°C	MC68705U5CS

12.9 CUSTOM MCUs

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the MCU it is necessary to first contact your local Motorola representative or Motorola distributor.

12.9.1 EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure for two MCM2716 EPROMs is shown in Figure 12-1.

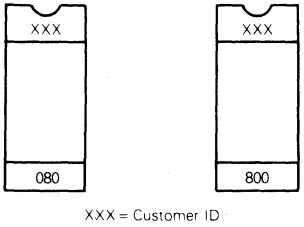


Figure 12-1. Recommended Marking Procedure

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

12.9.2 Verfication Media

All original pattern media (EPROMs or floppy disk) are filed for contractural 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 one blank EPROM from the data file used to create the custom mask to aid in the verification process.

12.9.3 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 5 volts. These RVUs are included in the mask charge and are not production parts. Therefore, the RVUs are not guaranteed by Motorola Quality Assurance, and should be discarded after verification is completed.

12.9.4 Flexible Disk

The disk media submitted must be single-sided, single-density, 8-inch MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The minimum MDOS system files as well as the absolute binary object file (filename .LO type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: filename .LX (EXORciser loadable format) and filename .SA (ASCII source code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

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

Date _____ Customer PO Number _____

Customer Company _____ Motorola Part Number
MC _____
Address _____ SC _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

3

MC6805R2/U2/R3/U3 OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Timer Clock Source
MC6805R2/U2 Only
- ☐ Internal $\phi 2$ clock
- ☐ TIMER input pin

Timer Prescaler
MC6805R2/U2 Only
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Port C Output Drive
- ☐ TTL
- ☐ Open-Drain

Pattern Media (All other media requires prior factory approval)
- ☐ EPROMs (MCM2716 or MCM2532)
- ☐ Floppy Disk
- ☐ Other* _____

Clock Frequency _____

Temperature Range _____
- 0°C to +70°C (Standard)
- −40°C to +85°C
- −40°C to +105°C

Marking Information (12 Characters Maximum)

Title _____

Signature _____

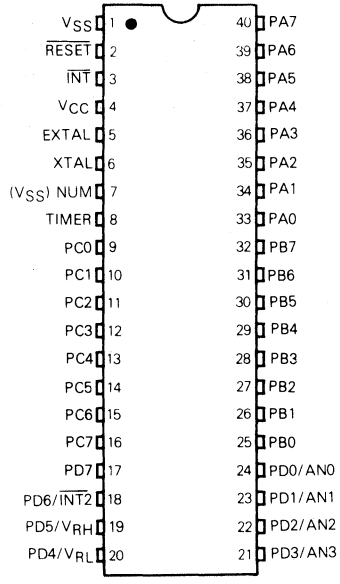
Figure 12-2. Sample Custom MCU Order Form

SECTION 13
MECHANICAL DATA

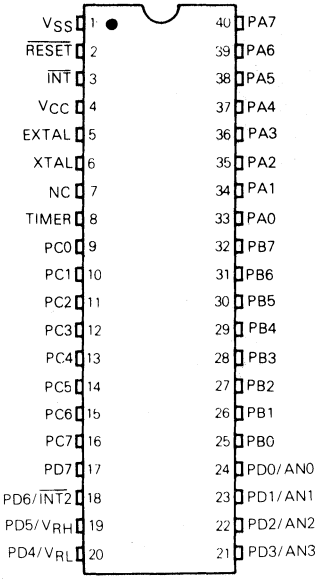
This section contains the pin assignments and package dimensions for the MC68(7)05 series devices.

13.1 PIN ASSIGNMENT

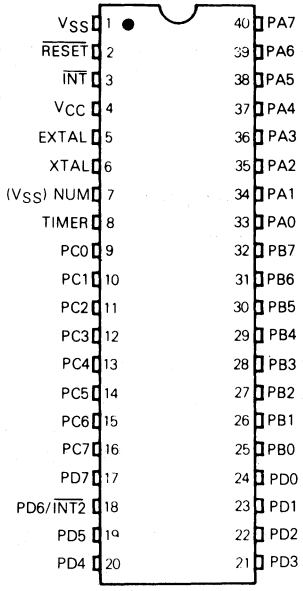
MC6805R2



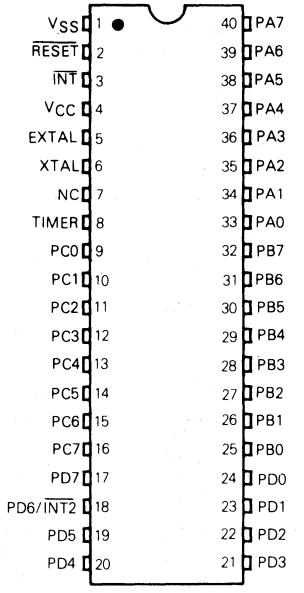
MC6805R3



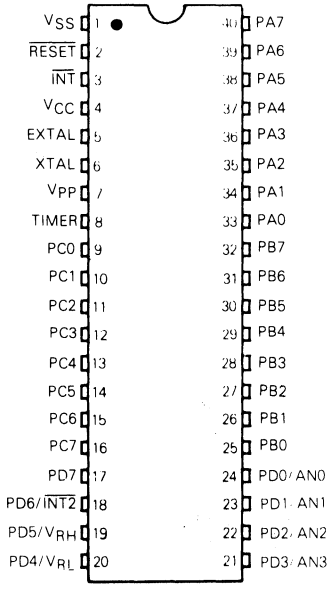
MC6805U2



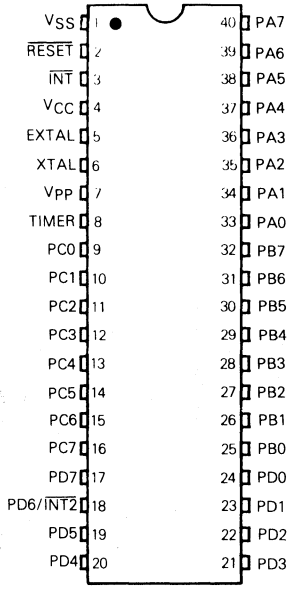
MC6805U3



MC68705R3/MC68705R5



MC68705U3/MC68705U5



MOTOROLA

3

MC6805S2

8-BIT MICROCOMPUTER WITH ANALOG-TO-DIGITAL CONVERTER, SERIAL PERIPHERAL INTERFACE, AND THREE TIMERS

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

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

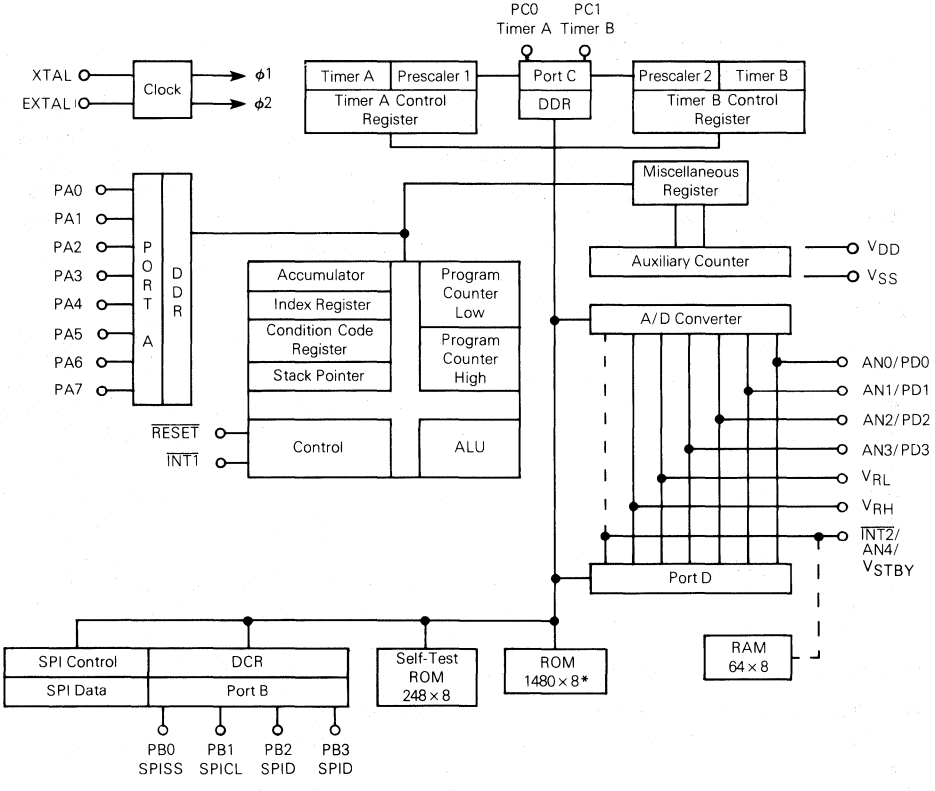
LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

SECTION 1
INTRODUCTION

The MC6805S2 microcomputer unit (MCU) is a member of the M6805 Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip clock, ROM, RAM, I/O, 4-channel 8-bit analog-to-digital (A/D) converter, three timers, two programmable prescalers, and a serial peripheral interface (a block diagram is shown in Figure 1-1). It is designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set.



*Includes 8 bytes for interrupt vectors.

Figure 1-1. Block Diagram

1.1 HARDWARE FEATURES

The following are some of the hardware features of the MC6805S2 MCU.

- A/D Converter
  - 8-Bit Conversion, Monotonic
  - Four Multiplexed Analog Inputs
  - Ratiometric Conversion
- 21 TTL Including Eight TTL/CMOS Compatible I/O Lines
  - 14 Bidirectional (Four Lines are LED Compatible)
  - 7 Input-Only
- 1480 Bytes of User ROM
- 64 Bytes of RAM
- Self-Check Mode
- Serial Peripheral Interface (SPI)
- Zero-Crossing Detect/Interrupt
- One 8-Bit and One 16-Bit Timer
- One 7-Bit and One 15-Bit Software Programmable Prescaler
- Three Bidirectional I/O Lines with TTL or Open-Drain Interface (Software Programmable)
- Auxiliary Counter with "Watchdog" Reset Feature
- 5-Volt Single Supply

1.2 SOFTWARE FEATURES

The following are some of the software features of the MC6805S2 MCU.

- 10 Powerful Addressing Modes
- Byte Efficient Instruction Set with True Bit Manipulation, Bit Test, and Branch Instructions
- Single Instruction Memory Examine/Change
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- User Callable Self-Check Subroutines
- Complete Development System Support on EXORciser, EXORset, and HDS-200 Available Now

1.3 USER SELECTABLE OPTIONS

The following are user selectable options of the MC6805S2 MCU.

- Eight Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer/SPI, Software, and External
- 16-Bytes Standby RAM Option
- Fifth A/D Channel Option

EXORciser is a registered trademark of Motorola Inc.
EXORset and HDS-200 are trademarks of Motorola Inc.

SECTION 2
SIGNAL DESCRIPTION, MEMORY, CPU, AND REGISTERS

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

2.1 SIGNAL DESCRIPTION

The following paragraphs provide a brief description of the signals and a reference (if applicable) to other paragraphs that contain more detail about the function being performed.

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 $\overline{INT1}$ and $\overline{INT2}$

These pins provide the capability for asynchronously applying an external interrupt to the MCU. Refer to **5.4 INTERRUPTS** for additional information.

2.1.3 XTAL and EXTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the user selected manufacturing mask option, can be connected to these pins to provide a system clock source with various stability/cost tradeoffs. Lead length and stray capacitance on these two pins should be minimized. Refer to **5.3 INTERNAL CLOCK GENERATOR OPTIONS** for recommendations about these inputs.

2.1.4 Timer A/PC0 and Timer B/PC1

These pins allow an external input to be used to decrement the internal timer circuitry. Refer to **SECTION 3 TIMERS** for additional information about the timer circuitry.

2.1.5 $\overline{RESET}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to **5.2 RESETS** for additional information.

2.1.6 Input/Output Lines (PA0-PA7, PB0-PB3, PC0-PC1, and PD0-PD6)

Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D has up to four analog inputs or five via mask option, plus two voltage reference inputs when the analog-to-digital (A/D) converter is used (PD5/V_{RH}, PD4/V_{RL}, and an $\overline{INT2}$ input). If any analog input is used, then the voltage reference pins (PD5/V_{RH}, PD4/V_{RL}) must be used in the analog mode. Refer to **6.1 INPUT/OUTPUT, 6.6 ANALOG-TO-DIGITAL CONVERTER (A/D),** and **5.4 INTERRUPTS** for additional information. Port D can also be used as a 7-bit digital input-only port.

2.2. MEMORY

As shown in Figure 2-1, the MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The MC6805S2 MCU has implemented 1802 of these locations. This consists of: 1480 bytes of user ROM including eight interrupt vectors, 248 bytes of self-check ROM, 64 bytes of user RAM, seven bytes of port I/O, five timer registers, two A/D registers, a miscellaneous register, and two serial peripheral interface (SPI) registers. The user ROM has been split into three areas. The first area is memory locations $080 to $0FF, and allows the user to access these ROM locations utilizing the direct and table look-up indexed addressing modes. The main user ROM area is from $9C0 to $EFF. The last eight user ROM locations at the top of memory are for the interrupt vectors.

The MCU reserves the first 18 memory locations for I/O features, of which 17 have been implemented. These locations are used for the ports, the port DDRs, the timers, the miscellaneous register, the SPI, and the A/D. Of the 64 RAM bytes, 31 are shared with the stack area, from $061 through $07F. The stack must be used with care when data shares the stack area. The lower 16 bytes of RAM, between $40 and $4F, may be powered through the $\overline{INT2}$/PD6 pin via a user-defined mask option. Selection of this option does not exclude any of the available functions of the $\overline{INT2}$/PD6 input.

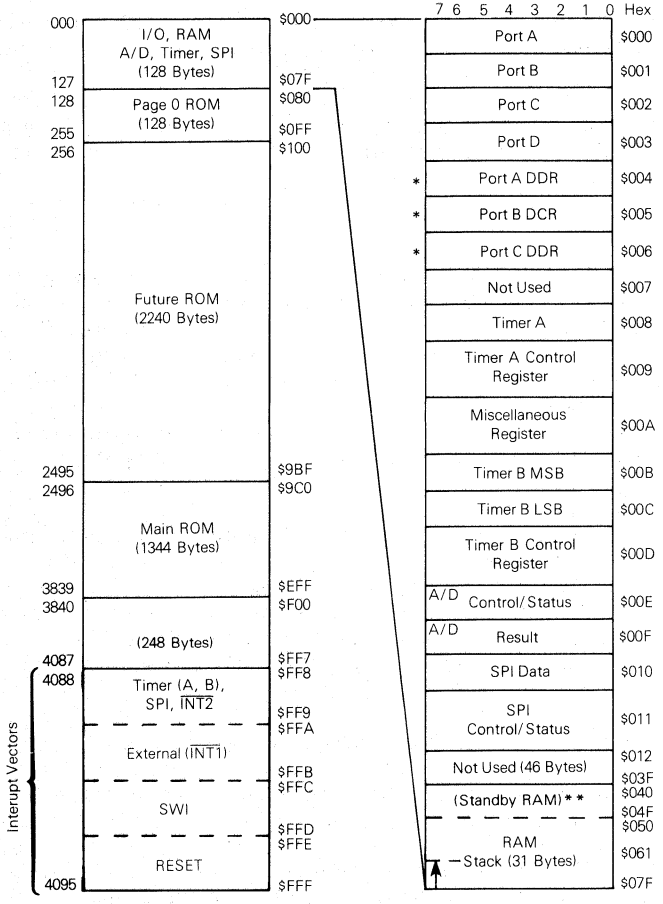
The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in the order shown in Figure 2-2. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack, since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed.

2.3 CENTRAL PROCESSING UNIT

The CPU of the M6805 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 address, data, and control buses.

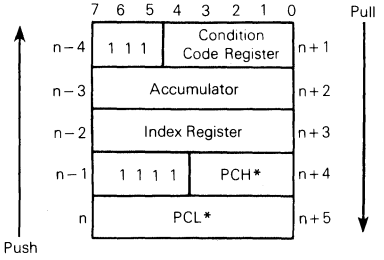
2.4 REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 2-3 and are explained in the following paragraphs.



* Data direction registers (DDRs) are write only; they read as $FF.
** Mask Option

Figure 2-1. Address Map



*For subroutine calls, only PCH and PCL are stacked.

Figure 2-2. Interrupt Stacking Order

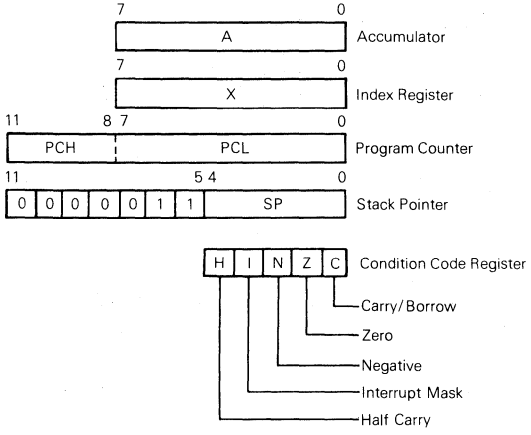


Figure 2-3. Programming Model

2.4.1 Accumulator (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

2.4.2 Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instructions and as a temporary storage area.

2.4.3 Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

2.4.4 Stack Pointer (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack. The seven most significant bits of the stack pointer are permanently configured to 0000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

2.4.5 Condition Code Register (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs. For more information concerning the condition code register refer to the *M6805 HMOS/M146805 CMOS Family Users Manual.*

2.4.5.1 HALF CARRY (H) — Set during ADD and ADC operations to indicate that a carry occurred before bits 3 and 4.

2.4.5.2 INTERRUPT (I) — When set, this bit masks (disables) the timer (both A and B), external ($\overline{INT1}$ and $\overline{INT2}$), and the serial peripheral interface interrupts. If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

2.4.5.3 NEGATIVE (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

2.4.5.4 ZERO (Z) — When set, this bit indicates that the result of the last arithmetic, logic, or data manipulation was zero.

2.4.5.5 CARRY/BORROW (C) — When set, this bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

SECTION 3
TIMERS

The MC6805S2 has three timers and two programmable prescalers (see Figure 3-1) which are described in this section.

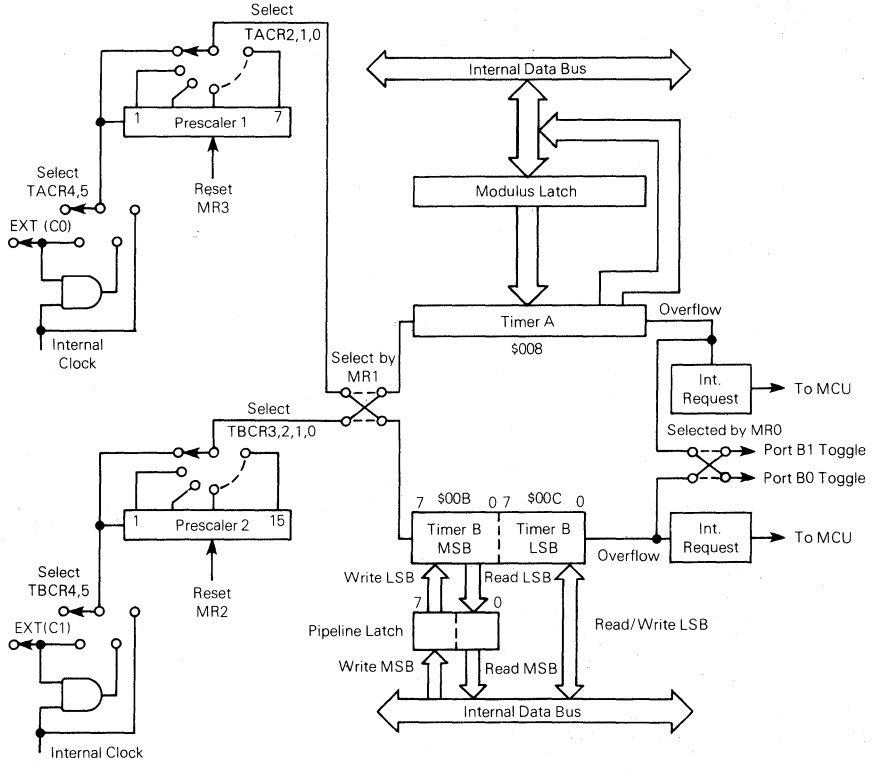


Figure 3-1. Timer A and B Block Diagram

3.1 TIMER A

Timer A is an 8-bit programmable down counter, which may be loaded under program control (see Figures 3-2 and 3-3). Included in this timer is a modulus latch which allows the timer to be "auto reloaded." Address $08 is the location of TIMER A's data register. Upon every clock input received, timer A decrements toward $00. Upon reaching this value, bit 7 in the timer A control register (TACR located at $09) is set, signifying a timer interrupt has been generated. At the same time, the timer is reloaded with the contents of the modulus latch. In addition to setting the interrupt bit, the transition to state $00 also generates an overflow condition which can be used to toggle bit 0 or bit 1 of port B directly, under the control of the miscellaneous register bit 3 (MR3), the serial peripheral interface control register, and the port B data direction register. The bit selected depends upon the state of bit 0 of the miscellaneous register. The timer interrupt may be masked by setting bit 6 of the TACR. Of course, the I bit in the condition code register will also prevent a timer interrupt from being processed. The timer interrupt vector locations are $FF8 and $FF9. The timer interrupt request bit *MUST* be cleared by software.

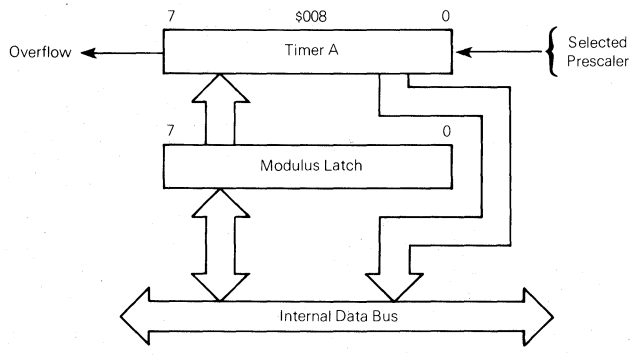


Figure 3-2. Timer A

TACR7	TACR6	TACR5	TACR4	TACR3	TACR2	TACR1	TACR0

TACR7 = Timer A Interrupt Request Flag
TACR6 = Timer A Interrupt Request Mask
TACR5 = External/Internal
TACR4 = External Enable
TACR3 = Timer A Load Mode Control
TACR2, TACR1, TACR0 } Prescaler 1 Divide Ratio Select

TACR5	TACR4	Prescaler 1 Clock
0	0	Internal Clock
0	1	AND of Internal Clock•PC0
1	0	Clock Disabled
1	1	PC0 Positive Transition

Figure 3-3. Timer A Control Register

There are three ways of loading data from the modulus latch into timer A as described in the following paragraphs.

3.1.1 Direct Loading

When the MCU writes to the timer A data register, the data is latched by the modulus latch, and forced into the timer. For this operation, TACR bit 3 must be clear.

3.1.2 Asynchronous External Event Loading

When TACR bit 3 is a logic one, the contents of the modulus latch are transferred to the timer at the rising edge of the $\overline{INT2}$ interrupt request bit (MR7) gated with interrupt request mask bit (MR6).

NOTE

If this feature is used, then care must be taken in programming as it will start an interrupt service routine if the I bit in condition code register (CC) is clear.

Loading $00 to timer A allows a countdown of 256 clocks before next $00 state is reached.

3.1.3 Auto-Loading

Auto-loading of the modulus latch occurs whenever the timer reaches the $00 state. This mode is independent of the status of bit 3 in TACR.

NOTE

Loading modes 1 and 2 are mutually exclusive, and auto-loading occurs in both modes 1 and 2.

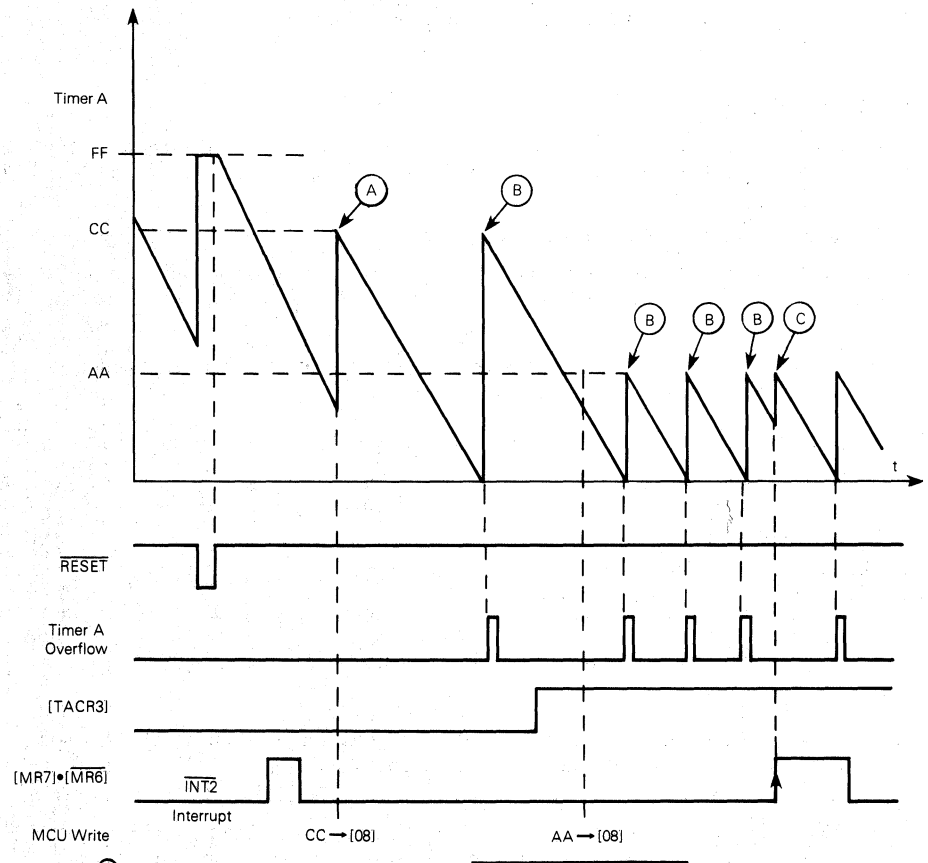
Timer A may be read at any time without disturbing the countdown mechanism of the timer. At reset, both the timer and modulus latch contents are set to $FF.

NOTE

Loading $01 to timer A should be avoided when operating with a divide-by-one prescaler. Doing so will inhibit timer A auto-loading, interrupt generation, and port B toggle mechanisms.

3.2 TIMER A CONTROL REGISTER

Timer A control register (TACR) occupies memory location $09 (see Figure 3-4). Five bits are allocated to timer A and three bits are used to control prescaler 1.



TACR3	Load Mechanism
0	A + B
1	B + C

Figure 3-4. Timer A Operation

TACR7 Timer A Interrupt Request Flag
Set at the transition of timer A to $00 state. Cleared by software or at reset.

TACR6 Timer A Interrupt Request Mask
Set at reset or under program control. When set, timer A interrupt requests to the processor are inhibited. Cleared under program control.

TACR5 External or Internal Bit
Set under program control. When set, selects the input clock source for prescaler 1 to be the PC0 input, otherwise the internal clock (f_{OSC} divided by four) is the input clock source. Cleared at reset or under program control.

TACR4 External Enable Bit
Control bit used to enable the external timer pin (PC0).

TACR5	TACR4	Prescaler 1 Clock Source
0	0	Internal Clock
0	1	AND of Internal Clock and PC0*
1	0	Inputs Disabled
1	1	PC0* Low-to-High Transition

TACR3 Timer A Load Mode Control
Set under program control. When set, allows asynchronous external event loading of timer A ($\overline{INT2}$ driven loading is enabled). Cleared under program control or at reset. When clear, allows direct loading of timer A. Auto-loading takes place independent of TACR3 status. Cleared by reset or by program control.

TACR2, TACR1, TACR0 Prescaler 1 Division Ratio Control Bits
Set or cleared under program control, also cleared at reset. When set, these bits select one of the eight possible outputs on prescaler 1.

TACR2	TACR1	TACR0	Prescaler 1 Division Ratio
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

3.3 TIMER B

Timer B is a 16-bit timer which is accessed via two registers at $0B for the most-significant byte (MSB) and $0C for the least-significant byte (LSB) (refer to Figure 3-5). Included within the MSB of timer B is a "pipeline" latch, which allows a "snap shot" value of the entire 16 bits to be read.

---

* The status of PC0 depends upon the data direction status of PC0. If PC0 is an output, then the clock source is equal to the port data register content, independent of the port electrical loading. If an input, then the clock source is the logic level of PC0.

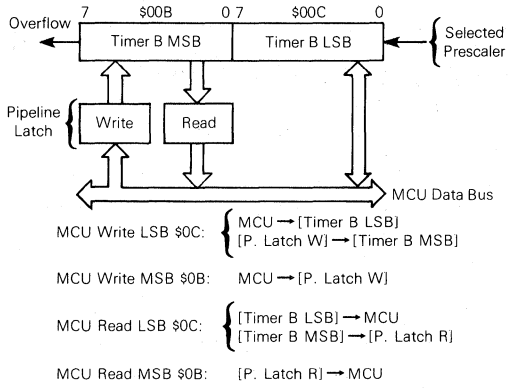


Figure 3-5. Timer B

Read/write operations to the LSB are direct. Reading the LSB can occur at any time without disturbing the count. At the time of the LSB read, the contents of the MSB are loaded into the pipeline latch, so when the MPU reads the MSB, it actually reads the latch.

Writing to the LSB of timer B may occur at any time, and the contents are immediately entered into the timer. At the same time the contents of the pipeline latch are forced into the MSB of the timer. Hence, a 16-bit word may be placed into the entire timer data register during a LSB write operation.

In order to manipulate a 16-bit word in timer B during a read, it is recommended that a read of the LSB be done first, then the MSB. A 16-bit write should be done in the opposite order. (First, write the MSB and then the LSB.) After reset, timer B contains $FFFF.

Like timer A, timer B decrements toward zero upon every clock input received and during the transition to state $00 TBCR7 in the timer B control register is set (TBCR is located at $0D).

The timer interrupt can be masked by setting the timer interrupt mask bit (TBCR6) (Figure 3-6). The I bit in the condition code register will also prevent a timer interrupt from being processed. The MCU responds to a timer interrupt by saving the current MCU state in the stack, fetching the vector from $FF8 and $FF9, and executing the interrupt routine. The timer interrupt routine bit *MUST* be cleared by software.

TBCR7	TBCR6	TBCR5	TBCR4	TCBR3	TBCR2	TBCR1	TBCR0

TBCR7 = Timer B Interrupt Request Flag
TBCR6 = Timer B Interrupt Request Mask
TBCR5 = External/Internal
TBCR4 = External Enable
TBCR3, TBCR2, TBCR1, TBCR0 } Prescaler 2 Divide Ratio Select

TBCR5	TBCR4	Prescaler 2 Clock
0	0	Internal
0	1	AND of Internal Clock•PC1
1	0	Clock Disabled
1	1	PC1 Positive Transition

Figure 3-6. Timer B Control Register

The transition to $00 generates an overflow pulse which may be used to force a port B data register toggle under the control of the miscellaneous register bit 3 (MR3), the SPI control register, and the port B data direction register. (See **6.5 MISCELLANEOUS REGISTER** and **4.3 SERIAL PERIPHERAL INTERFACE CONTROL AND STATUS REGISTER**.)

3.4 TIMER B CONTROL A STATUS REGISTER

Timer B control and status register (TBCR) occupies memory location $0D (see Figure 3-6). Four bits are allocated to timer B and four bits are used to control prescaler 2.

TBCR7 Timer B Interrupt Request Flag
Set at the transition of timer B to $00. Cleared by software or at reset.

TBCR6 Timer B Interrupt Request Mask
Set at reset or under program control. When set, inhibits timer B interrupt requests to the processor. Cleared under program control.

TBCR5 External or Internal Bit
Set under program control. When set, selects the input clock source for prescaler 2 to be the PC1 input, otherwise the internal clock (f_{OSC} divided by four) is the input clock source. Cleared at reset or under program control.

TBCR4 External Enable Bit
Set under program control or at reset. When set, this bit enables the external timer pin (PC1). Cleared under program control.

TBCR5	TBCR4	Prescaler 2 Clock Source
0	0	Internal Clock
0	1	AND of Internal Clock and PC1*
1	0	Inputs Disabled
1	1	PC1* Low-to-High Transition

TBCR3
TBCR2
TBCR1
TBCR0
Prescaler 2 Division Ratio Control Bits
Set or cleared under program control. When set, these bits select one of the 16 possible outputs on prescaler 2. All bits are cleared at reset.

*PC1 status depends on the data direction status of PC1. If PC1 is an output, then the clock source is equal to the port data register content, independent of the port electrical loading. If an input then the clock source is the logic level on PC1.

TBCR3	TBCR2	TBCR1	TBCR0	Division Ratio
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	512
1	0	1	0	1024
1	0	1	1	2048
1	1	0	0	4096
1	1	0	1	8192
1	1	1	0	16384
1	1	1	1	32768

3.5 PRESCALER 1

Prescaler 1 is a 7-bit binary down counter whose value is selected by TACR2, TACR1, and TACR0. The selected output is used as the clock input to either timer A or timer B, depending upon the status of the prescaler cross-couple bit (MR1). The type of clock source to prescaler 1 may be selected by TACR5 and TACR4 (see **3.1 TIMER A**).

Prescaler 1 is set to $7F at reset or under program control when a one is written to prescaler 1 clear bit (MR3).

NOTE

When changing outputs on the prescaler, a prescaler clear should be done first to avoid truncation errors.

3.6 PRESCALER 2

Prescaler 2 is a 15-bit down counter whose value is selected by TBCR3, TBCR2, TBCR1, and TBCR0. The selected output is used as the clock input to either timer A or timer B, depending upon the status of MR1. The type of clock source to prescaler 2 can be selected by TBCR5 and TBCR4 (see **3.3 TIMER B**).

Prescaler 2 is preset to $7FFF at reset, under program control when a logic one is written to prescaler 2 clear bit (MR2).

NOTE

When changing outputs on the prescaler, a prescaler clear should be done first to avoid truncation errors.

3.7 AUXILIARY COUNTER

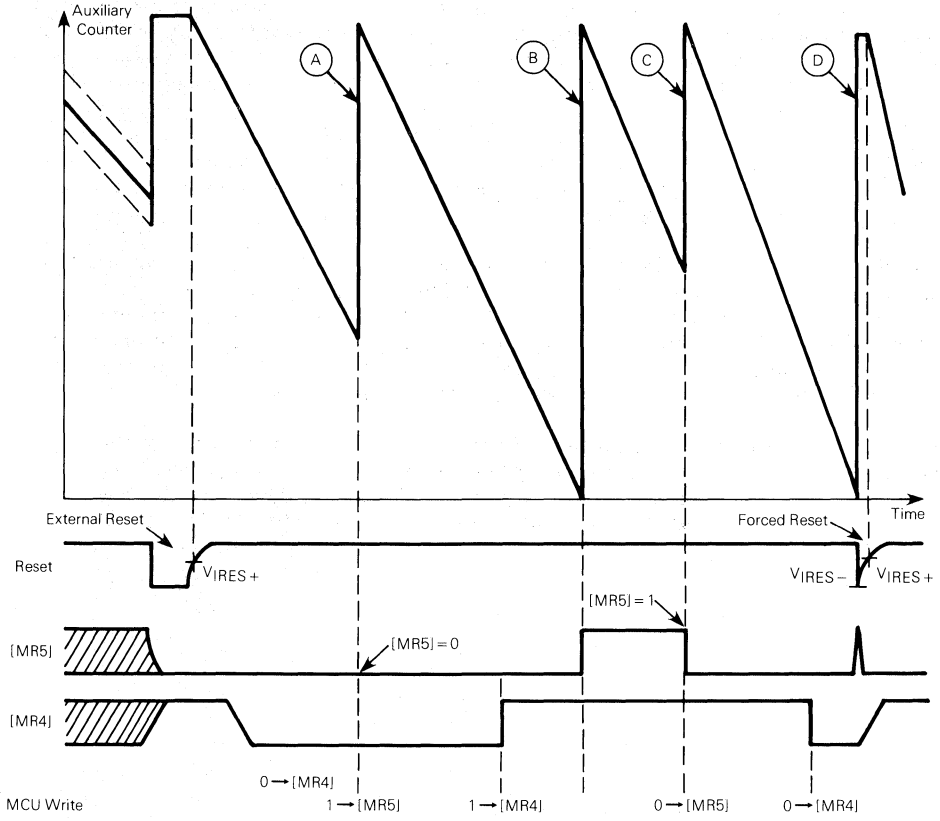
The third timer register in the MC6805S2 is the auxiliary counter, or "watchdog" timer. It is a fixed counter which is clocked by the internal clock (f_{OSC} divided by four). The total count period is 4095

cycles. The MCU communicates with this counter via the miscellaneous register bits 5 and 4 (MR5, MR4). Upon overflow, the auxiliary counter control/status bit in the miscellaneous register (MR5) is set. Countdown may be aborted at any time under program control, which also will reset the counter to 4095. To do this, the MCU must write to MR5 the inverse of what is read from MR5.

At reset, the counter is preset to its maximum count of 4095, and MR5 is cleared. The value of the counter is not accessible to the MCU; however, the possibility of detecting an underflow and presetting it at any time under program control allows it to be used as a fixed rate polled timer in applications requiring lengthy time out periods.

When the auxiliary counter reset mask bit in the miscellaneous register (MR4) is clear and the MR5 is set as a result of counter time out, the reset pin is internally pulled to ground (V_{SS}). This feature is useful in many applications, e.g., automotive, where the MCU operates in a noisy environment. Due to high energy spikes on the power supply and I/O lines, the MCU may lose control of the program and execute through non-valid memory space. The "watchdog" timer will bring the MCU back to reset. MR4 is automatically set at reset or under program control.

To return from a catastropic system runaway, the reset line is pulsed, which will restart the entire program. This program should regularly preset the auxiliary counter at a rate higher than the counter time out so as not to allow a forced reset. If program runaway does occur, it is likely that regular presetting of the auxiliary counter will not take place, and an overflow will force the MCU to regain control. (See Figure 3-7.)



(A) Counter Preset by Writing "1"

(B) Overflow: MR5 → 1; No Forced Reset

(C) Counter Reset by Writing "0"

(D) Overflow MR5 → 1 Forced Reset

Figure 3-7. Auxiliary Counter Operation

SECTION 4
SERIAL PERIPHERAL INTERFACE (SPI)

This section describes the operation of the serial peripheral interface (SPI) on the MC6805S2.

4.1 INTRODUCTION

The serial peripheral interface (SPI) on the MC6805S2 has several versatile operating modes. Arbitration on data and clock lines is provided. The SPI communicates with the MCU via data and control registers located at memory addresses $10 and $11, respectively. Operation of the SPI occurs via port B (see Figure 4-1).

The SPI consists of:

a) an 8-bit shift register (MSB out first; MSB in first) which may also be used as an SPI data register,
b) a divide-by-eight counter,
c) slave select/arbitration logic,
d) an SPI control register, and
e) start and stop bit detection capability.

Unlike the I/O port operation, the SPI data and clock inputs are always taken from their respective I/O port pins, regardless of the status of the data direction register relative to that port. This makes it easy to do data and clock arbitration.

Serial peripheral interface operation is enabled when the SPI enable bit (SPICR4) is set. When enabled, the SPI is capable of operating in the following modes:

a) one wire – auto clocked (e.g., NRZ),
b) two wire – half duplex,
c) two wire – half duplex with clock arbitration,
d) three wire – half duplex with slave select/busy line,
e) three wire – full duplex,
f) three wire – full duplex with clock arbitration, and
g) four wire – full duplex with slave select/busy line.

4.2 SPI TERMINOLOGY

The following explanations are provided to facilitate user understanding of the various operating modes of the serial peripheral interface (see Table 4-1).

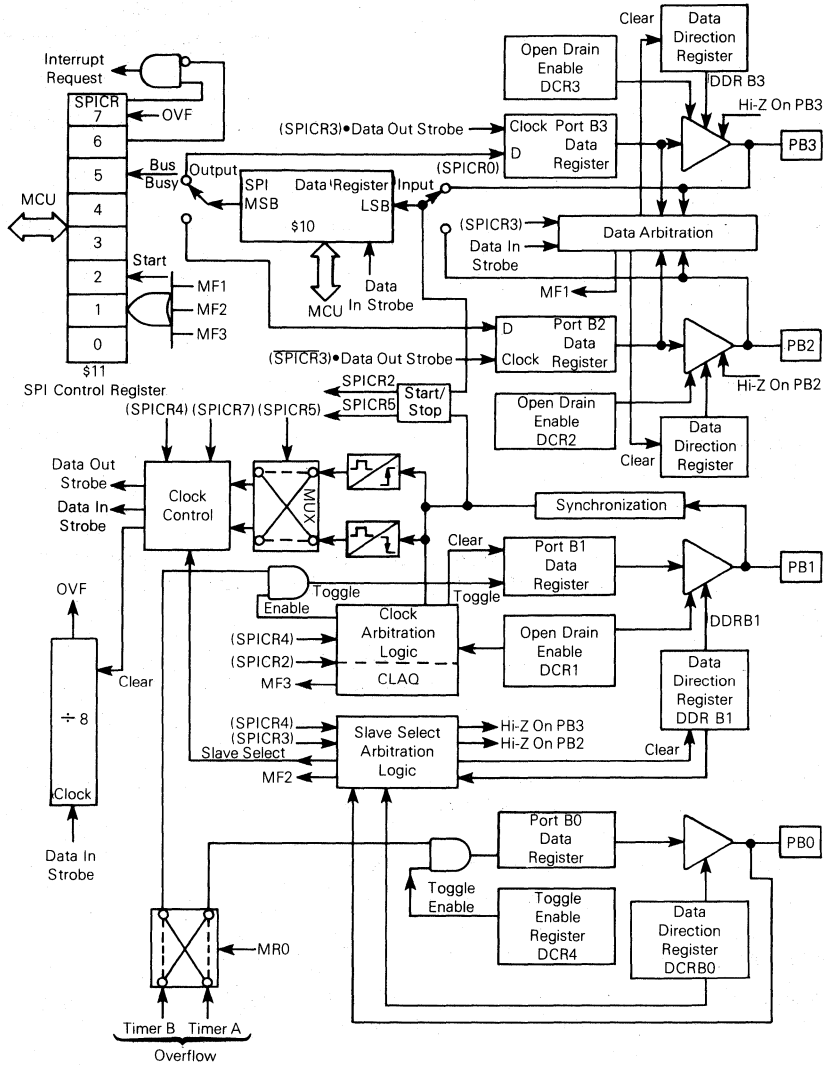


Figure 4-1. Serial Peripheral Interface (SPI)

Table 4-1. SPI Operation

DEFINTIONS

Transmitter – Data Master: DDRB2 or 3 = 1
Receiver – Data Slave: DDRB2 or 3 = 0
Clock Master: DDRB1 = 1
Clock Slave: DDRB1 = 0
Transaction Mode: SPICR4 = 1
1) Active: $\overline{SPICR7}$•($\overline{DDRB0}$•$\overline{PB0}$ + DDRB0) if DDRB1 = 0 (clock slave mode) or
$\overline{SPICR7}$•($\overline{DDRB0}$•PB0 + DDRB0) if DDRB1 = 1 (clock master mode)
Clock pulses allowed, data shifted
2) Idle: SPICR7 + $\overline{DDRB0}$•PB0 if DDRB1 = 0 (clock slave mode)
Clock pulses blocked, data output line in high-impedance state
Deselect Mode: SPICR4 = 0 – No SPI Operations

SLAVE SELECT INPUT

Slave Select Input: SPISS – PB0
If DDRB0 = 0 then no SPISS action on MCU
1) Master Mode: SPISS = 1 DDRB1 = 1
SPISS 1 – 0: Switch to Slave Mode (DDRB1 1 – 0)
Set SPICR1 (Mode Fault Flag)
2) Slave Mode: SPISS = 0 DDRB1 = 0
External clock is allowed to shift data in/out. If SPISS is pulled high, the external clock input pulses are inhibited; no data shift; divide-by-eight counter cleared; SPID (PB2 or PB3) switched to high-impedance state.
Used as Chip-Select Input

DATA ARBITRATION

Data master loses data mastership when data collision occurs during internal data strobe time.

If SPID output port (PB2 or PB3) = 1 while actual pin level is pulled low externally – conflict detected at internal data strobe time.

Then SPICR1 (mode fault flag) is set; SPID output port DDR (B2 or B3) 1 → 0 (high-impedance state).

CLOCK ARBITRATION

MCU has clock mastership (DDRB1 = 1)
1) Via SPISS line (DDRB0 = 0). If SPISS is pulled low, then clock mastership lost; DDRB1 1 → 0 (high-impedance state); SPICR1 is set (mode fault flag).
2) Via clock line SPICL (DDRB1 = 1 and DCRB5 = 0)
Condition: SPICL must have open-drain output (DCRB5 = 0)

If clock line is held low externally then clock mastership is not lost; minimum t_{CLH} and t_{CLK} times are guaranteed.

If SPICL goes low during idle mode then SPICR1 = 1 and clock line is switched low to inhibit the system clock.

MODE FAULT FLAG OPERATION (SPICR1)

Flag set when any of the following conditions occur.
Data arbitration occurs on SPID output.
Clock arbitration with SPISS during master to slave switching.
Clock arbitration via clock line if SPICL 1 → 0 during idle.

START, STOP, AND CLOCK IDLE CONDITIONS

Clock Idle: The clock level just prior to the transition that causes data on the serial output data line to be changed is defined as the SPI clock idle state.
SPICR5 = 0: SPICL Idle = Low State
SPICR5 = 1: SPICL Idle = High State

These definitions are necessary for determining start and stop conditions.

NOTE

Clock idle state can only be defined if SPICR4 = 0 (Deselect Mode)

Start Condition: Any negative transition of the data input line (PB2 or PB3) during an SPICL idle state.
Stop Condition: Any positive transition of the data input line during an SPICL idle state.

4.2.1 Clock Mastership

The SPI clock source is always taken from port B1. When the clock level on pin PB1 is defined by the MCU, it is said that the MCU has clock mastership. The principle condition for clock mastership during an SPI operation is that port B1 must be initialized by its DDR bit so that the port is in the output mode. When PB1 DDR is clear (i.e., configured as an input) during an SPI operation, and external device provides the SPI clock on pin PB1. This is referred to as the "clock slave" mode.

4.2.2 Data Mastership

SPI data transactions (transmission/reception) can occur through port B2, port B3, or through both of these ports as determined by the software. The MCU is said to have data mastership when the data output on the selected data output port is defined by the processor. The main requirement for data mastership during an SPI operation is that the selected SPI data output port, PB2 or PB3, be initialized by its DDR bit to be in the output mode. Routing of output data to the proper port data register is done by SPICR3. The MC6805S2 may be a "receiver" in any mode of operation.

4.2.3 SPI Transaction Mode

This is the mode where the SPI is allowed to operate (see Figure 4-2). Operation takes place via port B lines. SPI transactions are enabled when the SPI control register bit 4 (SPICR4) is set.

4.2.4 SPI Deselect Mode

SPICR4 is clear in this mode. All SPI operations and actions relative to the SPI operation are blocked in the SPI deselect mode. This mode is selected at reset.

4.2.5 SPI Active Mode

The SPI active mode is part of the transaction mode (Figure 4-3). In addition to the transaction mode requirements, the two following requirements must be met for the MC6805S2 to operate in the SPI active mode: 1) SPICR7 = 0, and SPISS (port B0) = 0 if PB1 DDR = 0 (clock slave mode) and 2) SPICR7 = 0 and SPISS = 1 if PB1 DDR = 1 (clock master mode). In this mode, the SPI clock pulses are allowed to shift serial information.

4.2.6 SPI Idle Mode

This is part of the transaction mode and is characterized additionally by 1) SPICR 7 = 1 or 2) slave select input (port B0) = 1 if DDRB1 = 0 (clock slave mode). In this mode all SPI clock pulses are blocked and, if the MCU is in the clock slave mode, the serial data out line is forced to high impedance if slave select input PB0 = 1. In this mode the MCU is processing serial data or is deselected under external control.

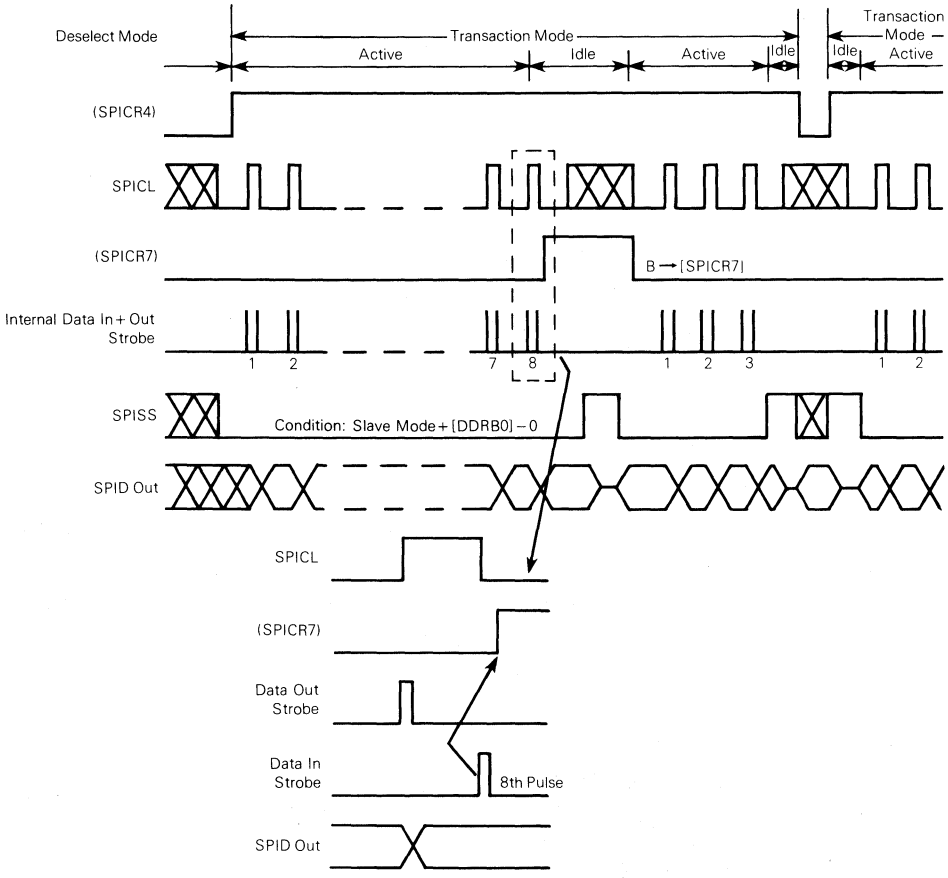


Figure 4-2. SPI Operation (Example: Clock Slave Mode)

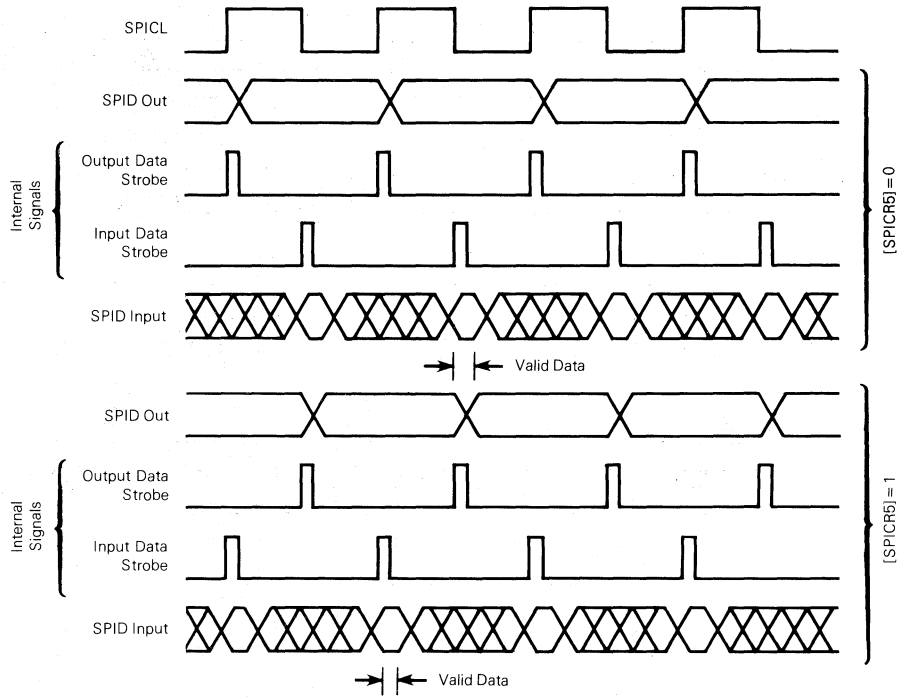


Figure 4-3. SPI Clock (Active Transaction)

4.3 SPI CONTROL AND STATUS REGISTER

This register, illustrated in Figure 4-4, contains the status and control bits related to SPI operations.

SPICR7 SPI Interrupt Request Bit
This bit is cleared at reset or under program control. When the eighth SPI data input strobe is detected from the SPI clock input this SPI interrupt request bit becomes set. When set, it forces the SPI into the idle mode. It remains in the idle mode until it is serviced. Only if SPICR7 is not masked by SPICR6 is the processor allowed to receive an interrupt request. The processor services this interrupt if the I bit is clear in the condition code register. It does so by fetching the interrupt vector from addresses $FF8 and $FF9. As long as SPICR7 is set the SPI remains in the idle mode during SPI transactions. SPICR7 is also cleared at the zero to one transition of SPICR2 due to a "start bit" detection during the transaction mode.

SPICR6 SPI Interrupt Request Mask Bit
This bit is set at reset or under program control. When set, it inhibits interrupt requests from SPICR7. This bit is cleared under program control, or at the zero to one transition of SPICR2 due to a "start bit" detection during the transaction mode.

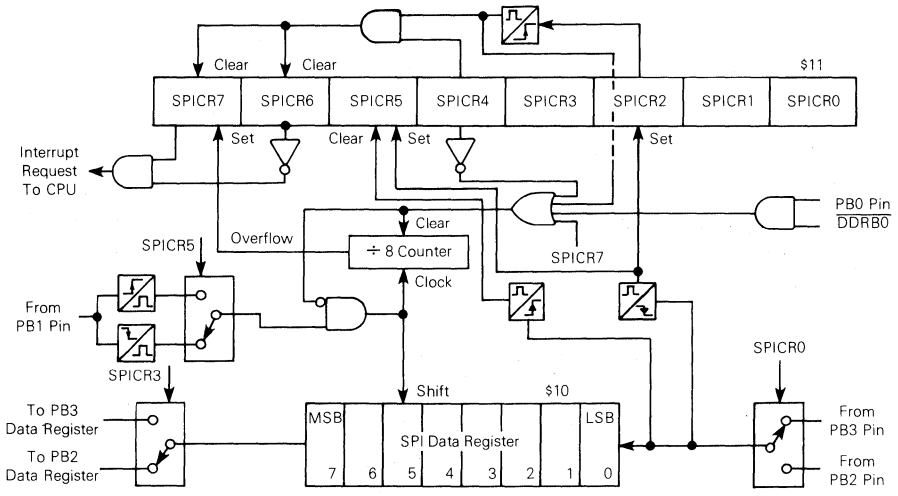


Figure 4-4. SPI Control Register Operation

SPICR5 SPI Clock Sense Bit/Bus-Busy Flag

This is a dual-function bit controlled by the status of SPICR4. The function of this bit is the following:

SPICR4	Mode	SPICR5 Function
0	SPI Deselect	SPI Clock Sense Bit (Read/Write)
1	SPI Transaction	SPI Bus-Busy Flag (Read Only)

If the SPI is in the deselect mode (SPICR4=0), SPICR5 becomes a read/write bit that controls the clock sense and SPICL idle level. When low, this bit causes SPI input data to be latched into the SPI data register on the negative edge of the SPI clock and output data to be changed on the positive edge of the SPI clock. This corresponds to a low SPICL idle level. When high, input data is latched on the positive edge and output data changed on the negative edge of the SPI clock. This corresponds to a high SPICL idle level. Data in the SPI data register is shifted by one location to the left at the SPI clock edge that latches SPI input data. This clock edge is referred to as the data input strobe.

During SPI operation (SPICR4=1), SPICR5 becomes a read-only bit that serves as a "bus-busy" flag. This flag is set due to a start condition and cleared due to a stop condition or at reset. A received MCU or a clock slave can poll this flag to determine the appropriate time to "capture the bus" and become a transmitter or clock master. This flag provides a "clean" hook-unhook mechanism to the serial bus to allow true multi-master operation.

In a properly ordered system, only one MCU has data mastership between a given start and stop condition. Outside this busy zone, the serial bus is considered free and is

signalled to the MCUs via the bus-busy flag. In the case that more than one processor attempts to gain access to the bus during this free zone, a normal data arbitration will take place. The MCUs with low priority can then get off the bus and remain as slaves until the next free zone.

SPICR4 SPI Operation Enable Bit
This bit is cleared at reset or under program control. When set under program control, it allows SPI operation and actions relative to it. When it is cleared, the divide-by-eight counter is reset; the SPI data register is disabled from shifting; and data and clock arbitration logic, as well as the slave select input logic, actions are inhibited. Logic status of this bit determines which of the dual functions is performed by SPICR2 and SPICR5.

SPICR3 SPI Data Output Select Bit
This bit is cleared at reset or under program control. When set under program control, this bit allows the output of the SPI data register to be loaded to the port B3 data register at the appropriate SPI clock edge selected by SPICR5, during the active transaction mode. When clear, the port B2 data register is loaded with the output of the SPI data register at the appropriate SPI clock edge during the active transaction mode.

SPICR2 Port B1 Toggle Enable/Start Bit
This is a dual function bit controlled by the status of SPICR4. The function of this bit is the following:

SPICR4	Mode	SPICR2 Function
0	SPI Deselect	Port B1 Toggle Enable
1	SPI Transaction	Start Bit

During non-SPI operations (SPICR4 = 0), when set under program control, SPICR2 enables port B1 data register toggle facility. Its prime use is in applications where continuous toggle operation may be required. This bit is cleared under program control or at reset.

During SPI operation (SPICR4 = 1), this bit is set by the negative transition of the data input of the SPI data shift register while the clock is in its idle level. The (SPICL) idle level is defined as the high level of SPICL if SPICR5 = 1 or the low level of SPICR5 = 0. Note that SPICR5 must be defined during the SPI deselect mode (SPICR4 = 0).

At the protocol level, this means that a "start" condition may be defined as an exceptional change of state of data input while this condition does not occur or should not be allowed to occur during the data transmission. "Start" condition information may be used to distinguish address and data transmissions, as well as transmission resync after transmission synchronization has been lost. This bit is cleared or set under program control.

SPICR1 Mode Fault Flag
This bit is cleared at reset or under program control. It is set under the following conditions:

1) When SPI data output arbitration occurs on the SPI data output port (PB3 or PB2) selected by SPICR3, the SPI data output port DDR is cleared (switches to high-impedance state), MCU loses data mastership, and the mode fault flag is set.

2) When the MCU has clock mastership (i.e., port B1 DDR = 1), slave select input PB0, if used as such in the application, should stay high. If a low level is detected on this input, then the MCU loses clock mastership, switches to clock slave mode, the port B1 DDR is cleared, and the mode fault flag is set.

3) When the MCU operates in the master mode where clock arbitration is done via the clock line, then the mode fault flag is set during the idle mode when a negative clock edge is detected on the SPI clock input. Simultaneously the port B1 data register is cleared.

This feature allows the MCU to detect that some other device has attempted to drive the SPI clock input while the MCU was not ready to perform a serial transaction; or that MCU has lost data mastership or clock mastership.

SPICR0 SPI Input Data Select Bit
This bit is cleared at reset or under program control. When set under program control, it allows SPI data from port B3 to be latched into the SPI data register. When clear, SPI data from port B2 is routed to the input of the SPI data register.

4.4 SPI DATA REGISTER

This register can be written into at any time. It can be read "on the fly" irrespective of serial operation without disturbing the data. Data is shifted left by one bit every time there is a data input strobe while the LSB is loaded with data from port B2 or B3 according to the status of SPICR0.

The MSB is loaded to the data register of port B2 or B3 according to the status of SPICR3 every time there is a data output strobe. Data input and output strobes are generated during the transitions of the SPI clock input to the MCU under the control of SPICR5. Data input and output strobes are generated internally only during the active transaction time.

4.5 SPI DIVIDE-BY-EIGHT COUNTER

This counter is cleared during SPI deselect or idle modes. It counts at every data input strobe during the SPI active transaction mode. At overflow, it sets SPICR7, which in turn puts the SPI in the idle mode and blocks all data input and output strobes. This counter is also cleared when the slave select input (PB0) is high while the MCU is operating the SPI with slave select in the slave mode, or when a "start" condition is detected. Clearing of the counter by the "start" condition allows resynchronization of data transmission between MCUs.

4.6 SPI OPERATION

The SPI may operate in a variety of ways depending on user application needs. The main modes are described below; however, this list is neither exhaustive nor absolute. Software assisted protocols may be defined to upgrade the hardware versatility and/or system performance of the MC6805S2. Some features common to all operating modes are outlined below.

1) The SPI data input and output paths may be individually routed under program control via SPICR3 and SPICR0 to or from either PB2 or PB3 (see Table 4-2). This gives rise to four possible routings useful in half duplex and full duplex operations, as well as allowing bidirectional information to flow in daisy-chained systems.

2) When data input and output is done on the same pin (PB2 or PB3), i.e., SPICR3 ⊕ SPICR0 = 0, then half duplex operation takes place. The unused port line (PB2 or PB3) is free for any other use.

3) Data input is always relative to the port pin logic level regardless of the data direction register status on that pin.

   If SPICR3 ⊕ SPICR0 = 0, then in case of data arbitration on the data output line, the data input to the SPI data register is always equal to the logic level imposed on the data input pin by the device which wins the data arbitration.

Table 4-2. Port B Status During SPI Operation

Port Name	Use	Input	Output	Comments
PB0	SPISS	Yes	No	Used as slave select input
PB0	Data	No	Yes	Used as "busy" signal or any digital output
PB1	SPICL	Yes	No	Clock slave
PB1	SPICL	No	Yes	Clock master
PB2	SPID	Yes	No	SPI data input SPICR0 = 0
PB2	SPID	No	Yes	SPI data output SPICR3 = 0
PB2	Data	Yes	Yes	Any digital signal SPICR3 = 1
PB3	SPID	Yes	No	SPI data input SPICR0 = 1
PB3	SPID	No	Yes	SPI data output SPICR3 = 1
PB3	Data	Yes	Yes	Any digital signal SPICR3 = 0

4) When full duplex operation is required, then SPICR3 ⊕ SPICR0 = 1. In this mode, 16 bits of information may be transferred with eight clock pulses between at least two devices with transmit capability. In this mode both PB2 and PB3 are used for SPI data transfer. Moreover, the same shift register is used for data out and data in. Thus, the byte transmitted is replaced by the byte received, removing the need for separate status bits for XMIT EMPTY and REC FULL. A single status bit, SPICR7, is used to signify that the input/output operation is complete.

5) The SPI clock is always provided on port B1. In the clock slave mode, the port B1 DDR is clear (i.e., input mode). In the clock master mode, the port B1 DDR is set and hence the MCU imposes the clock level on pin PB1 until there is clock arbitration on the clock line or until the MCU loses clock mastership when the slave select input PB0 goes low.

6) Clock pulse generation in the case of clock mastership is accomplished via the data register toggle facility provided on port B1. According to the status of MR0, the overflow pulse of either timer A or B is used as a toggle clock source during the active transaction time. Hence, the port B1 data register changes state every time there is a timer overflow. Clock frequency generated by this method is therefore half the overflow frequency of the selected timer. There is no fixed baud rate generation. The clock frequency is dependent on the prescaler clock source option, prescaler divide ratio, and timer divide ratio as well as the port C status in case of external clocking for the timer. Toggling of the port B1 data register is automatically allowed during the active transaction mode.

7) For correct transfer of data between devices connected to the SPI, all devices must have their output data strobe and input data strobe on the same clock edges.

8) For proper transmission, the first clock edge during the active transaction mode must be the output data strobe. When this occurs, the MSBs of the data registers of all transmitters are copied on to the data output pins (e.g., this is valid for devices with such output capability) and the MCU copies the MSB of its SPI on to the port B2 or B3 data register, according to SPICR3 status.

   On the opposite clock edge, all receivers internally generate the data input strobe and shift by one location the contents of the SPI data register. Data for the receivers is assumed to be stable on this clock edge. Hence, error-free master-slave type serial data transfer is accomplished. It is therefore important that before a serial data transfer starts, the master clock level has to be initialized under program control so as to create an output data strobe on the initial SPI clock edge.

   NOTE

   If the initial clock edge is the input data strobe, the MSB of all receivers are lost, and transmitted MCU data will have a framing error. However, if a peripheral transmitter device (without the selective data output and input strobe feature) is transmitting data to the MCU, then, the first clock edge should generate the data input strobe for the MCU.

9) The data direction registers of port B are always accessible during SPI operation. This is also true for data control registers of port B which control open-drain enables and the port B output toggle enables (DCR7 through DCR4). However, during SPI active transaction mode, the following data registers are not write accessible under program control:
   a) PB1 data register;
   b) PB2 data register if SPICR 3=0, and
   c) PB3 data register if SPICR 3=1.

   This allows write instructions to port B lines not used for SPI operation during the active transaction mode without affecting the contents of data registers used for SPI.

10) The toggle enable of the port B1 data register is asserted during the active transaction mode by the SPI logic. This starts the generation of SPI clock pulses if the MCU has the clock mastership. If the MCU is in the clock slave mode (DDR B1=0), then an external device provides the clock pulses.

11) Port B lines not used for SPI can be used for other digital functions, e.g.; a) in half-duplex or one-wire operation the unselected SPI data port may be used as I/O, and b) port B0 may always be used as digital output in the modes where SPI operates without slave select input.

4.7 START BIT OPERATION

In all operating modes of the SPI, it is implied that all data transmissions are sensitive to the clock edges. Depending on the state of SPICR5, data changes either as the result of the rising or falling edge of the clock SPICL.

The clock level prior to the transition that causes data on the serial data line to be changed is called the "idle" level. It is assumed that data must be stable just prior to and during the idle level during transmission.

Optional creation of an exception to this rule may be interpreted as additional information such as to 1) signal the beginning of a transmission; 2) to separate address and data fields and/or 3) to synchronize transmitter and receivers.

Negative transition of data input while the clock line (SPICL) is in its "idle" level is being defined as an exceptional condition on the MC6805S2 SPI. This condition causes SPICR2 and SPICR5 to be set and is defined as the start condition.

The rising edge of SPICR2 causes the divide-by-eight counter, SPICR7 and SPICR6 to be cleared.

Refer to Figure 4-5 for clock idle level definition, to Figure 4-6 for the start bit definition, and to Figure 4-7 for stop bit definition.

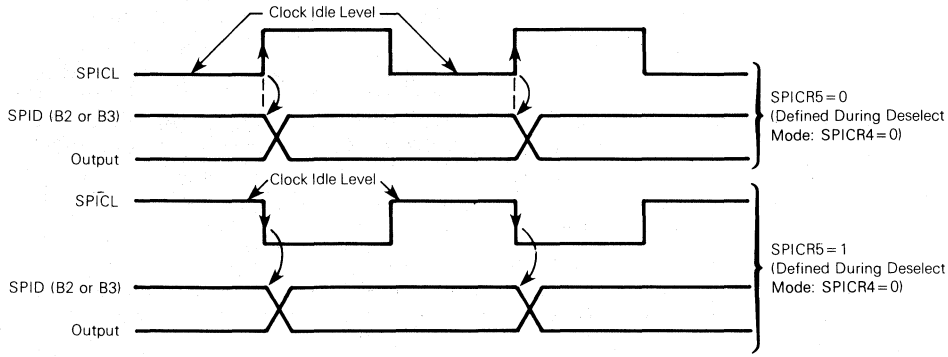


Figure 4-5. SPI Clock Idle Level Definition

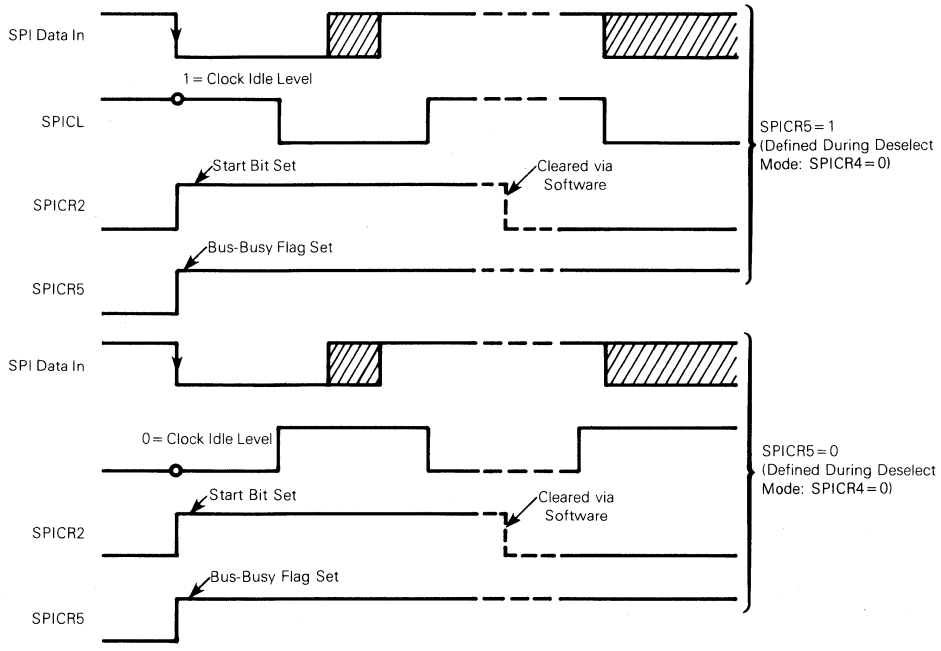


Figure 4-6. SPI Start Bit Definition

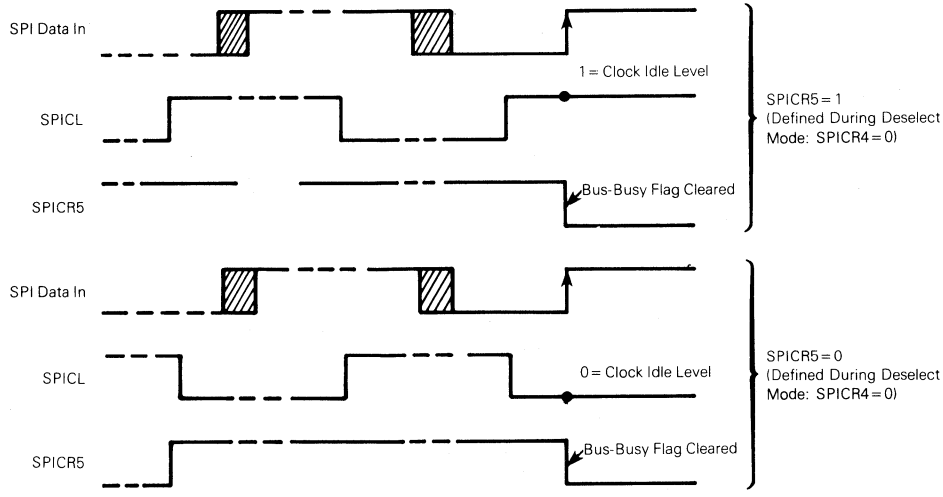


Figure 4-7. SPI Stop Bit Definition

4.8 ADDRESS AND DATA FIELD SEPARATION

In systems connected together on a serial bus without individual chip-selects for individual elements connected to this bus, serial transmission must convey not only data but the address of the receiver element to which data is sent. Since all transmissions are byte long, recognition of address from a data pattern is not possible unless the address can be distinguished from data by a start bit. In many standard accepted systems, an address field follows a start condition which is then followed by a number of data fields depending on the transaction relative to that address.

Detection of this start condition sets SPICR2, hence at the end of an 8-bit transmission it is possible to check if the received byte corresponds to address or data fields. It must be emphasized that a start condition does not occur normally during the transmission but it is provoked by the transmitter, prior to address field transmission.

Secondly, zero-to-one transition of SPICR2 caused by the start condition causes the divide-by-eight counter to be cleared, hence allows all receiver MCUs to be synched-up simultaneously.

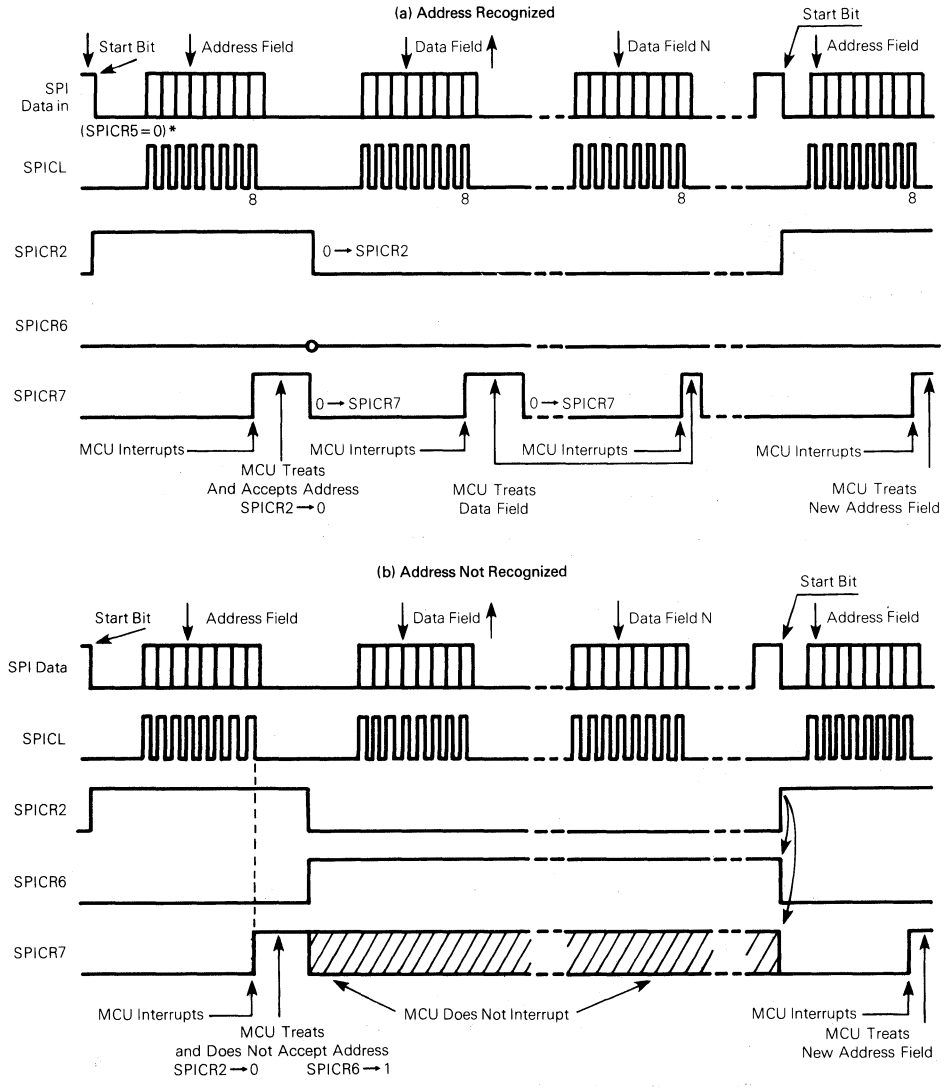
The third important consideration is the rate of occupation of the MCU in serving the information flow on the serial bus with respect to the background tasks.

In case of high-speed transmissions (up to 100K Baud) and heavy information flow not related to a given MCU on the serial data bus, it is possible, if no precaution is taken, that a non-selected receiver MCU has to analyze every field; data or address, to check for a particular address field that is of concern. This causes a very high interruption rate to service the SPI and leaves very little time for background tasks. In order to mask an undesirable data field transmission, that requires interrupt driven analysis, SPICR6 may be set and SPICR2 can be cleared after analyzing an invalid address field. Then, the MCU becomes immune to all SPI interrupt requests due to subsequent data fields. On the next start bit preceding a new message, SPICR2 is set which in turn causes SPICR7 and SPICR6 to be cleared. The MCU is then ready to service an incoming new address field via interrupt.

Refer to Figure 4-8 which illustrates the time for SPI address and data field separation (reception).

4.9 DATA FIELD ONLY OPERATION

In applications where: 1) only data patterns are transmitted or 2) the effect of the rising edge of SPICR2 having cleared SPICR6, SPICR7, and divide-by-eight counter needs to be inhibited, it is sufficient to set SPICR2 under program control before transmission. SPICR7 and SPICR6 are not cleared by the software controlled setting of SPICR2.

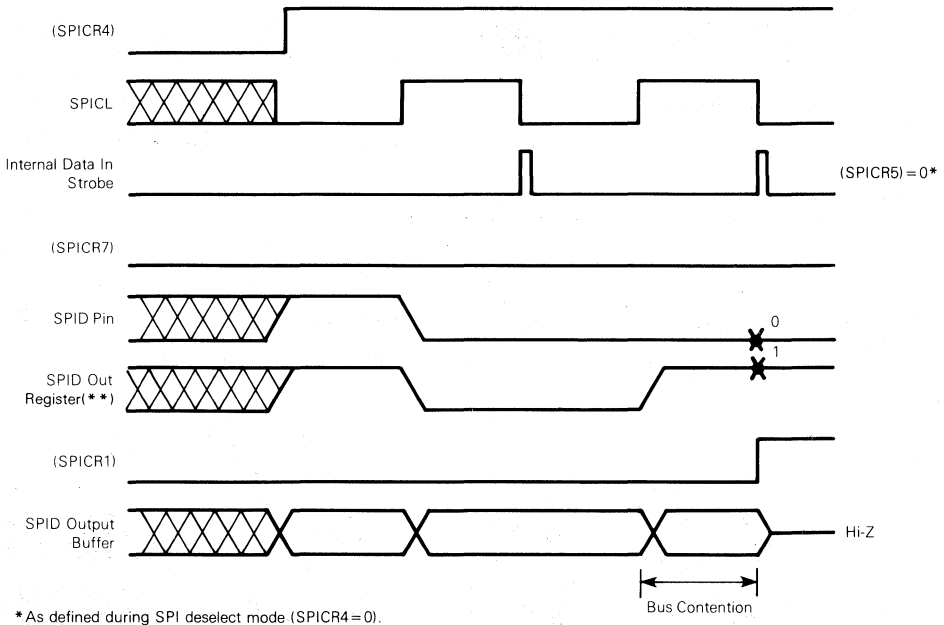


*As defined during SPI deselect mode (SPICR4=0).

Figure 4-8. SPI Address and Data Field Separation (Reception)

4.10 DATA ARBITRATION

Data arbitration occurs when two or more transmitters try to control a common data line. Refer to Figure 4-9 for data arbitration timing.



*As defined during SPI deselect mode (SPICR4 = 0).
**SPIDout = PB2 if $\overline{SPICR3}$•DDRB2 = 1
SPIDout = PB3 if SPICR3•DDRB3 = 1

Figure 4-9. SPI Data Arbitration Timing Diagram

The MCU handles the data arbitration in the following ways:

Starting Conditions

1) The MCU has data mastership, i.e., port B2 or B3 are used for SPI data transfer and have their data direction registers in the output mode.
2) SPICR3 is preset properly to output the SPI data on the selected data output port (PB2 or PB3)
3) The SPI is in the active transaction mode.

Arbitration Criterion

The SPI data output line logic level on the pin is compared with contents of the data register of that line during the data input strobe. If the data register content is one while the SPI data output line logic level is zero then it is decided that an external device(s) is (are) trying to control the data line.

Action

When the arbitration criteria are met, the mode fault flag (SPICR1) is set, the MCU loses data mastership and the SPI data output line DDR is cleared putting the line in the input mode.

NOTE

Complementary type of arbitration (i.e., output data line equals one; port data register equals zero) is not implemented and should not occur in the system as this will cause excessive dissipation on the port and may result in a catastrophic failure of the circuit.

4.11 CLOCK ARBITRATION

Clock arbitration is done in two ways: 1) via the slave select input line and 2) via the serial peripheral interface clock line. Both types of arbitration may be used simultaneously in an application.

4.11.1 Clock Arbitration via Slave Select Input Line

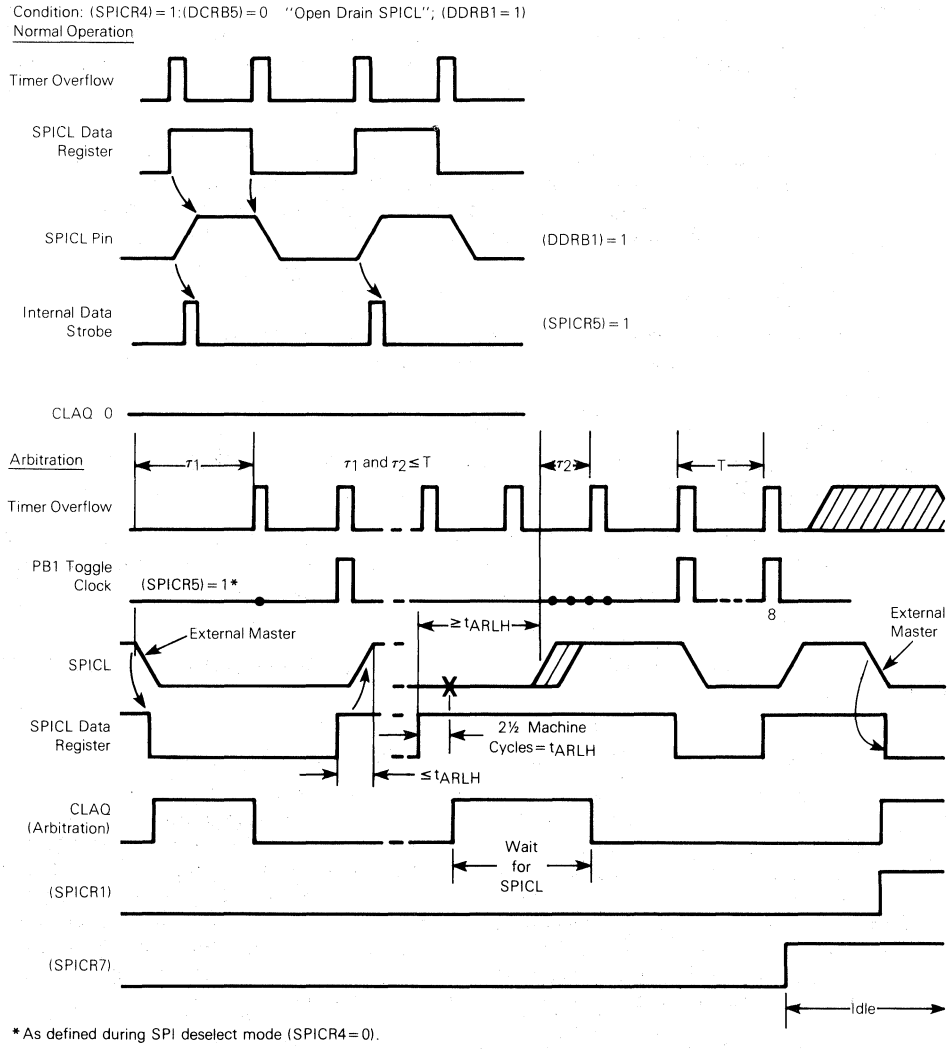
During serial peripheral interface transactions, port B0 serves as the slave select input if port B0 is in the input mode (DDR B0 = 0).

When the MCU has clock mastership, PB0 should remain high. When an external device requests clock mastership this input is pulled low. The MCU loses clock mastership and switches to slave type operation, the clock line data direction register bit is cleared, (DDR B1 = 0), and the mode fault flag is set.

This clock arbitration may happen during active or idle transaction modes (see **4.12 SLAVE SELECT INPUT OPERATION**).

4.11.2 Clock Arbitration via Serial Peripheral Interface Clock Line

This type of arbitration is enabled only when the MCU operates as clock master while the clock line output buffer works in the open-drain mode (DCR B5 = 0). Unlike the clock arbitration described previously, the MCU does not lose clock mastership. The clock output data register status is monitored under control of the clock arbitration flip-flop to guarantee minimum clock high and clock low times on the clock line, in case two or more clock masters are trying to control the clock line simultaneously. Each clock master may be assumed to be asynchronous with respect to the other(s) and to run with different clock frequencies. When set, the clock arbitration flip-flop (CLAQ) blocks the toggle enable of port B1 effectively inhibiting the port data register from changing state by toggling during the toggle pulse. Refer to Figure 4-10 for timing.



*As defined during SPI deselect mode (SPICR4 = 0).

Figure 4-10. Clock Arbitration via Clock Line Timing

CLAQ status is modified under the following conditions:

a) CLAQ is cleared when:
   1) SPICR 4=0 or DDR B1=1
   2) Toggle pulse is generated for port B1.

b) CLAQ is set when:
   1) A negative edge is detected on the SPI clock input if the port B1 data register is high. Simultaneously, the port B1 data register is cleared. If this occurs in the idle mode, while the MCU is not ready for serial transmission, the mode fault flag (SPICR 1) is set as well. In this way, the MCU will keep the clock line low, effectively blocking all clock pulses on the clock line, and detecting that the clock line was driven low during the idle mode. If the MCU was set up as a transmitter, the clock edge occurring during the idle mode cannot generate an internal data output strobe. Hence, during subsequent serial transmission receivers it would "miss" the MSB of the data transmitted from the MCU. Protocols can be set up to avoid, or recover from, this type of framing error.
   2) If the SPI clock line is still low 2½ machine cycles after the port B1 data register is set, the CLAQ set command will remain active, as long as the SPI clock line remains low. The clock arbitration operation is explained in more detail in **4.13.3 Two-Wire Half Duplex Mode with Clock Arbitration.**

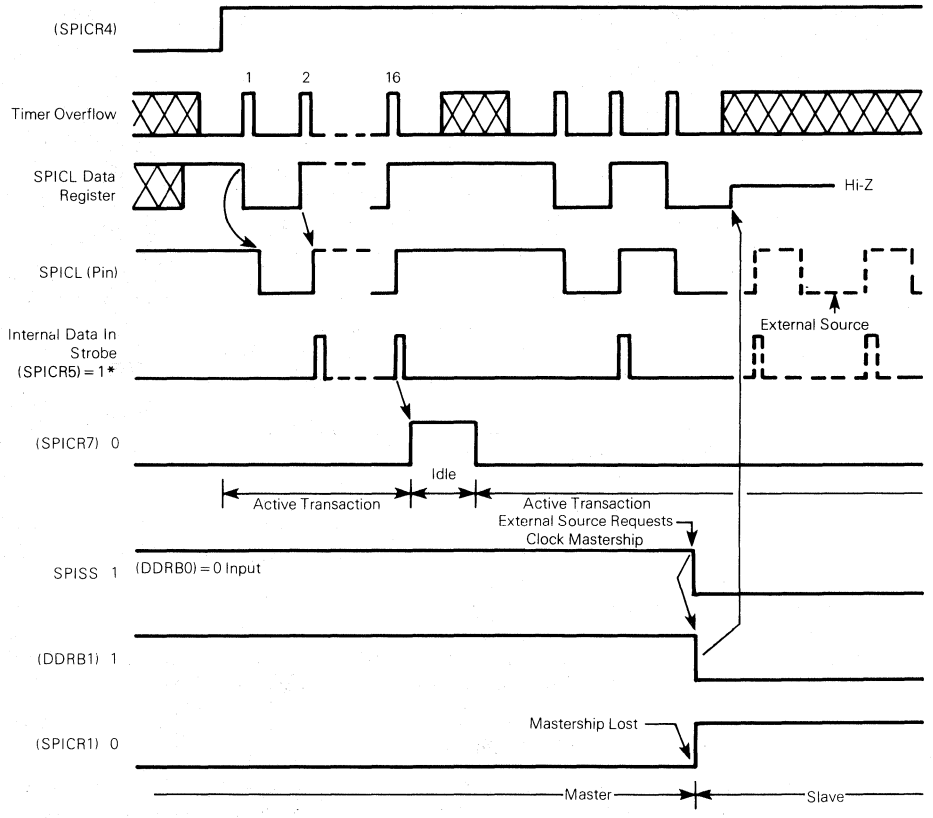
4.12 SLAVE SELECT INPUT OPERATION

Slave select information is supplied to the MCU via port B0 by an external device. If port B0 is in the output mode then slave select actions are inhibited. If the slave select feature is not used in an application, port B0 should be used in the output mode.

Slave select input generates various actions depending on whether the SPI is operating in the clock master mode or clock slave mode. These are outlined in the following paragraphs.

4.12.1 Slave Select Input Actions During Master Mode

In this mode, the slave select input is monitored to assure that it stays false (high). If slave select becomes true (low), the device immediately exits the master mode and becomes a slave (DDR B1=0). The significance of this is that a collision has occurred; that is, two devices have both become or are willing to become masters. This is normally the result of a software error, although some systems may allow the default master to "knock all other masters off the bus" if an erroneous bus state is detected. This is a castastrophic event and it is the responsibility of the default master to completely "clean up" the system. Moreover, the mode fault flag is set to signal to the MCU that clock mastership is lost. These actions can take place during either active or idle transaction modes. Refer to Figure 4-11.



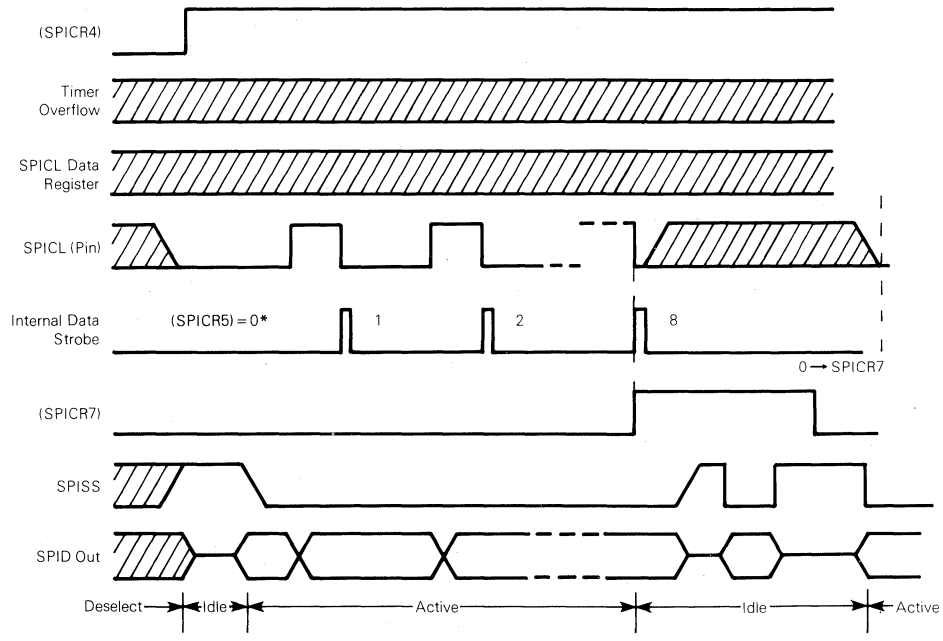
*As defined during SPI deselect mode (SPICR4 = 0).

Figure 4-11. Clock Arbitration via Slave Select Input — Master Mode

4.12.2 Slave Select Input Actions During Slave Mode

The slave select (SS) input is generated by the current clock master (parallel port may be used) and used to enable one of several possible slaves to accept and/or return data. The SS signal must be low prior to occurrence of serial clock pulses and must not become high until the eighth (last) serial clock cycle. A high level on SS forces serial data output to the high-impedance state without affecting the data direction register status relative to the data output. Also, when SS is high the serial clock input pulses (if any) are inhibited from generating internal data output and input strobe pulses, and also the eight-bit counter is cleared.

The significance of this is that the slave select acts as a chip-enable line and the MCU receives and/or is allowed to transmit back information only when SS is pulled low by the current clock master. Individual lines must be used from the master for each slave select input. A single line is sufficient in the case of daisy chain or cascade connection of multiple slaves. Refer to Figure 4-12.



*As defined during SPI deselect mode (SPICR4 = 0).

Figure 4-12. Clock Arbitration via Slave Select Input — Slave Mode

4.13 SPI OPERATING MODES

A brief description of the serial peripheral interface (SPI) operating modes is contained in the following paragraphs.

4.13.1 One-Wire — Autoclocked Mode

In this mode, various circuits are connected to each other via a single wire one which data transfer takes place. The clock is implicit during transmission and each circuit is its own clock master. The MCU should be initialized as clock master and port B1 is not connected externally. In order to achieve the precise timing required for this transmission it may be useful to start the active transaction mode with an interrupt. Hence, the data input/output line can be connected on the MCU to the $\overline{INT2}$ line.

With the assistance of software to generate the start bit and stop bits, and swap the order of bits in the data, NRZ-type serial transmission compatible with MC6801 can be achieved in this mode. (See Figures 4-13 and 4-14.) Unused SPI data port B2 or B3 may be used as a normal input/output. Port B0 may be used only as an output.

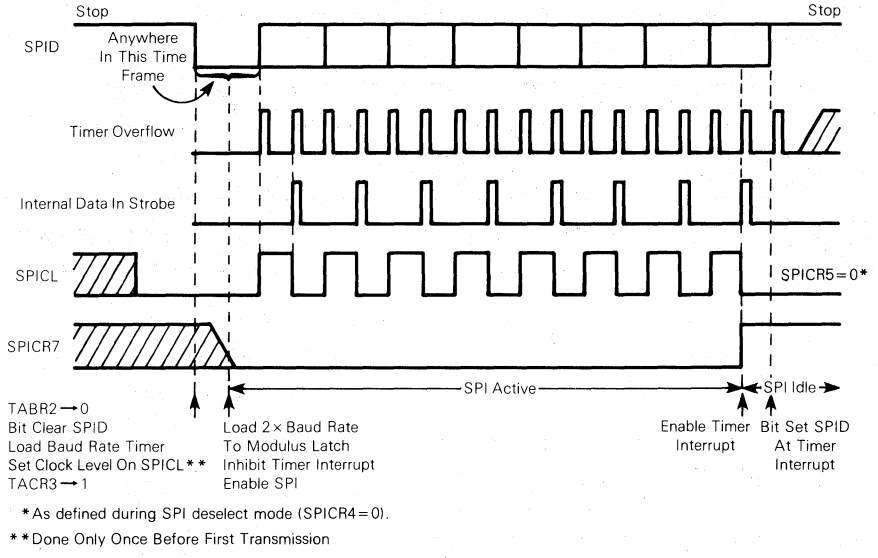


Figure 4-13. SPI — NRZ Operation (Transmit) Timing

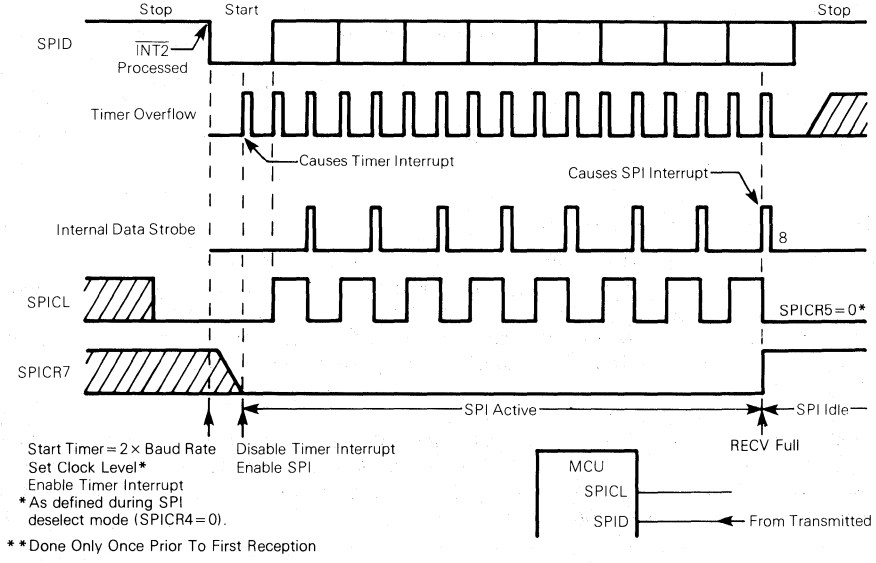


Figure 4-14. SPI NRZ Operation Timing (Receive)

4.13.2 Two-Wire Half-Duplex Mode

In this mode, the data and clock lines are connected between various circuits in the system. Data and clock mastership should be monitored via protocol included in the data patterns transmitted between circuits. Moreover, data arbitration is possible on the MCU data line. Any transmitter can "knock out" all others by transmitting all zeros.

4.13.3 Two-Wire Half-Duplex Mode with Clock Arbitration

In this mode, the MCU is assumed to operate as a clock master with an open-drain SPI clock output buffer. Clock and data arbitration is accomplished as explained in **4.11 CLOCK ARBITRATION.** More than one clock master (and transmitter) is allowed at the same time in this mode.

An interesting protocol occurs when the clock lines of all masters operate with open-drain outputs. If no master other than the MCU is operating on the clock line, then the clock arbitration flip-flop (CLAQ) is never set and every toggle pulse creates an edge on the SPI clock line (SPICL). This is the normal mode of operation.

However, if an external master pulls the SPI clock line low, the MCU sets CLAQ to inhibit the next timer overflow from generating a toggle pulse on the SPI clock port. The SPI clock port data register is also cleared. At the next timer overflow, CLAQ is reset and the SPI clock port is allowed to toggle during future timer overflows. In the meantime, other master clock outputs may go high. However, the SPI clock line is held low by the MCU until a low-to-high transition occurs on its SPICL data register line. (In wire-or configuration, any master with a low output imposes a low clock line on the total system.)

This mechanism guarantees that in case of clock arbitration (a process which is asynchronous to the timer overflows) the SPI clock low time is not shorter than one toggle period. Hence, narrow negative glitches are avoided on the clock line. Some devices in the system may be operated totally under software control by using polling techniques. Polling is generally much slower than hardwired logic. Potential appearance of narrow glitches could cause castastrophic system faults, as some devices in the system might respond to them and some might not.

The clock arbitration flip-flop is also set when the SPICL data register toggles high while an external master keeps the SPI clock line low after two and one-half machine cycles. CLAQ remains set until the SPICL line returns to a high state. At the next timer overflow, CLAQ is reset. Future timer overflows will be allowed to toggle the SPICL data register to the low state.

This mechanism guarantees that in case of a clock arbitration situation, the SPI clock high time is not shorter than one toggle period. This avoids narrow positive glitches. The same comments are applicable to positive glitches with regard to system performance.

In such a system, the longest clock low time is imposed by the clock master with the longest clock low time. The shortest high time is determined by the device with the shortest high clock time.

4.13.4 Three-Wire Half-Duplex Mode with Slave Select Input

This mode is similar to the two-wire half-duplex mode except that the slave select input provides the possibility of using the MCU as a peripheral circuit in a system (or in systems) where clock mastership may be passed through the slave select line.

A typical method of doing this is to wire the slave select lines together. The current master puts its slave select line (SPISS) in the output mode prior to a serial transmission and pulls the SPISS line low signifying that the system is busy. In this way, the clock master will keep its mastership until the end of the transmission. Software protocol can be arranged such that slaves do not request mastership until their SPISS lines go high. At the end of a transmission, the current master pulls the SPISS line high and puts its SPISS port (PB0) in the input mode. A slave requesting clock mastership can now pull the SPISS line low, "knocking out" the current master. To avoid simultaneous mastership requests, time multiplexed protocols may be required.

4.13.5 Three-Wire Full-Duplex Mode

In this mode, the MCU can operate as a transmitter and receiver at the same time. Bus oriented or daisy chain type networks are feasible. Protocols included in the data stream are required to change the clock masters, number of transmitters in the system, or the direction of information flow in daisy chained systems with "collision." In this mode, it is possible for the MCU to shift out one byte while receiving another. This removes the need for XMIT EMPTY or REC FULL status bits. Refer to Figure 4-15.

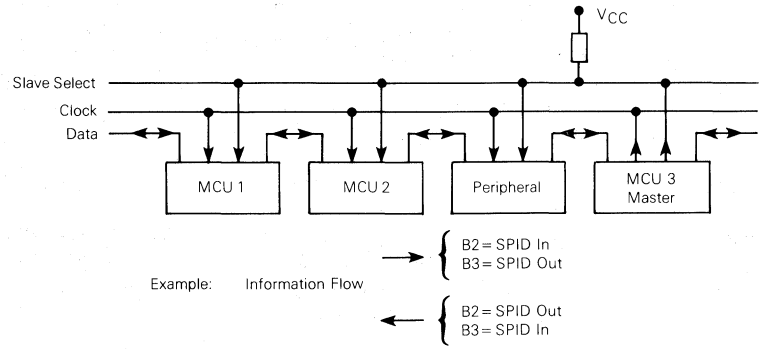


Figure 4-15. Daisy Chain/Cascade Organization

4.13.6 Three-Wire Full-Duplex Mode with Clock Arbitration

This mode is a mix of the three-wire full-duplex mode and the two-wire half-duplex mode with clock arbitration, where the SPI clock line operates in a wire-or fashion in the system. Simultaneous masters are allowed and clock arbitration is accomplished via the clock line.

4.13.7 Four-Wire Full-Duplex Mode with Slave-Select Input

This mode is similar to the three-wire full-duplex mode with regard to network and to the three-wire half-duplex mode with slave-select input in respect to clock arbitration and slave selection. Refer to Figure 4-16.

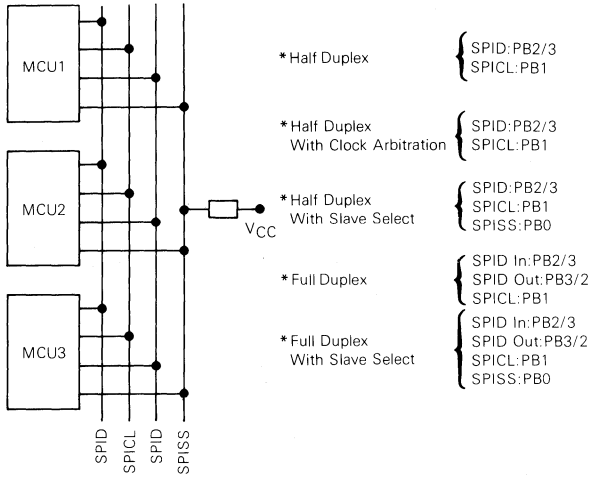


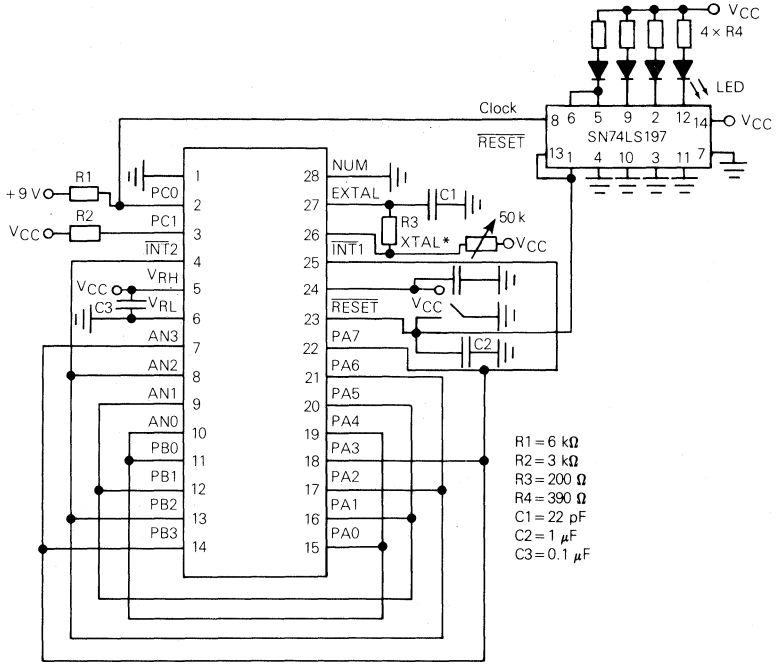
Figure 4-16. SPI Operation Bus Organization

SECTION 5
SELF-CHECK, RESETS, CLOCK GENERATOR OPTIONS, AND INTERRUPTS

This section describes the self-check capability, resets, clock generator options, and interrupts.

5.1 SELF-CHECK

The self-check capability of the MC6805S2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 5-1 and monitor the output of port C bit 0 for an oscillation of approximately 7 Hz. A 9-volt level on PC0, pin 2, detected as the device under test comes out of reset, energizes the ROM-based self-check feature. The self-check program exercises the CPU, RAM, ROM, A/D, timers, interrupts, I/O ports, and auxiliary counter.



*RC Oscillator Option Shown If Q0-Q2 LEDs Blinking = Device Passes Test
Q3 Blinking = Watchdog Reset Problem

Figure 5-1. Self-Check Connections

Several of the self-check subroutines can be called by a user program with a JSR or BSR instruction. They are the RAM, ROM, and 4-channel A/D tests. The timer routine may also be called if the timer input is the internal phase two clock.

5.1.1 RAM Self-Check Subroutine

The RAM self-check is called at location $F39 and returns with the Z bit clear if any error is detected; otherwise, the Z bit is set. The walking diagnostic pattern method is used.

The RAM test must be called with the stack pointer at $07F. When run, the test checks every RAM cell except for $07F and $07E which are assumed to contain the return address.

The A and X registers and all RAM locations except the top two are modified.

5.1.2 ROM Checksum Subroutine

The ROM self-check is called at location $F54 and returns with the Z bit cleared if any error was found; otherwise Z = 1, X = 0 on return, and A = 0 if the test passed. RAM locations $040-$043 are overwritten.

5.1.3 Analog-to-Digital Converter Self-Check

The A/D self-check is called at location $F6E and returns with the Z bit cleared if any error was found; otherwise Z = 1.

The A and X register contents are lost. The X register must be set to four before the call. On return, X = 8 and A/D channel 7 is selected. The A/D test uses the internal voltage references and confirms port connections.

5.1.4 Timer Self-Check Subroutine

The timer self-check is called at location $F99 and returns with the Z bit cleared if any error was found; otherwise Z = 1.

In order to work correctly as a user subroutine, the internal phase two clock must be the clock source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask is not set so the caller must protect from interrupts if necessary.

The A and X register contents are lost. The timer self-check routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler (1) is a power of two. If not, the timer probably is not counting correctly. The routine also detects if timer A is not running.

5.2 RESETS

The MCU can be reset four ways: by initial power up, by the external reset input ($\overline{\text{RESET}}$), by a forced reset generated by a timeout of the MCUs auxiliary or "watchdog" counter, and by an optional internal low voltage detect circuit. The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. A typical reset Schmitt trigger hysteresis curve is shown in Figure 5-2. The Schmitt trigger provides an internal reset voltage if it senses a logic zero on the $\overline{\text{RESET}}$ pin. Refer to the reset circuit in Figure 5-3 and to Figure 5-9, under **5.4 INTERRUPTS,** for the complete reset sequence.

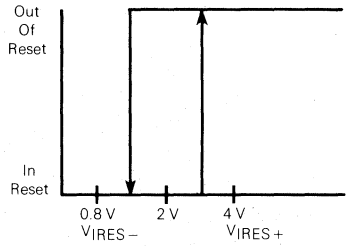


Figure 5-2. Typical Reset Schmitt Trigger Hysteresis

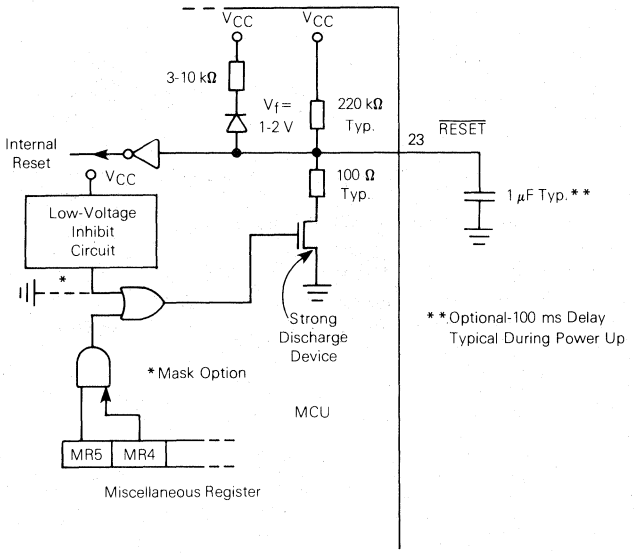


Figure 5-3. Reset Circuit

5.2.1 Power-On Reset (POR)

An internal reset is generated upon power up that allows the internal clock generator to stabilize. A delay of t_{RHL} milliseconds is required before allowing the $\overline{RESET}$ input to go high. Refer to the power and reset timing diagram of Figure 5-4. Connecting a capacitor to the $\overline{RESET}$ input (as illustrated in Figure 5-5) typically provides sufficient delay. During power up, the Schmitt trigger switches on (removes reset) when the $\overline{RESET}$ rises to V_{IRES+}.

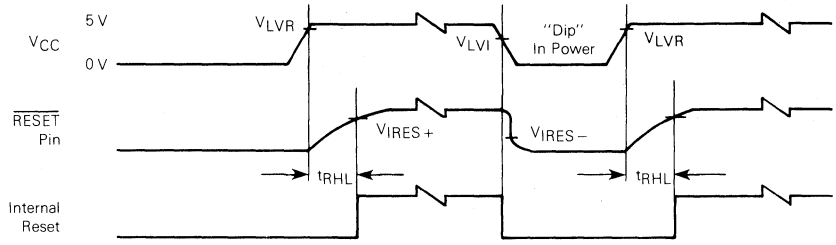


Figure 5-4. Power and Reset Timing

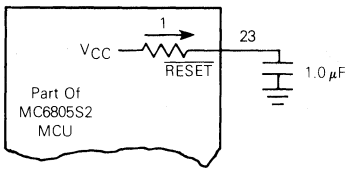


Figure 5-5. Power-Up Reset Delay Circuit

5.2.2 External Reset Input

The MCU will be reset if a logic zero is applied to the $\overline{RESET}$ input for a period longer than one machine cycle (t_{CYC}). Under this type of reset, the Schmitt trigger switches off at V_{IRES-} to provide an internal reset voltage.

5.2.3 Low Voltage Inhibit (LVI)

The optional low-voltage detection circuit causes a reset of the MCU if the power supply voltage falls below a certain level (V_{LVI}). The only requirement is that V_{CC} remains at or below the V_{LVI} threshold for one t_{CYC} minimum. In typical applications, the V_{CC} bus filter capacitor will eliminate negative-going voltage glitches of less than one t_{CYC}. The output from the low-voltage detector is connected directly to the internal reset circuitry. It also forces the $\overline{RESET}$ pin low via a strong discharge device through a resistor. The internal reset will be removed once the power supply voltage rises above a recovery level (V_{LVR}), at which time a normal power-on-reset occurs.

5.2.4 Forced Reset

If the auxiliary counter reset mask bit in the miscellaneous counter (MR4) is clear and the auxiliary counter status bit (MR5) is set, as a result of counter overflow, a switch to V_{SS} is turned on, pulling the $\overline{RESET}$ pin low. A consequent voltage drop below V_{IRES-} on $\overline{RESET}$ causes an MCU reset, which in turn sets MR4. Switching to V_{SS} when the $\overline{RESET}$ pin is turned off allows voltage to rise above V_{IRES+}, after which the MCU reset is released.

$\overline{RESET}$ pin voltage variations occurring as a result of forced reset may be amplified externally in order to provide a reset to other peripheral circuits in the system. The reset output from the MCU is not TTL compatible.

5.2.5 Reset Initialization

The minimum low time for all four modes of reset is one t_{CYC} + 250 nanoseconds (t_{CYC} = oscillator frequency divided by four). When reset is detected, the MCU initialization takes place. The following are the actions taken on the internal circuitry:

a) FF	Timer A Modulus Latch and Timer A	i) 40	Serial Peripheral Interface Control Register
b) FFFF	Timer B	j) 00	Port A Data Direction Register*
c) 7F	Prescaler 1	k) FC	Port C Data Direction Register*
d) 7FFF	Prescaler 2	l) F0	Port B Data Control Register*
e) 50	Timer A Control Register	m) 1	Interrupt (Mask Bit I in Condition Code Register)
f) 50	Timer B Control Register	n) 7F	Stack Pointer
g) 50	Miscellaneous Register	o) FFE	Program Counter
h) 07	A/D Status Control Register		

5.3 INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to control the internal clock generator with various stability/cost tradeoffs. A manufacturing mask option is used to select the crystal or resistor option. The oscillator frequency is internally divided by four to produce the internal system clocks.

The different connection methods are shown in Figure 5-6. The crystal specifications and suggested PC board layout are given in Figure 5-7. A resistor selection graph is shown in Figure 5-8.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitance, IC parameters, ambient temperatures, and supply oscillator start-up. Neither the crystal characteristics nor the load capacitances should exceed recommendations.

---

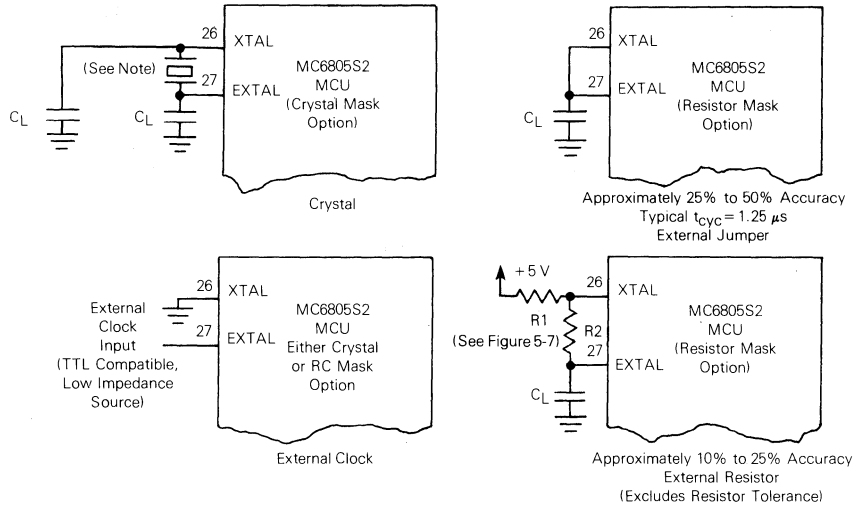
*Reads as $FF

When utilizing the on-board oscillator, the MCU should remain in a reset condition (reset pin voltage below V_{IRES+}) until the oscillator has stabilized at its operating frequency. Several factors are involved in calculating the external reset capacitor required to satisfy this condition: the oscillator start-up voltage, the oscillator stabilization time, the minimum V_{IRES+}, and the reset charging current specification.

Once V_{CC} minimum is reached, the external $\overline{RESET}$ capacitor will begin to charge at a rate dependent on the capacitor value. The charging current is supplied from V_{CC} through a large resistor, so it appears almost like a constant current source until the reset voltage rises above V_{IRES+}. Therefore, the $\overline{RESET}$ pin will charge at approximately:

$$(V_{IRES+}) \bullet C_{ext} = I_{RES} \bullet t_{RHL}$$

Assuming the external capacitor is initially discharged.



NOTE:
The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 50 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

Figure 5-6. Clock Generator Options

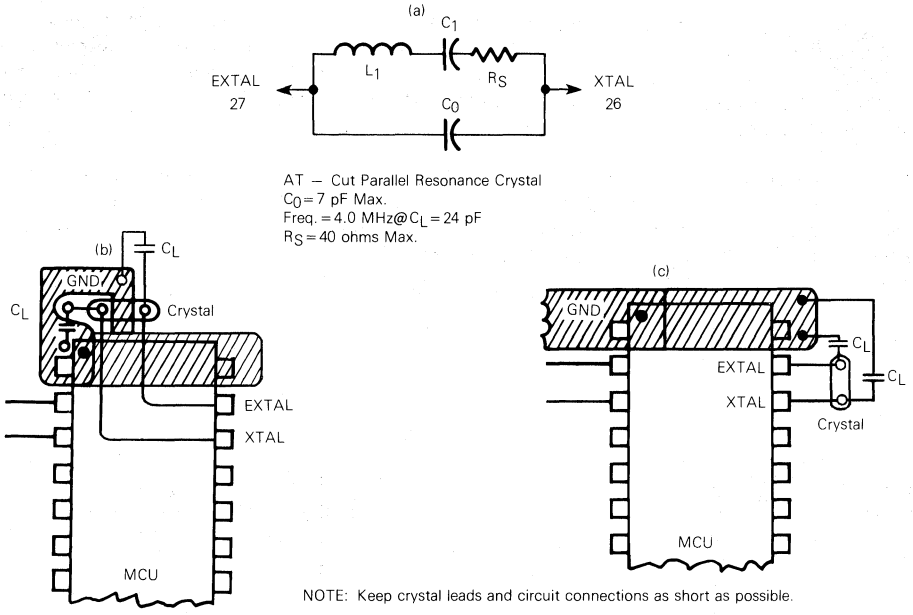


Figure 5-7. Crystal Motional Arm Parameters and Suggested PC Board Layout

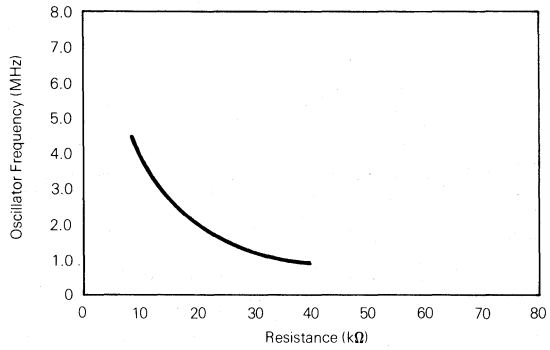


Figure 5-8. Typical Frequency Selection for Resistor Oscillator Option

5.4 INTERRUPTS

The MC6805S2 MCU can be interrupted seven different ways: at reset, through the external interrupt ($\overline{INT1}$) input pin, the internal timer (either A or B) interrupt request, the SPI interrupt request, the external port D bit 6 ($\overline{INT2}$) input pin, and a software interrupt instruction (SWI).

The reset interrupt has priority over all other interrupts and is not maskable. It is serviced immediately at its occurrence independent of the instruction being executed (see **5.2 RESETS**). All other interrupts are maskable and do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete. Pending $\overline{INT1}$, $\overline{INT2}$, timer A, timer B, or SPI interrupts are acknowledged by the MCU only if the I bit in the condition code register is clear.

When any interrupt (except reset) is acknowledged, processing is suspended following completion of the current instruction being executed, the present MCU state is pushed onto the stack, the interrupt bit (I bit) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{CYC} periods for completion. Note that interrupts which are masked are latched internally for later interrupt service once the mask bit(s) is (are) cleared. Refer to Figure 5-9 for a flowchart. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt.

Table 5-1 provides a listing of the interrupts, their priority, and the address of the vector which contains the starting address of the appropriate interrupt service routine. This priority applies to those interrupts pending when the CPU is ready to accept an interrupt. In addition, each of these interrupts, except $\overline{INT1}$, have a separate mask bit which must also be cleared, in addition to the I bit, for the MCU to acknowledge the interrupt. Specifically, the $\overline{INT2}$, timer A, timer B, and SPI interrupts each have their own independent mask bits contained in MR6, TACR6, TBCR6, and SPICR6, respectively.

NOTE

The timer A, timer B, $\overline{INT2}$, and SPI interrupts share the same vector address. The interrupt routine must determine the source of the interrupt by examining the interrupt request bits, namely TACR7, TBCR7, MR7, and SPICR7. These bits are not automatically cleared following interrupt servicing and must be cleared via software. The $\overline{INT1}$ interrupt has its own unique vector address. Therefore, the $\overline{INT1}$ interrupt request is cleared automatically when the $\overline{INT1}$ vector is serviced.

Table 5-1. Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RESET}$	1	$FFE and $FFF
SWI	2*	$FFC and $FFD
$\overline{INT1}$	3	$FFA and $FFB
TIMER/$\overline{INT2}$/SPI	4	$FF8 and $FF9

* Priority 2 applies when the I bit in the condition code register is set. When I = 0, SWI has a priority of four (like any other instruction). The priority of $\overline{INT1}$ thus becomes two and the TIMER/$\overline{INT2}$/SPI becomes three.

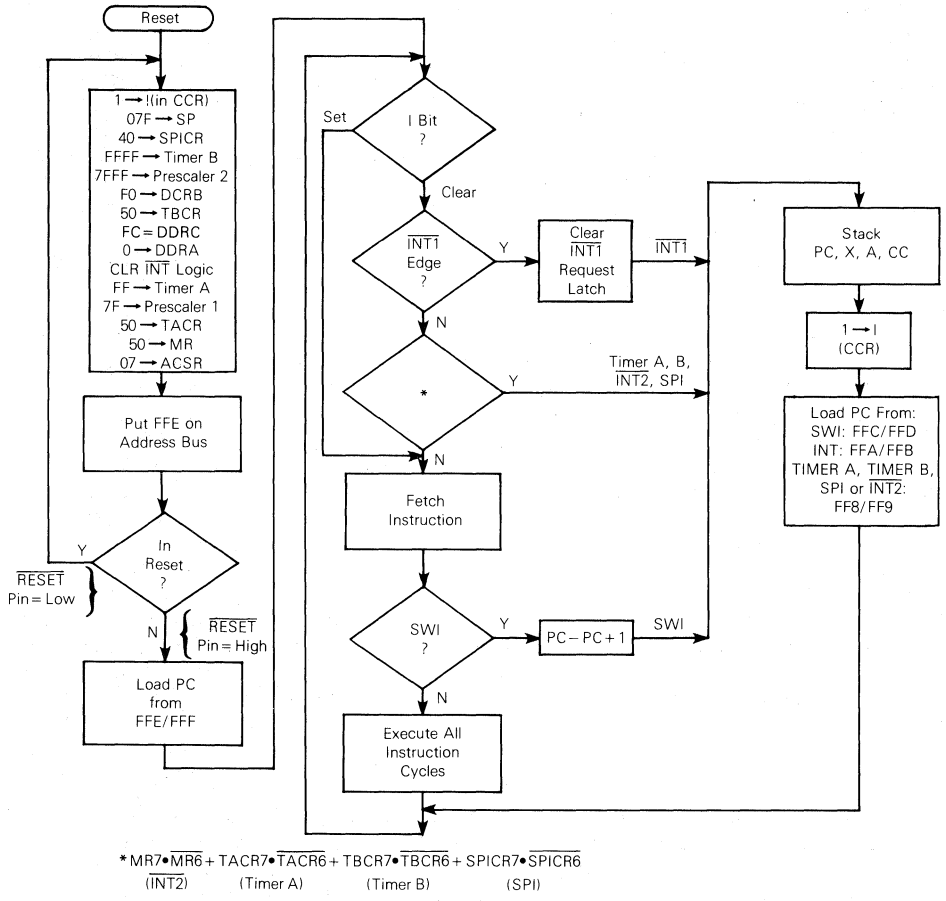


Figure 5-9. Reset and Interrupt Processing Flowchart

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register. SWIs are usually used as breakpoints for debugging or as system calls.

The external interrupts, $\overline{INT1}$ and $\overline{INT2}$, are latched and/or sensed on the falling edge of the input signal. Timer A and B interrupt request bits are set when these timers make transition to $00 and $0000, respectively.

A sinusoidal input signal (f_{INT} maximum) can be used to operate an external interrupt ($\overline{INT1}$), as shown in Figure 5-10, for use as a zero-crossing detector with hysteresis included. An interrupt request is generated for each negative-slope zero crossing of the ac signal. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.

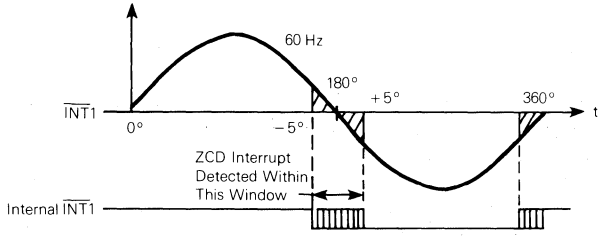
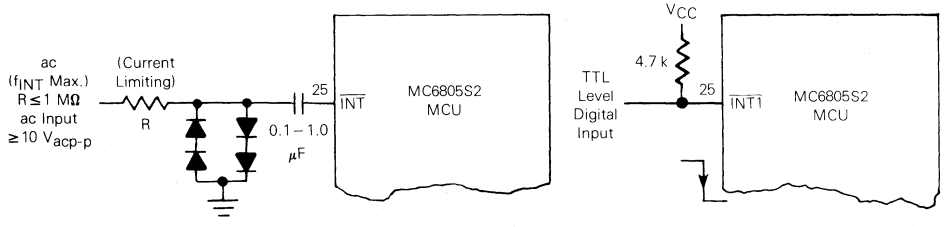


Figure 5-10. Interrupt Timing

For digital applications, $\overline{INT1}$ can be driven directly by a digital signal. The maximum frequency of a signal that can be recognized by the timer and $\overline{INT1}$ pin logic is dependent on the parameters labeled t_{WL} and t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin, in order to re-arm the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply t_{WL}, t_{WH}. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 5-11. For the $\overline{INT1}$ function, the maximum allowable frequency is also determined by the software response of the $\overline{INT1}$ service routine.



(a) Zero-Crossing Interrupt

(b) Digital-Signal Interrupt

Figure 5-11. Typical Interrupt Circuits ($\overline{INT1}$)

SECTION 6
INPUT/OUTPUT PORTS AND ANALOG-TO-DIGITAL CONVERTERS

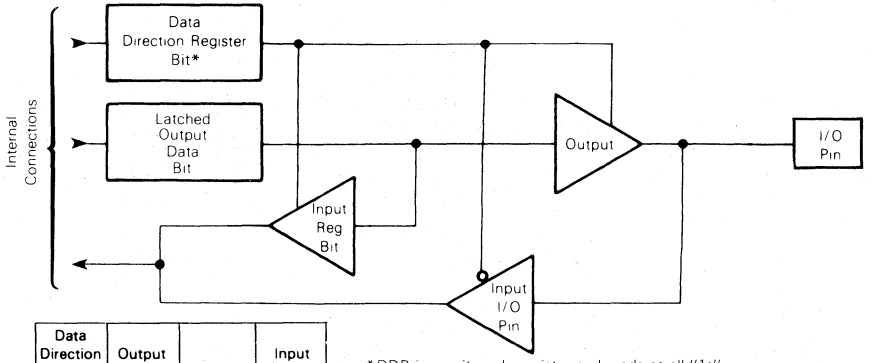
This section describes the input/output pins, the port data registers, the miscellaneous register, and the analog-to-digital converter.

6.1 INPUT/OUTPUT

There are 14 input or input/output pins. The $\overline{INT1}$ pin may also be polled with branch instructions to provide an additional input pin. All pins on ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding data direction registers (DDRs). The port I/O programming is accomplished by setting 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, placing the ports in the input mode. The port output registers are not initialized on reset and should be initialized by software before changing the DDRs from input to output. When programmed as outputs, all I/O pins read latched output data, regardless of the logic levels at the output pin due to output loading; refer to Figure 6-1 and Table 6-1.

CAUTION

The port data registers are not initialized during reset. The contents of these registers should be written to a known state for any port pins that are expected to become outputs. This will avoid any spurious transitions before initializing the corresponding DDR bits to the output mode.



Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

*DDR is a write-only register and reads as all "1s".
**Ports A (with CMOS drive disabled), B, and C are three-state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

Figure 6-1. Typical Port I/O Circuitry

Table 6-1. Digital Input/Output Ports

Name	Number	Input	Output	Input		Output		Comment
				TTL	Special	TTL	CMOS	
Port A	8	Yes	Yes	*		*	*(a)	a: If Pull-Up Option
Port B LED Drive	4	Yes	Yes	*(b)		*(a)		a: 10 mA Sink; Current Limited Source. PB1-PB3 can be programmed to open-drain configuration via PB DCR. b: Hi-Z Input
Port C	2(b)	Yes	Yes	*(a)		*		a: Hi-Z Input b: Shared with EXT Timer Inputs
Port D	7(a)	Yes	No	*				a: PD5 and PD4 Share a 15 kilohm resistor (typ)
$\overline{INT1}$	1	Yes	No		*			BIL/BIH Instruction

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs using a mask option. Ports B1, B2, and B3 can be software programmed to operate as open-drain outputs. Port B, C, and D lines are CMOS compatible as inputs. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1 milliampere on each pin (TTL output state). Port D lines are input only; thus, there is no corresponding DDR.

Port D provides the multiplexed analog inputs, reference voltages, and $\overline{INT2}$. All of these lines are shared with the port D digital inputs. PD0-PD3 may always be used as digital inputs and may also be used as analog inputs. The V_{RL} and V_{RH} lines (PD4 and PD5) are internally connected by the A/D resistor. Analog inputs may be prescaled to attain the V_{RL} and V_{RH} recommended input voltage range.

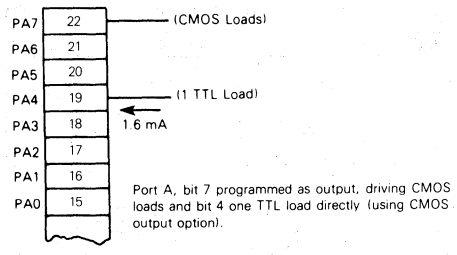
Figure 6-2 provides some examples of port connections. The address map in Figure 2-1 gives the addresses of data registers and DDRs.

CAUTION

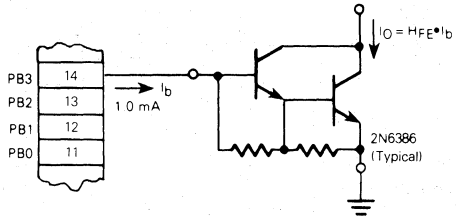
The corresponding DDRs for ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers reads as all ones. Since BSET and BCLR are read-modify-write functions, they cannot be used to set a DDR bit (all unaffected bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit may always be written. Therefore, any write to a port writes 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 (zero) and corresponds to the latched output data when the DDR is an output (one).

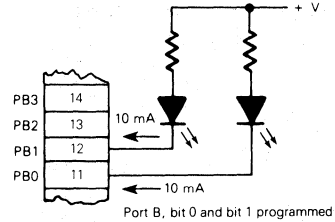
(a) Output Modes



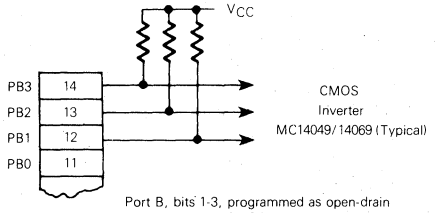
Port A, bit 7 programmed as output, driving CMOS loads and bit 4 one TTL load directly (using CMOS output option).



Port B, bit 3 programmed as output, driving Darlington-base directly.

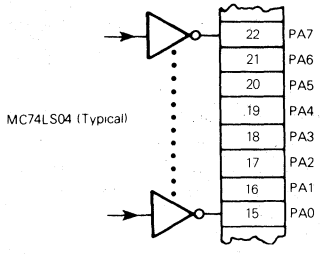


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

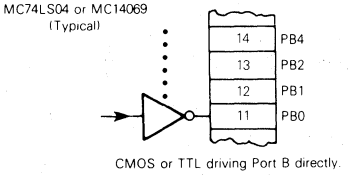


Port B, bits 1-3, programmed as open-drain outputs, driving CMOS loads using external pull-up resistors.

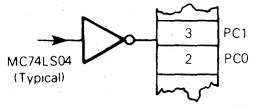
(b) Input Modes



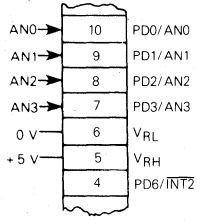
TTL driving Port A directly.



CMOS or TTL driving Port B directly.



CMOS and TTL driving Port C directly.



Port D used as 4-channel A/D input

Figure 6-2. Typical Port Connections

6.2 PORT B TOGGLE CAPABILITY

Port B1 and B0 data registers have toggle capability at the timer overflow times. Under the control of the timer output cross-couple bit in the miscellaneous register, (MR0), the overflow pulses from timer A and timer B are directed to port B1 and B0 data registers. (See Figures 6-3 and 6-4.)

An incoming toggle pulse on port B0 is allowed to toggle the data register if port B DCR bit 4 (DCR4) is cleared. At reset this bit is set. An incoming toggle pulse on port B1 is allowed to toggle the port B1 data register under the following conditions governed by control bits in SPI control register and SPI clock arbitration flip-flop status.

PB1 toggle enable = $(\overline{SPICR7} \bullet SPICR4 \bullet (\overline{PB0} + DDRB0) + SPICR2 \bullet \overline{SPICR4}) \bullet \overline{CLAQ}$

where: SPICR7 = SPI interrupt request flag bit,
SPICR4 = SPI transaction enable bit,
SPICR2 = port B 1 toggle enable bit, and
CLAQ = clock arbitration flip-flop output.

When PB1 toggle enable is asserted, MCU write to PB1 data register is inhibited. When SPI is not used, SPICR4 and CLAQ are reset. Therefore, SPICR2 can directly control the port B1 toggle capability. (See **4.6 SERIAL PERIPHERAL INTERFACE OPERATION**.)

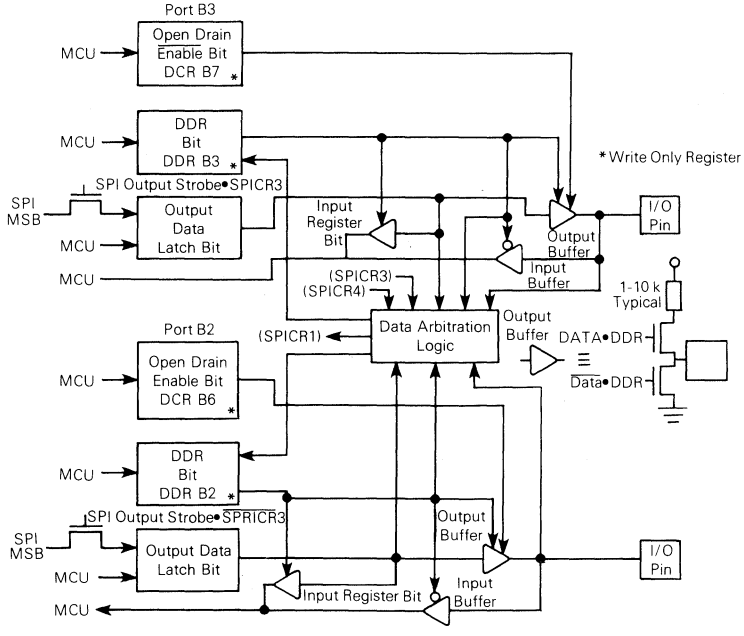
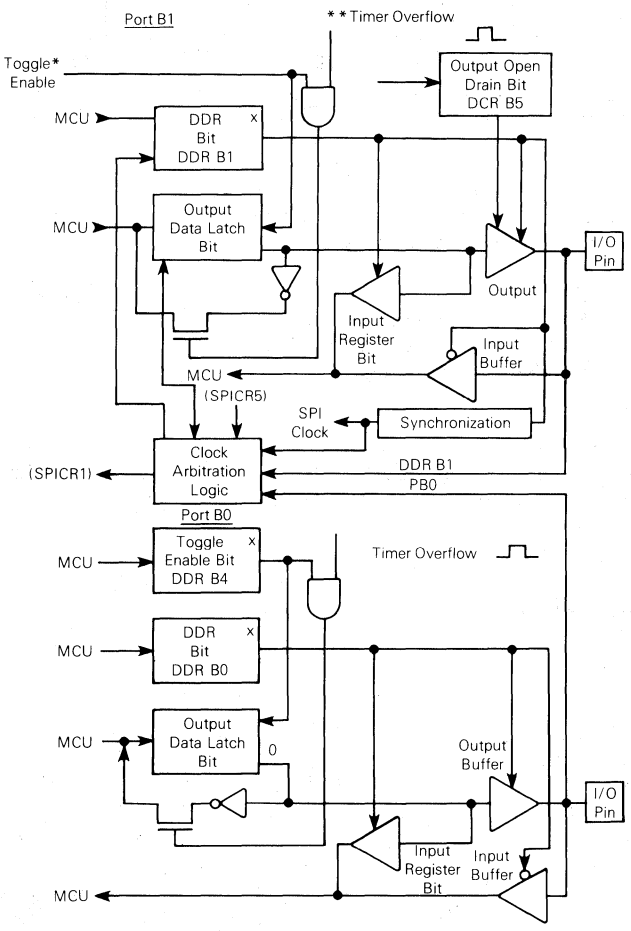


Figure 6-3. Port B Configuration (Sheet 1 of 2)



*Toggle Enable B1 = ($\overline{SPICR7}$•SPICR4•($\overline{PB0}$ + DDRB0))•SPICR2•$\overline{SPICR4}$)•$\overline{CLAQ}$
* *A, or B Depends on (MR0)
x Write Only Register

Figure 6-3. Port B Configuration (Sheet 2 of 2)

DCR7	DCR6	DCR5	DCR4	DDR3	DDR2	DDR1	DDR0	*
				Data Direction				

DCR7 – PB3 Output Buffer Open Drain Enable
DCR6 – PB2 Output Buffer Open Drain Enable
DCR5 – PB1 Output Buffer Open Drain Enable
DCR4 – PB0 Toggle Enable
DDR3 – PB3 Output Mode
DDR2 – PB2 Output Mode
DDR1 – PB1 Output Mode
DDR0 – PB0 Output Mode

*Write Only Register: All Bits Read as "1"

Figure 6-4. Port B Data Control Register

Port toggle capability allows action on port B0 or B1 or both as a result of timer overflows. This speeds up timer overflow to port service, compared to the normal program controlled method, and is very useful in critical real-time related applications.

Toggle capability on port B1 is fundamental for SPI operation in the clock master mode, where the clock pulses are generated by the MCU using this feature as controlled by one of the two available timers.

A write to port B0 or B1 data registers is inhibited while the individual port toggle enable is asserted. This allows a write to other port B data registers without disturbing the toggle feature of the selected port line.

6.3 PORT B DATA CONTROL REGISTER

The port B data control register consists of four status bits (DCR7 through DCR4) and four data direction bits (DCR3 through DCR0). DCR7, DCR6, and DCR5 are respectively port B3, B2, and B1 open-drain output control bits. These bits are set at reset or under program control and cleared under program control. When clear, the port output buffers operate in the open-drain mode, if the port lines are in the output mode. When set, the port output buffers operate in the push-pull mode.

DCR4 is a toggle enable control bit for port B0. This bit is set at reset or under program control and cleared under program control. When cleared, the timer overflow pulse causes the data register on port B0 to toggle.

When PB0 toggle enable is asserted by clearing DCR4, MCU write to the PB0 data register is inhibited.

DCR3, DCR2, DCR1, and DCR0 are respectively the port B3, B2, B1, and B0 data direction registers.

6.4 PORT A AND C DATA DIRECTION REGISTERS

Port A has an 8-bit and port C has a 2-bit wide data direction register. All bits are cleared at reset to the input mode. These registers are write only; they read as $FF.

6.5 MISCELLANEOUS REGISTER

The miscellaneous register (shown below), at memory location $0A, contains control and status information related to $\overline{INT2}$, auxiliary counter, prescaler 1 and 2, and timer overflow.

MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0	$0A

MR7 $\overline{INT2}$ Interrupt Request Bit
This bit is set when a negative edge is detected on $\overline{INT2}$ pin. If not masked by $\overline{INT2}$ interrupt mask bit, (MR6) it causes an interrupt request to the MCU. If the I bit in the condition code register is clear, the MCU will acknowledge interrupt by executing the interrupt procedure. The interrupt vector is fetched from address $FF8 and $FF9. This bit is cleared under program control or at reset.

MR6 $\overline{INT2}$ Interrupt Request Mask
This bit is set at reset or under program control. When set, it inhibits the $\overline{INT2}$ interrupt request from being acknowledged by the MCU. MR6 is cleared under program control.

MR5 Auxiliary Counter Status/Preset Bit
This bit is set whenever the auxiliary counter overflows. If it is not masked by the auxiliary counter reset mask MR4, it will drive a switch to V_{SS} on the $\overline{RESET}$ pin causing an MCU reset. This bit is cleared at reset or under program control. MR5 may be used as an auxiliary counter preset bit. If MR5 is clear it is possible to preset the auxiliary counter by writing a logic one to MR5. If MR5 is set (i.e., the auxiliary counter has already overflowed at least once) a logic zero written to MR5 clears the MR5 bit and presets the auxiliary counter. MR5 is cleared at reset. Refer to Figure 3-7 for auxiliary counter timing information.

MR4 Auxiliary Counter Reset Mask Bit
This bit is set at reset or under program control. When set, it inhibits activation of the reset switch controlled by MR5 on the $\overline{RESET}$ pin. MR4 is cleared under program control.

MR3 Prescaler 1 Clear Bit
This bit is used to preset the contents of prescaler 1 to $7F. This bit reads as a zero. In order to preset prescaler 1, a logic one must be written into MR3.

MR2 Prescaler 2 Clear Bit
This bit is used to preset the contents of prescaler 2 to $7FFF. This bit reads as a zero. In order to preset prescaler 2, a logic one must be written into MR2.

MR1 Prescaler Cross-Couple Bit
This bit controls the outputs of prescalers 1 and 2 and directs them to either timer A or timer B clock inputs. This bit is cleared at reset or under program control and set under program control. When MR1 is clear the output of prescaler 1 is used as a clock input of timer A and the output of prescaler 2 is used as clock input for timer B. When MR1 is set, outputs of the prescalers are cross-coupled. Thus, prescaler 1 feeds the timer B clock input and prescaler 2 feeds the timer A clock input.

To avoid truncation errors at the time of cross coupling, both prescalers may be preset by writing a one to MR3 and MR2 simultaneously.

MR0 Port B Toggle Cross-Couple Bit
This bit controls the overflow pulses of timers A and B and directs them to either port B1 or B0. This bit is cleared at reset or under program control and set under program control. When MR0 is clear, the overflow output pulse of timer A is used as a port B1 data register toggle clock source. Similarly, the timer B overflow output pulse is directed to port B0 toggle clock input.

When MR0 is set, timer A overflow output is directed to port B0 and timer B output is directed toward port B1.

6.6 ANALOG-TO-DIGITAL CONVERTER (A/D)

The MC6805S2 microcomputers have an 8-bit analog-to-digital (A/D) converter implemented on the chip using a successive approximation technique. Up to four external analog inputs, via port D, are connected to the A/D through a multiplexer. Four internal analog channels may be selected for calibration purposes (V_{RH}-V_{RL}, V_{RH}-$V_{RL}/2$, V_{RH}-$V_{RL}/4$, and V_{RL}). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

A fifth external analog input (AN4) is available via mask option. When selected, it replaces the V_{RH} internal channel. Due to signal routing, the accuracy of this fifth channel may be slightly less than AN0-AN3. The fifth A/D channel could be used to conveniently monitor the standby RAM supply voltage, as an example.

The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2.

Whenever the ACR is written, the conversion in progress is aborted, the conversion complete flag (ACR bit 7) is cleared, and the selected input is sampled for five machine cycles and held internally. During these five cycles the analog input will appear approximately like a 25 picofarads (maximum) capacitor charging through a 2.6 kilohm resistor, typical (see Figure 6-5).

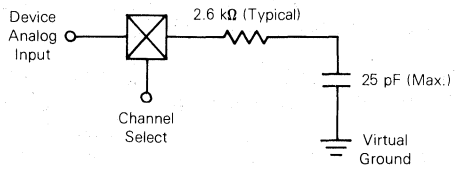


Figure 6-5. Effective Analog Input Impedance (During Sampling Only)

The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When the conversion is complete, the digitized sample of digital value is placed in the A/D result register (ARR), the conversion complete flag is set, the selected input is sampled again, and a new conversion is started.

The A/D is ratiometric. Two reference voltages (V_{RH} and V_{RL}) are supplied to the converter via port D pins. An input voltage equal to V_{RH} converts to $FF (full scale) and an input voltage equal to V_{RL} converts to $00. An input voltage greater than V_{RH} converts to $FF and no overflow indication is provided. Similarly, an input voltage less than V_{RL}, but greater than V_{SS} converts to $00. Maximum and minimum ratings must not be exceeded. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}. To maintain the full accuracy on the A/D, V_{RH} should be equal to or less than V_{DD}, V_{RL} should be equal to or greater than V_{SS} but less than the maximum specification and (V_{RH}-V_{RL}) should be equal to or greater than 4 volts.

The A/D has a built-in ½ LSB offset intended to reduce the magnitude of the quantizing error to ± ½ LSB, rather than +0, −1 LSB with no offset. This implies that, ignoring errors, the transition point from $00 to $01 occurs at ½ LSB above V_{RL}. Similarly, the transition from $FE to $FF occurs 1.5 LSB below V_{RH}, ideally. Refer to Figures 6-6 and 6-7.

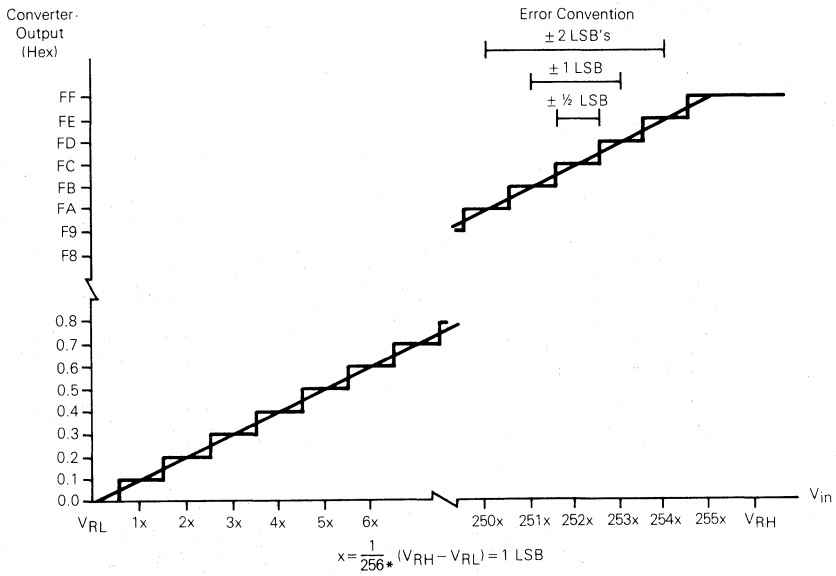


Figure 6-6. Ideal Converter Transfer Characteristic

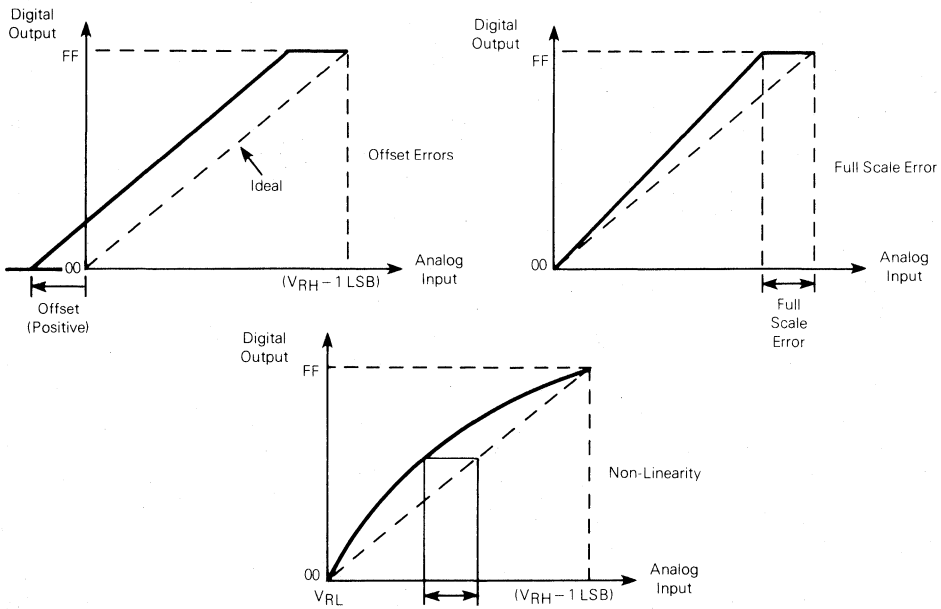


Figure 6-7. Types of Conversion Errors

On release of reset the A/D control register (ACR) is cleared therefore after reset channel zero will be selected and the conversion complete flag will be clear. Refer to Figure 6-8 and Table 6-2.

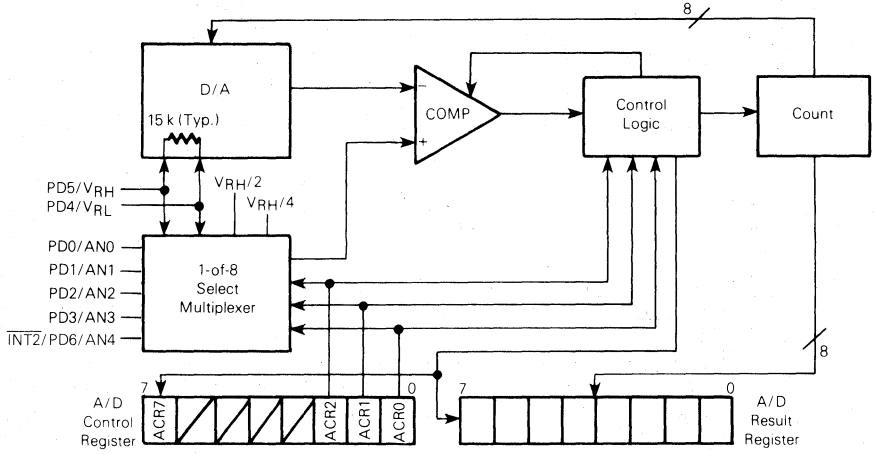


Figure 6-8. A/D Block Diagram

Table 6-2. A/D Input Mux Selection

A/D Control Register				A/D Output (Hex)		
ACR2	ACR1	ACR0	Input Selected	Min	Typ	Max
0	0	0	AN0			
0	0	1	AN1			
0	1	0	AN2			
0	1	1	AN3			
1	0	0	V_{RH}**	FE**	FF**	FF**
1	0	1	V_{RL}*	00	00	01
1	1	0	$V_{RH/4}$*	3F	40	41
1	1	1	$V_{RH/2}$*	7F	80	81

*Internal (calibration) levels

**AN4 may replace the V_{RH} calibration channel if selected via mask option.

SECTION 7
SOFTWARE AND INSTRUCTION SET

This section describes the software and instruction set for the MC6805S2.

7.1 SOFTWARE

The following paragraphs describe the software available to the user.

7.1.1 Bit Manipulation

The MC6805S2 MCU has the ability to set or clear any single RAM or input/output bit (except the data direction registers; see Caution under INPUT/OUTPUT) with a single instruction (BRSET, BCLR). Any bit in page zero, including ROM except the DDRs, can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit (C) equals the value of the bit referenced by BRSET or BRCLR. The capability of working with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

The coding example in Figure 7-1 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, LSB 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 a RAM location.

```
*       BRSET   2,PORTA,*   WAIT FOR READY

        BSET    1, PORTA    CLOCK NEXT BIT IN
        BRCLR   0, PORTA, NEXT  PICKUP BIT IN C-BIT
NEXT    BCLR    1, PORTA    RETURN CLOCK LINE HIGH
        ROR     RAMLOC      MOVE C-BIT INTO RAM
```

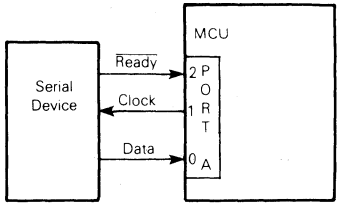


Figure 7-1. Bit Manipulation Example

7.1.2 Addressing Modes

The MC6805S2 MCU has ten addressing modes available for use by the programmer. They are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the *M6805 HMOS/M146805 CMOS Family Users Manual.*

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

7.1.2.1 IMMEDIATE. In the immediate addressing mode, the operand is contained in the byte immediately 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).

7.1.2.2 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 lowest 256 bytes in memory with a single two-byte instruction. This address area includes all on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

7.1.2.3 EXTENDED. In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

7.1.2.4 RELATIVE. The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if, and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from − 126 to + 129 from the opcode address. The programmer need not worry about calculating the correct offset if he uses the Motorola assembler, since it calculates the proper offset and checks to see if it is within the span of the branch.

7.1.2.5 INDEXED, NO OFFSET. In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

7.1.2.6 INDEXED, 8-BIT OFFSET. In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this two-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

7.1.2.7 INDEXED, 16-BIT OFFSET. In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended, the Motorola assembler determines the shortest from of indexed addressing.

7.1.2.8 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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction. See Caution under **6.1 INPUT/OUTPUT.**

7.1.2.9 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 and condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the condition code registers. See Caution under **6.1 INPUT/OUTPUT.**

7.1.2.10 INHERENT. In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

7.2 INSTRUCTION SET

The MC6805S2 MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 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 instruction within a given type are presented in individual tables.

7.2.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump-to-subroutine (JSR) instructions have no register operand. Refer to Table 7-1.

Table 7-1. Register/Memory Instructions

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DF	3	6
Store A in Memory	STA	–			B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–			BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–		BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–		BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

7.2.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify or test it contents, and write the modified value back to memory or to the register; see Caution under **6.1 INPUT/OUTPUT.** The test for negative or zero (TST) instruction is included in the read-modify-write instruction though it does not perform the write. Refer to Table 7-2.

Table 7-2. Read-Modify-Write Instructions

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

7.2.3 Branch Instructions

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 7-3.

Table 7-3. Branch Instructions

		Relative Addressing Mode		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

7.2.4 Bit Manipulation Instructions

The instructions are used on any bit in the first 256 bytes of memory; see Caution under **6.1 INPUT/OUTPUT.** One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 7-4.

Table 7-4. Bit Manipulation Instructions

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0 7)	—		—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0 7)	—		—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0 7)	10 + 2 • n	2	7			
Clear bit n	BCLR n (n = 0 7)	11 + 2 • n	2	7			

7.2.5 Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 7-5.

Table 7-5. Control Instructions

		Inherent		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No Operation	NOP	9D	1	2

7.2.6 Alphabetical Listing

The complete instruction set is given in a alphabetical order in Table 7-6.

Table 7-6. Instruction Set (Sheet 1 of 2)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	↕
BRSET										X	•	•	•	•	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	↕	↕	↕
COM	X		X			X	X				•	•	↕	↕	1
CPX		X	X	X		X	X	X			•	•	↕	↕	↕
DEC	X		X			X	X				•	•	↕	↕	•
EOR		X	X	X		X	X	X			•	•	↕	↕	•
INC	X		X			X	X				•	•	↕	↕	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
↕ Test and Set if True, Cleared Otherwise
• Not Affected

Table 7-6. Instruction Set (Sheet 2 of 2)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LDA		X	X	X		X	X	X			●	●	^	^	●
LDX		X	X	X		X	X	X			●	●	^	^	●
LSL	X		X			X	X				●	●	^	^	^
LSR	X		X			X	X				●	●	0	^	^
NEG	X		X			X	X				●	●	^	^	^
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	^	^	●
ROL	X		X			X	X				●	●	^	^	^
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols:
H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
^ Test and Set if True, Cleared Otherwise
• Not Affected

7.2.7 Opcode Map

Table 7-7 is an opcode map for the instruction used on the MCU.

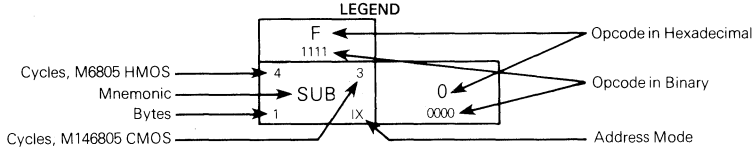
Table 7-7. M6805 HMOS/M146805 CMOS Family Instruction Set Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write				
	BTB	BSC	REL	DIR	A	X	IX1	IX
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111
0 0000	10 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEG 1 A	4 3 NEG 1 X	7 6 NEG 2 IX1	6 5 NEG 1 IX
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL					
2 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL					
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COM 1 A	4 3 COM 1 X	7 6 COM 2 IX1	6 5 COM 1 IX
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSR 1 A	4 3 LSR 1 X	7 6 LSR 2 IX1	6 5 LSR 1 IX
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL					
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 ROR 1 A	4 3 ROR 1 X	7 6 ROR 2 IX1	6 5 ROR 1 IX
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASR 1 A	4 3 ASR 1 X	7 6 ASR 2 IX1	6 5 ASR 1 IX
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSL 1 A	4 3 LSL 1 X	7 6 LSL 2 IX1	6 5 LSL 1 IX
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROL 1 A	4 3 ROL 1 X	7 6 ROL 2 IX1	6 5 ROL 1 IX
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DEC 1 A	4 3 DEC 1 X	7 6 DEC 2 IX1	6 5 DEC 1 IX
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL					
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INC 1 A	4 3 INC 1 X	7 6 INC 2 IX1	6 5 INC 1 IX
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TST 1 A	4 3 TST 1 X	7 5 TST 2 IX1	6 4 TST 1 IX
E 1110	10 7 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL					
F 1111	10 7 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLR 1 A	4 3 CLR 1 X	7 6 CLR 2 IX1	6 5 CLR 1 IX

Abbreviations for Address Modes

INH	Inherent	EXT	Extended	IX	Indexed (No Offset)
A	Accumulator	REL	Relative	IX1	Indexed, 1 Byte (8-Bit) Offset
X	Index Register	BSC	Bit Set/Clear	IX2	Indexed, 2 Byte (16-Bit) Offset
IMM	Immediate	BTB	Bit Test and Branch	*	M146805 CMOS Family Only
DIR	Direct				

	Control		Register/Memory						
	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
	9 9 RTI 1 INH		2 2 SUB 2 IMM	4 3 SUB 2 DIR	5 4 SUB 3 EXT	6 5 SUB 3 IX2	5 4 SUB 2 IX1	4 3 SUB 1 IX	0 0000
	6 6 RTS 1 INH		2 2 CMP 2 IMM	4 3 CMP 2 DIR	5 4 CMP 3 EXT	6 5 CMP 3 IX2	5 4 CMP 2 IX1	4 3 CMP 1 IX	1 0001
			2 2 SBC 2 IMM	4 3 SBC 2 DIR	5 4 SBC 3 EXT	6 5 SBC 3 IX2	5 4 SBC 2 IX1	4 3 SBC 1 IX	2 0010
	11 10 SWI 1 INH		2 2 CPX 2 IMM	4 3 CPX 2 DIR	5 4 CPX 3 EXT	6 5 CPX 3 IX2	5 4 CPX 2 IX1	4 3 CPX 1 IX	3 0011
			2 2 AND 2 IMM	4 3 AND 2 DIR	5 4 AND 3 EXT	6 5 AND 3 IX2	5 4 AND 2 IX1	4 3 AND 1 IX	4 0100
			2 2 BIT 2 IMM	4 3 BIT 2 DIR	5 4 BIT 3 EXT	6 5 BIT 3 IX2	5 4 BIT 2 IX1	4 3 BIT 1 IX	5 0101
			2 2 LDA 2 IMM	4 3 LDA 2 Dir	5 4 LDA 3 EXT	6 5 LDA 3 IX2	5 4 LDA 2 IX1	4 3 LDA 1 IX	6 0110
		2 2 TAX 1 INH		5 4 STA 2 DIR	6 5 STA 3 EXT	7 6 STA 3 IX2	6 5 STA 2 IX1	5 4 STA 1 IX	7 0111
	2 2 CLC 1 INH	2 2 EOR 2 IMM	2 2 EOR 2 DIR	4 3 EOR 3 EXT	5 4 EOR 3 IX2	6 5 EOR 2 IX1	5 4 EOR 1 IX	4 3 EOR	8 1000
	2 2 SEC 1 INH	2 2 ADC 2 IMM	4 3 ADC 2 DIR	5 4 ADC 3 EXT	6 5 ADC 3 IX2	5 4 ADC 2 IX1	4 3 ADC 1 IX		9 1001
	2 2 CLI 1 INH	2 2 ORA 2 IMM	4 3 ORA 2 DIR	5 4 ORA 3 EXT	6 5 ORA 3 IX2	5 4 ORA 2 IX1	4 3 ORA 1 IX		A 1010
	2 2 SEI 1 INH	2 2 ADD 2 IMM	4 3 ADD 2 DIR	5 4 ADD 3 EXT	6 5 ADD 3 IX2	5 4 ADD 2 IX1	4 3 ADD 1 IX		B 1011
	2 2 RSP 1 INH		3 2 JMP 2 DIR	4 3 JMP 3 EXT	5 4 JMP 3 IX2	4 3 JMP 2 IX1	3 2 JMP 1 IX		C 1100
	2 2 NOP 1 INH	8 6 BSR 2 REL	7 5 JSR 2 DIR	8 6 JSR 3 EXT	9 7 JSR 3 IX2	8 6 JSR 2 IX1	7 5 JSR 1 IX		D 1101
	* 2 STOP 1 INH		2 2 LDX 2 IMM	4 3 LDX 2 DIR	5 4 LDX 3 EXT	6 5 LDX 3 IX2	5 4 LDX 2 IX1	4 3 LDX 1 IX	E 1110
	* 2 WAIT 1 INH	2 2 TXA 1 INH		5 4 STX 2 DIR	6 5 STX 3 EXT	7 6 STX 3 IX2	6 5 STX 2 IX1	5 4 STX 1 IX	F 1111



SECTION 8
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications for the MC6805S2.

8.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage PC0 in Self-Check Mode All Other	V_{in}	−0.3 to +15.0 −0.3 to +7.0	V
Port A and C Source Current per Pin (One at a Time)	I_{out}	10	mA
Operating Temperature Range MC6805S2 MC6805S2C	T_A	0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	
Junction Temperature Plastic Package Ceramic Package Cerdip	T_J	150 175 175	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical 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 tied to an appropriate voltage level (e.g., either V_{SS} or V_{CC}).

8.2 THERMAL CHARACTERISTICS

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

8.3 POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

8.4 ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}+0.7	V
(V_{CC} ≤ 4.75)		V_{CC}–0.5	–	V_{CC}+0.7	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}+0.7	
(V_{CC} ≤ 4.75)		V_{CC}–0.5	*	V_{CC}+0.7	
All Other		2.0	–	V_{CC}+0.7	
Input High Voltage PC0					
Port/Timer Mode	V_{IH}	2.0	–	V_{CC}+1.0	V
Self-Check Mode		9.0	10.0	15.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other (Except A/D Inputs)		V_{SS}	–	0.8	
$\overline{RESET}$ Hysteresis Voltages (See RESETS)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
Standby Supply Voltage ($\overline{INT2}$ Input Option)	V_{STBY}	3.0	–	5.75	V
Standby Current ($\overline{INT2}$ Input Option) (V_{STBY} = 3.0 V)	I_{STBY}	–	1.0	TBD	mA
Power Dissipation – No Port Loading					
(V_{CC} = 5.75 V, T_A = 0°C)	P_D	–	600	TBD	mW
(V_{CC} = 5.75 V, T_A = –40°C)	P_D	–	670	TBD	
Input Capacitance (Except Analog Inputs – See Note)	C_{in}	–	10	–	pF
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	2.75	3.75	4.70	V
Input Current					
$\overline{INT}$					
(V_{in} = 2.4 V to V_{CC})	I_{in}	–	20	50	μA
EXTAL					
(V_{in} = 2.4 V to V_{CC} Crystal Option)		–	–	10	
(V_{in} = 0.4 V Crystal Option)		–	–	–1600	
$\overline{RESET}$					
(V_{in} = 0.8 V)		–4.0	–	–50	
(External Charging Current)					

TBD = To Be Determined

NOTE: Port D analog inputs, when selected, C_{in} = 25 pF for the first 5 out of 30 cycles.

* This input (when unused) floats to approximately 2.0 V due to internal biasing.

8.5 SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time ($4/f_{osc}$)	t_{cyc}	0.95	–	10	μs
$\overline{INT}$, $\overline{INT2}$, and TIMER Pulse Width (See 5.4 INTERRUPTS)	t_{WL}, t_{WH}	t_{cyc}+250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc}+250	–	–	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1 μF)	t_{RHL}	–	100	–	ms
$\overline{INT}$ Zero-Crossing Detection Input Frequency (for ±5° Accuracy)	f_{INT}	0.03	–	1	kHz
External Clock Input Duty Cycle (EXTAL)	–	40	50	60	%
Oscillator Startup Time Crystal*	t_{su}	–	–	100	ms
SPICL High Time	t_{SPICLH}	4	–	–	t_{cyc}
SPICL Low Time	t_{SPICHL}	4	–	–	t_{cyc}
SPICL Rise and Fall Time	t_{Sr}, t_{Sf}	–	–	1	μs
SPID Input Data Setup Time	t_{SDS}	2	–	–	t_{cyc}
SPID Input Data Hold Time	t_{SDH}	2	–	–	t_{cyc}
SPICL to SPISS Lag Time	t_{SStG}	4	–	–	t_{cyc}
SPISS to SPICL Lead Time	t_{SSLD}	4	–	–	t_{cyc}
SPISS High and Low Time	t_{SSH}, t_{SSL}	4	–	–	t_{cyc}
Start Bit to First Clock Lead Time	t_{STL}	1	–	–	t_{cyc}
External Timer Input to Timer Change Time	t_{PCT}	3	–	–	t_{cyc}
Timer Change to Port B Toggle Time	t_{TPB}	2	–	–	t_{cyc}
$\overline{INT2}$ to Timer A Load Time	t_{INTL}	3	–	–	t_{cyc}

*See Figure 5-7 for typical crystal parameters.

8.6 A/D CONVERTER CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Min	Typ	Max	Unit	Comments
Resolution	8	8	8	Bits	
Non-Linearity*	–	–	± ½	LSB	After removing zero-offset and full-scale errors
Quantizing Error	–	–	± ½	LSB	
Full Scale Error*	–	–	TBD	LSB	Deviation of step $FE to $FF from ideal
Zero Offset*	–	–	TBD	LSB	Deviation of step $00 to $01 from ideal
Absolute (Total) Error*	–	–	TBD	LSB	Includes errors due to all sources, including quantizing
Conversion Range V_{RH} V_{RL}	 – V_{SS}	 – –	 V_{CC} 0.2	 V	A/D accuracy may decrease proportionately as V_{RH}-V_{RL} is reduced below 4.0 V. The sum of V_{RH} and V_{RL} must not exceed V_{CC}.
Conversion Time	30	30	30	t_{cyc}	Includes sampling time
Monotonicity	(Inherent with in total error)				
Sample Time	5	5	5	t_{cyc}	
Sample/Hold Capacitance, Input	–	–	25	pF	
Analog Input Voltage	V_{RL}	–	V_{RH}	V	Transients on any analog lines (pins 19-24) are not allowed at any time during sampling or accuracy may be degraded.

*For V_{RH} = 4.0 V to 5.0 V and V_{RL} = 0 V.

8.7 PORT ELECTRICAL CHARACTERISTICS (VCC = +5.25 Vdc, ±0.5 Vdc, VSS = 0 Vdc, TA = TL to TH, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Output High Voltage I_{Load} = – 10 μA	V_{OH}	V_{CC} – 1.0	–	–	V
Input High Voltage I_{Load} = – 300 μA (Maximum)	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage I_{Load} = – 500 μA (Maximum)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	– 300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	– 500	μA
Port B					
Output Low Voltage I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.0	V
Output High Voltage* I_{Load} = – 200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source)* V_O = 1.5 V	I_{OH}	– 1.0	–	– 10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	< 2	10	μA
Port C and Port A with CMOS Device Disabled					
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	< 2	10	μA
Port D (Digital Inputs Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Input Current**	I_{in}	–	< 1	5	μA

*Not applicable if programmed to open-drain state.

**PD4/V_{RL} PD5/V_{RH}:
The A/D conversion resistor (15 kilohm typical) is connected internally between these two lines, impacting their use as digital inputs in some applications.

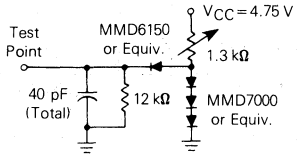


Figure 8-1. TTL Equivalent Test Load (Port B)

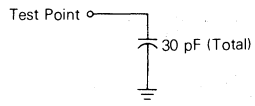


Figure 8-2. CMOS Equivalent Test Low (Port A)

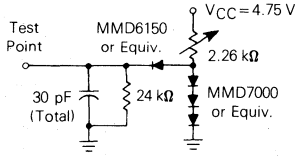


Figure 8-3. TTL Equivalent Test Load (Ports A and C)

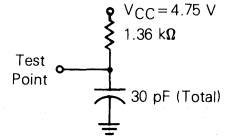


Figure 8-4. Open-Drain Equivalent Test Load (PB1, PB2, and PB3)

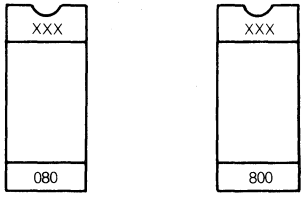
SECTION 9
ORDERING INFORMATION

The information required when ordering a custom MCU is given in the following paragraphs. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the MCU it is necessary to first contact your local Motorola representative or Motorola distributor.

9.1 EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM must be clearly marked to indicate which EPROM corresponds to which address space. The recommended marking procedure for two MCM2716 EPROMs is illustrated below.



XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

9.2 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 one blank EPROM from the data file used to create the customer mask to aid in the verification process.

9.3 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 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs are thus not guaranteed by Motorola Quality Assurance, and should be discarded after verification is completed.

9.4 FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name of the disk with a felt-tip pen. The minimum MDOS system files as well as the absolute binary object file (file name LO type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: file name LX (EXORciser loadable format) and file name SA (ASCII source code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

MDOS is Motorola's disk operating system available on development systems such as EXORcisers, EXORsets, etc.

MC6805S2 MCU CUSTOM ORDERING INFORMATION

Motorola Use Only

Motorola Part Numbers
MC_____
SC_____

Date_____

Customer PO Number_____

Customer Company_____

_____

Address_____

City_____ State_____ Zip _____

Country_____

Telephone_____ Extension_____

Custom Contact Person_____

Customer Part Number_____

OPTION LIST
Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Standby RAM
- No Standby
- 16-Byte Standby

Internal Oscillator Input
- Crystal
- Resistor

Port A Output Drive
- CMOS and TTL
- TTL Only

Analog-to-Digital Channels
- Four (Standard)
- Five

Low Voltage Inhibit
- Disable
- Enable

Pattern Media (All other media require prior factory approval)
- EPROMs (MCM2716 or MCM2532)
- Floppy Disk
- Other*_____

_____

*Requires prior factory approval

Clock Frequency_____

Temperature Range: 0°C to +70°C (Standard) −40°C to +85°C

Marking Information (12 Characters Maximum)_____

Signature_____

Title_____

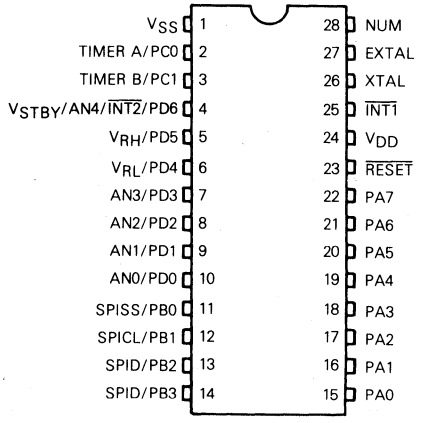
GENERIC INFORMATION

Package Type	Temperature	Generic Number
Plastic	0°C to 70°C	MC6805S2P
P Suffix	−40°C to 85°C	MC6805S2CP
Ceramic	0°C to 70°C	MC6805S2L
L Suffix	−40°C to 85°C	MC6805S2CL
Cerdip	0°C to 70°C	MC6805S2S
S Suffix	−40°C to 85°C	MC6805S2CS

SECTION 10
MECHANICAL DATA

This section contains the pin assignments and package dimensions for the MC6805S2.

10.1 PIN ASSIGNMENT



MOTOROLA

MC6805T2

Advance Information

HMOS

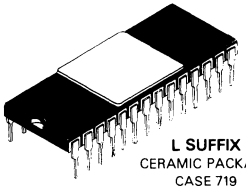
(HIGH DENSITY, N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

8-BIT MICROCOMPUTER WITH PLL LOGIC

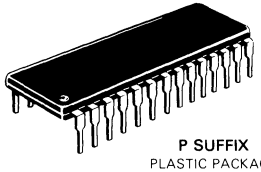
3

8-BIT MICROCOMPUTER UNIT WITH PLL LOGIC

The MC6805T2 Microcomputer Unit (MCU) with PLL logic is a member of the M6805 HMOS Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains a CPU, on-chip clock, ROM, RAM, I/O, timer, and the PLL logic for an RF synthesizer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set, as well as the necessary logic required for frequency synthesis applications. The following are some of the hardware and software highlights of the MC6805T2 MCU.

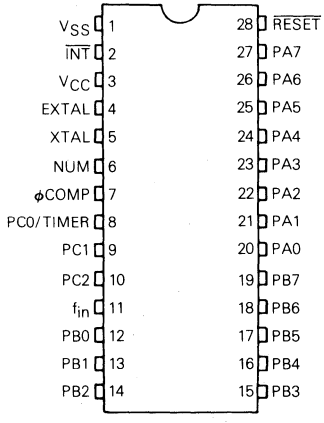


L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENT



HARDWARE FEATURES

- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 2508 Bytes of User ROM
- Timer Start/Stop and Source Select
- 19 TTL/CMOS Compatible Bidirectional I/O Lines (Eight Lines are LED Compatible)
- On-Chip Ciock Generator
- Zero-Crossing Detection
- Self-Check Mode
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply
- 14-Bit Binary Variable Divider
- 10-Stage Mask-Programmable Reference Divider
- Three-State Phase and Frequency Comparator
- Suitable for TV Frequency Synthesizers

SOFTWARE FEATURES

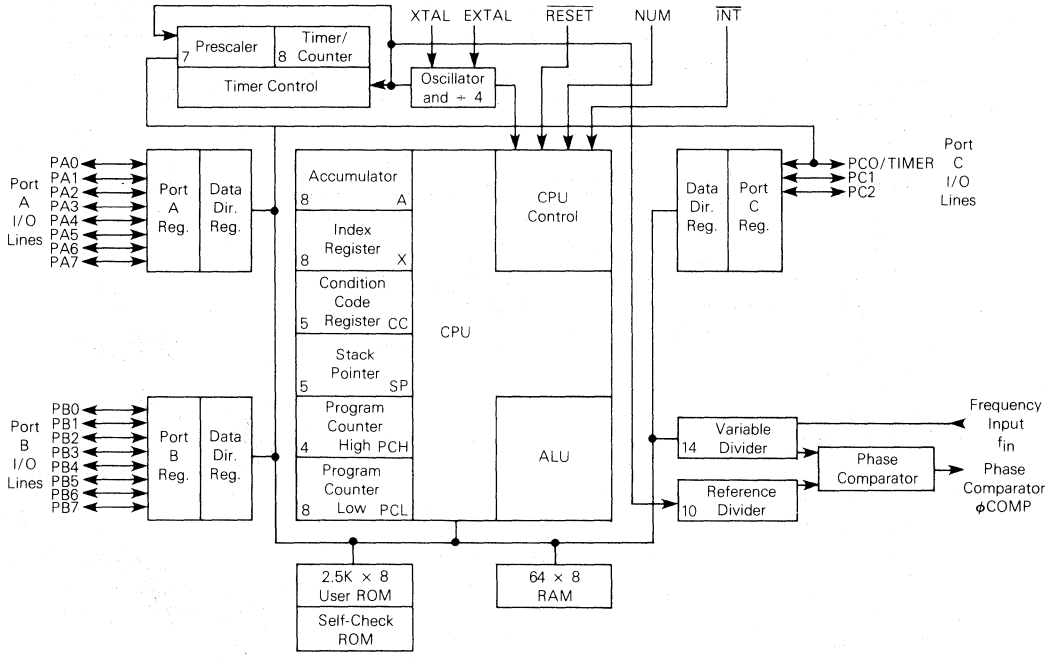
- Similar to M6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O

USER SELECTABLE OPTIONS

- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (Seven Bits 2^n)
- Eight Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option or Open-Drain Drive
- Four Vectored Interrupts; Timer, Software, and two External
- Low Voltage Inhibit Option

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

FIGURE 1 — MC6805T2 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage (Except Pin 6)	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	120	°C/W
Ceramic		50	
Cerdip		60	

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

POWER CONSIDERATIONS

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

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

Where:

T_A ≡ Ambient Temperature, °C

θ_{JA} ≡ Package Thermal Resistance, Junction-to-Ambient, °C/W

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

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

P_{PORT} ≡ Port Power Dissipation, Watts — User Determined

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

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

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

Solving equations 1 and 2 for K gives:

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

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

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25 Vdc ±0.5 Vdc, V_{SS} = GND, T_A = 0 to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} – 0.5	–	V_{CC}	
$\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	*	V_{CC}	
(V_{CC} < 4.75)		V_{CC} – 0.5	*	V_{CC}	
All Other Except f_{in}		2.0	–	V_{CC}	
Input High Voltage ϕCOMP					
Normal Mode	V_{IH}	–	–	V_{CC} + 1	V
Self-Check Mode		–	10.0	10.0	
Input Low Voltage					
$\overline{RESET}$	V_{IL}	V_{SS}	–	0.8	V
$\overline{INT}$		V_{SS}	*	1.5	
All Other Except f_{in}		V_{SS}	–	0.8	
$\overline{INT}$ Zero-Crossing Input Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	$V_{ac\ p\text{-}p}$
Internal Power Dissipation – No Port Loading, V_{CC} = 5.75 V, T_A = 0°C	P_{INT}	–	400	–	mW
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Others		–	10	–	
AC Coupled Input Voltage Swing on f_{in}	V_{FIP}	0.5	1.2	–	$V_{ac\ p\text{-}p}$
Input Current (V_{IH} = V_{CC}) on Pin 11 (f_{in})	I_{FH}	–	–	40	μA
Output Low Current (V_{OL} = 1.0 V) on Pin 7 (ϕCOMP)	I_{CML}	–	–300	–	μA
Output High Current (V_{OH} = V_{CC} – 1 V) on Pin 7 (ϕCOMP)	I_{CMH}	–	200	–	μA
Leakage Current (V_{in} = V_{CC}) on Pin 7 (ϕCOMP)	I_{OFF}	–	2	–	nA
$\overline{RESET}$ Hysteresis Voltage (See Figures 10 and 11)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{INT}$ (V_{in} = 0.4 V)		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC})		–	–	10	
_____ (V_{in} = 0.4 V)		–	–	–1600	
$\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Charging Current)		–4.0	–	–40	
Low Voltage Receiver	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit 0°C to 70°C	V_{LVI}	2.75	3.5	–	V
–40°C to 85°C		3.1	3.5	–	

See MC68(7)05 Series Data Sheet for port I/V curves and input protection schematics.
* Due to internal biasing, this $\overline{INT}$ input (when unused) floats to approximately 2.0 volts.

SWITCHING CHARACTERISTICS (V_{CC} = 5.25 Vdc ±0.5 Vdc, V_{SS} = GND, T_A = 0 to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Typ	Max	Unit
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time ($4/f_{osc}$)	t_{cyc}	0.95	–	10	μs
$\overline{INT}$ and TIMER Pulse Width (See TIMER and INTERRUPT Sections)	t_{WH}, t_{WL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	–	100	–	ms
Input Frequency	f_{in}	1	–	16	MHz
Input Frequency Rise Time (f_{in} = max)	t_{INR}	–	–	20	ns
Input Frequency Fall Time (f_{in} = max)	t_{INF}	–	–	20	ns
Duty Cycle of f_{in}		40	–	60	%
Injection Pulse Active Time	t_{ERR}	–	70	–	ns
$\overline{INT}$ Zero-Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)	–	40	50	60	%

PORT ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = GND, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A with CMOS Drive Enabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = – 10 μA	V_{OH}	V_{CC} + 1	–	–	V
Input High Voltage, I_{Load} = – 300 μA (max)	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, I_{Load} = – 500 μA (max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	– 300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	– 500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = – 200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	– 1.0	–	– 10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA
Port C and Port A with CMOS Drive Disabled					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	10	μA

FIGURE 2 – TTL EQUIVALENT TEST LOAD (PORT B)

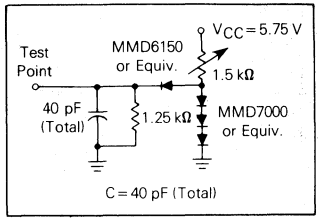


FIGURE 3 – CMOS EQUIVALENT TEST LOAD (PORT A)

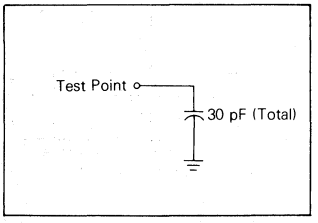
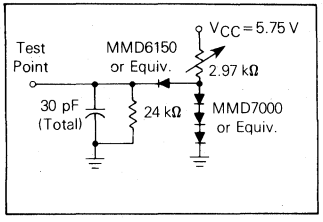


FIGURE 4 – TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

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.

$\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to **INTERRUPTS** for additional information.

XTAL AND EXTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal or an external signal can be connected to these pins to provide input to the internal oscillator. Lead length and stray capacitance on these two pins should be minimized. Refer to **INTERNAL OSCILLATOR** for recommendations about these pins.

f_{in}

This pin provides the high frequency digital input to the variable divider portion of the on-chip frequency synthesizer. The reference frequency for the phase lock loop is divided down from the crystal oscillator. Refer to the **PHASE LOCK LOOP** for details on the frequency synthesizer features.

ϕCOMP

This three-state output is the result of comparing the internal reference frequency to the variable divider signal. Refer to **PHASE LOCK LOOP** for details. In self-check, ϕCOMP is raised to ≈9 Volts.

$\overline{\text{RESET}}$

A low voltage level on this Schmitt trigger input will reset the MPU. Refer to **RESETS** for additional information.

NUM

This pin is not for user application and must be connected to V_{SS}.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC2)

These 19 lines are arranged into two 8-bit ports (A and B) and one 3-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to **INPUT/OUTPUTS** for additional information. The PC0/TIMER pin also serves as an external input to the internal timer. Refer to the **TIMER** section for information on the timer modes.

MEMORY

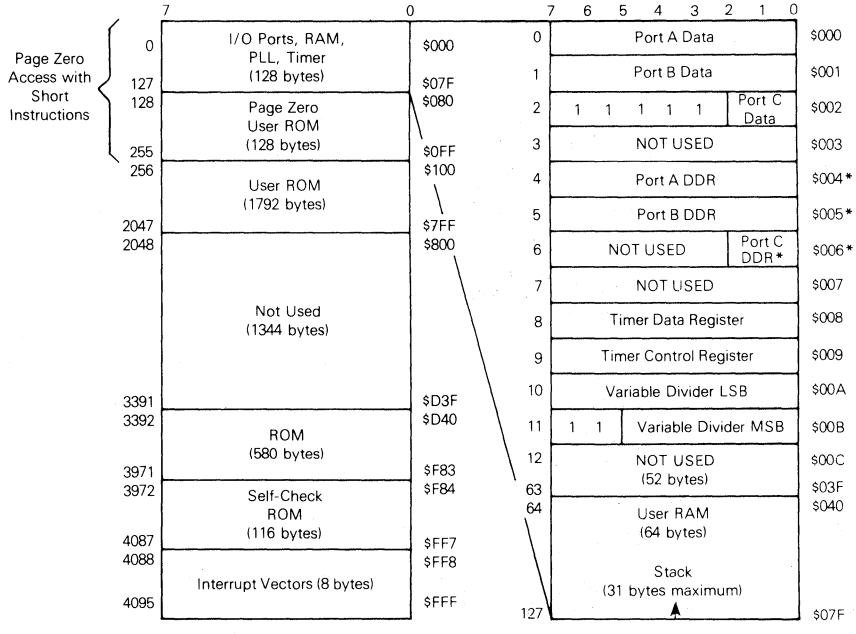
The MCU memory is configured as shown in Figure 5. The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The MCU has implemented 2698 of these memory locations. This consists of: 2508 bytes user ROM, 116 bytes self-check ROM, 64 bytes of user RAM, six bytes of port I/O, two timer registers, and two PLL registers. The user ROM is split into two areas. The first area begins at memory location $080 and continues through $7FF. The lower 128 bytes of this ROM area (part of page zero) allows the user to access ROM locations utilizing the direct and table look-up indexed adressing modes. The second user ROM area begins at memory location $D40 and continues through $F83. The last eight user ROM locations, at the top of memory, are for the interrupt vectors.

The MCU reserves the first 16 memory locations for I/O features, of which 10 have been implemented. These locations are used for the ports, the port DDRs, the timer, and the PLL registers.

Sixty-four bytes of user RAM are provided. Of the 64 bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

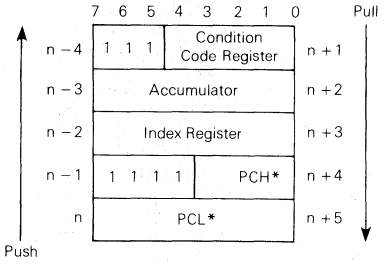
The shared stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 6. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly following pulls from the stack, since the stack pointer increments when it pulls data from the stack. A subroutine call results in only the program counter (PCH, PCL) contents being pushed onto the stack, the remaining CPU registers are not pushed.

FIGURE 5 — MC6805T2 MCU ADDRESS MAP



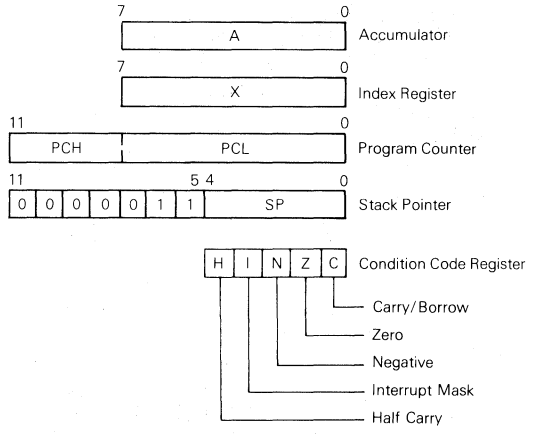
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 6 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCH and PCL are stacked.

FIGURE 7 — PROGRAMMING MODEL



3

REGISTERS

The MCU has five registers available to the programmer. They are shown in Figure 7 and are explained in the following paragraphs.

ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read/modify/write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC)

The program counter is a 12-bit register that contains the address of the *NEXT* instruction to be executed.

STACK POINTER (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The seven most-significant bits of the stack pointer are permanently set to 0000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum) which allows the programmer to use up to 15 levels of subroutine calls. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

HALF CARRY (H) — Set during ADD and ADC instructions to indicate that a carry occurred between bits 3 and 4.

INTERRUPT (I) — This bit is set to mask (disable) the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

NEGATIVE (N) — Used to indicate that the result of the *LAST* arithmetic, logical, or data manipulation was negative (bit 7 in result equal to a logic one).

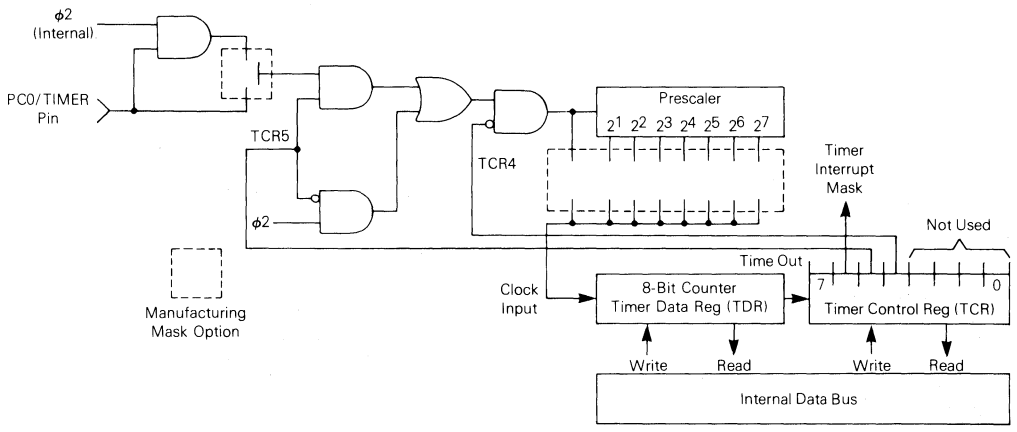
ZERO (Z) — Used to indicate that the result of the *LAST* arithmetic, logical, or data manipulation was zero.

CARRY/BORROW (C) — Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the *LAST* arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The MC6805T2 timer circuitry is shown in Figure 8. The 8-bit counter may be loaded under program control and is decremented toward zero by the clock input (prescaler output). When the timer reaches zero the timer interrupt request

FIGURE 8 — TIMER BLOCK DIAGRAM



bit (bit 7) in the timer control register (TCR) is set. The timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the condition code register also prevents a timer interrupt from being processed. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine; see **INTERRUPTS**. *The timer interrupt request bit (TCR7) must be cleared by software.*

The clock input to the timer is established via bit 5 (TCR5) in the timer control register. When this bit is set (external mode), the timer clock source is the PC0/TIMER pin. In this mode a mask option is used to select either the $\phi 2$ gated with PC0/TIMER or the positive transition on PC0/TIMER as timer clock source. This allows easily performed pulse width or pulse count measurements. When TCR5 is low, logic zero, the timer clock source is the internal $\phi 2$.

Bit 4 in the timer control register (TCR4) disables the timer clock source when set to logic one.

The maximum frequency of a signal that can be recognized by the PC0/TIMER pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period).

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily long period (250 nanoseconds twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the PC0/TIMER input pin allowing the user to easily perform pulse-width measurements. (Note: for ungated $\phi 2$ clock inputs to the timer prescaler, the PC0/TIMER pin should be tied to V_{CC}.) The source of the clock input is one of the mask options that is specified before manufacture of the MC6805T2.

A prescaler option, divide by 2^n, can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling mask option is also specified before manufacture.

The timer continues to count past zero, falling through to $FF from zero, and then continuing to count. Thus, the counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred, and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logic ones; the timer interrupt request bit (TCR7) is cleared; the timer interrupt mask bit (TCR6) is set; the external timer source bit (TCR5) is cleared and the timer disable bit (TCR4) is cleared.

SELF-CHECK

The self-check capability of the MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 9 and monitor the output of Port C bit 3 for an oscillation of approximately 3 hertz. A 9-volt level on the ϕCOMP input, pin 7, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, timer, interrupts, and I/O ports.

RESETS

The MCU is reset whenever the $\overline{\text{RESET}}$ input line senses a logic zero. This can be accomplished in two different ways: (1) during power-up when a capacitor is used to hold the $\overline{\text{RESET}}$ pin low for a specified time (t_{RHL}); and (2) any time after power-up that the $\overline{\text{RESET}}$ line falls to a logic zero for a period longer than one t_{cyc}. See Figures 10 and 11.

The internal circuit connected to the $\overline{\text{RESET}}$ pin consists of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic 0 on the $\overline{\text{RESET}}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{\text{RESET}}$ pin voltage rises to V_{IRES+}. When the $\overline{\text{RESET}}$ pin voltage falls to a logic 0 for a period longer than one t_{cyc}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 11(b).

Upon power-up, a delay of t_{RHL} is needed before allowing the reset input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{\text{RESET}}$ input, as shown in Figure 11(a), will provide sufficient delay. See Figure 15 under **INTERRUPTS** for the complete reset sequence.

3

INTERNAL OSCILLATOR

The internal oscillator circuit has been designed to require a minimum of external components. The use of a crystal or an external signal may be used to drive the internal oscillator. The different connection methods are shown in Figures 12 and 13. The crystal specifications and suggested PC board layout are given in Figure 14.

The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

INTERRUPTS

The MC6805T2 MCU can be interrupted three different ways: through the external interrupt ($\overline{\text{INT}}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires a total of 11 t_{cyc} periods for completion.

A flowchart of the interrupt sequence is shown in Figure 15. The interrupt service routine must end with a return from interrupt instruction (RTI) which allows the MCU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state).

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause execution of the current instruction to be halted. Hardware interrupts are considered pending until execution of the current instruction is complete.

As shown in Figure 15, when execution of the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked (I bit=0), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interupts are latched internally for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI (software interrupt) instruction is executed as any other instruction and as such will take precedence over hardware interrupts only if the I bit is set (interrupts masked).

Table 1 shows the execution priority of the $\overline{\text{RESET}}$, $\overline{\text{INT}}$ and timer interrupts, and instructions (including the software interrupt, SWI). Two conditions are shown, one with the I bit set and the other with I bit clear; however, in either case $\overline{\text{RESET}}$ has the highest priority of execution. If the I bit is set as per Table 1(a), the second highest priority is assigned to any instruction including SWI. This is illustrated in Figure 15 which shows that the $\overline{\text{INT}}$ or timer interrupts are not tested when the I bit is set. If the I bit is cleared as per Table 1(b), the priorities change in that the next instruction (including SWI) is not fetched until after the $\overline{\text{INT}}$ and timer interrupts have been tested (and serviced). Also, when the I bit is clear, if both $\overline{\text{INT}}$ and timer interrupts are pending, the $\overline{\text{INT}}$ interrupt is always serviced before the timer interrupt.

The external interrupt is internally synchronized and will set a latch on the falling edge of $\overline{\text{INT}}$. A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 16(a) for use as a zero-crossing detector with hystersis included. This allows for applications such as time-of-day routines and engaging/disengaging ac power control devices. As shown in Figure 17(a), off-chip clamping limits the ac input to the V_{INT} specification while still providing an interrupt at every zero crossing of the ac signal.

For digital applications, as shown in Figure 16(b), the $\overline{\text{INT}}$ input can be driven directly by a digital signal. The maximum frequency of a signal that can be recognized by the $\overline{\text{INT}}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period).

$$t_{cyc} \times 2 + 250\ \text{ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 nanoseconds twice). For the $\overline{\text{INT}}$ function, the maximum allowable frequency is also determined by the software response of the INT service routine.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I bit in the condition code register; however, if the I bit is clear, a software interrupt (or any other instruction) cannot be executed until after the $\overline{\text{INT}}$ and timer interrupts have been serviced. SWIs are usually used as breakpoints for debugging or as system calls.

FIGURE 9 — SELF-CHECK CONNECTIONS

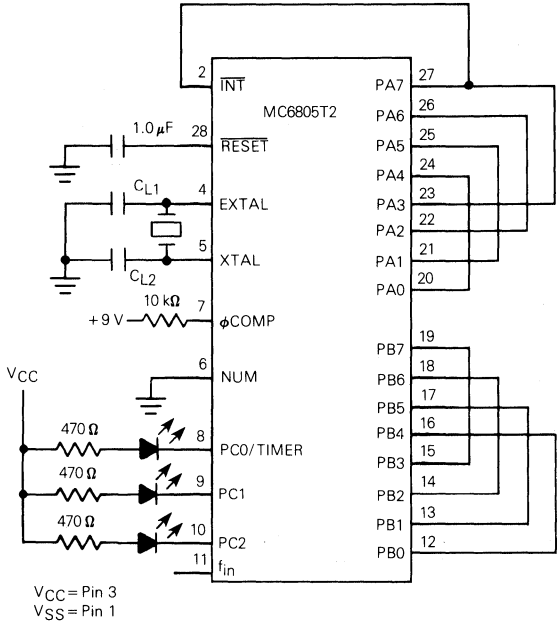


FIGURE 10 — POWER AND $\overline{\text{RESET}}$ TIMING

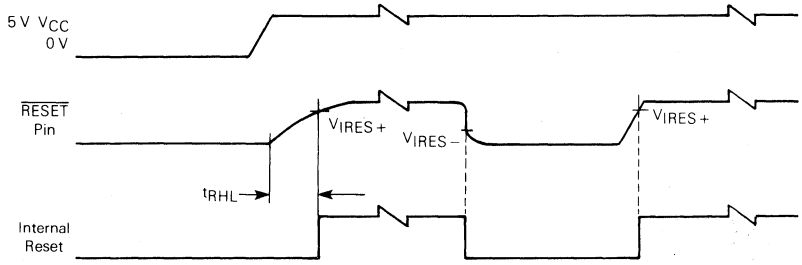


FIGURE 11 — POWER-UP RESET DELAY CIRCUIT

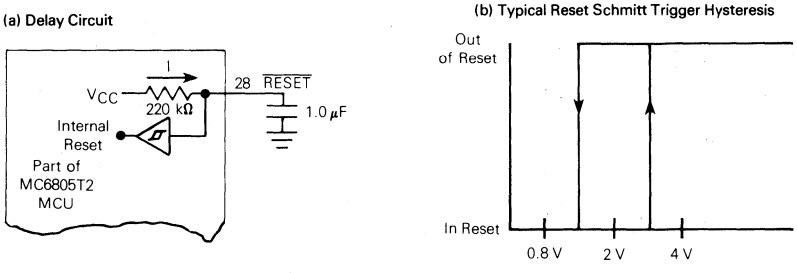


FIGURE 12 — CRYSTAL OSCILLATOR

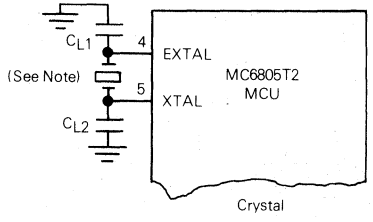
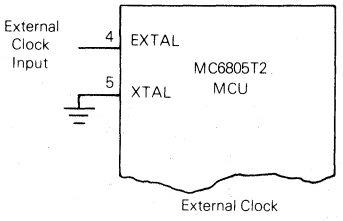


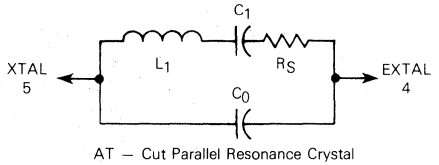
FIGURE 13 — EXTERNAL OSCILLATOR



NOTE: The recommended value of both C_{L1} and C_{L2} with a 4.0 megahertz crystal is 27 picofarads maximum, including system distributed capacitance. For crystal frequencies other than 4 megahertz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2-megahertz crystal, use approximately 50 picofarads for both C_{L1} and C_{L2}. The exact value depends on the motional-arm parameters of the crystal used.

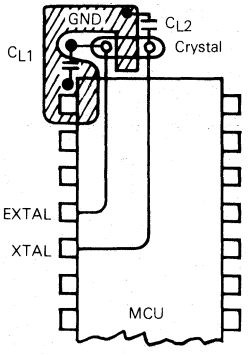
FIGURE 14 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT

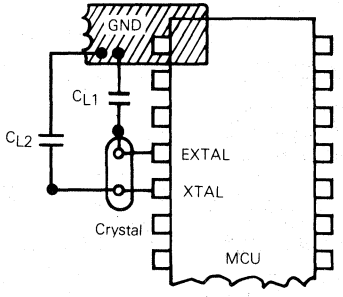
(a) Recommended Crystal Motional-Arm Parameters



AT — Cut Parallel Resonance Crystal
C_0 = 7 pF Max.
FREQ = 4.0 MHz @ C_{L1} and C_{L2} = 24 pF
R_S = 50 ohms Max.

(b) Suggested PC Board Layouts





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

FIGURE 15 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

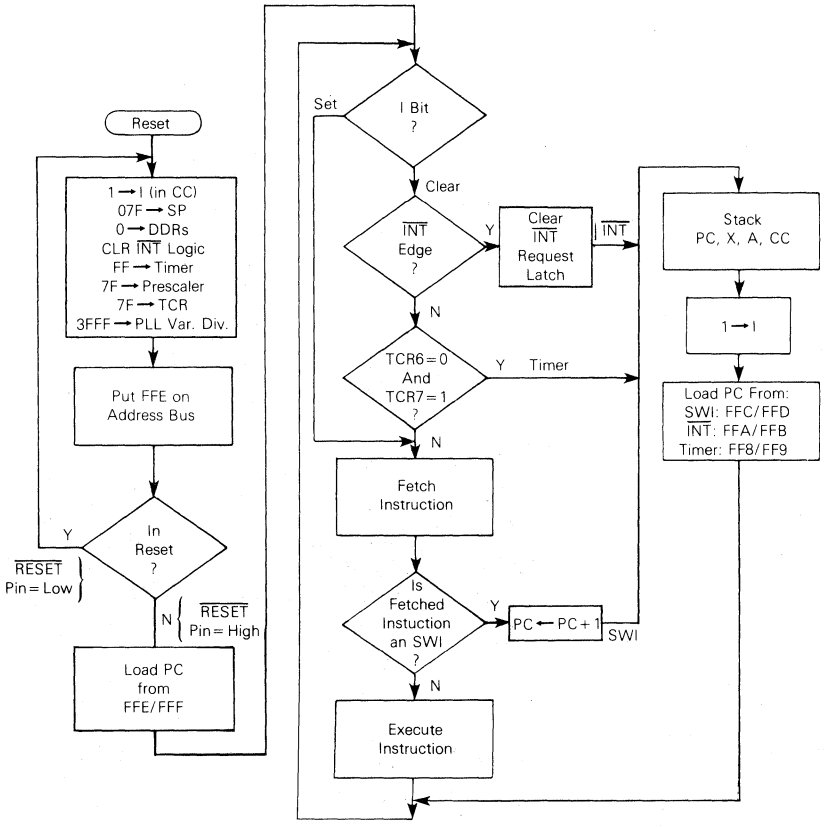


TABLE 1 — INTERRUPT/INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$FFE-$FFF
SWI (or Other Instruction)	2	$FFC-$FFD

NOTE: $\overline{\text{INT}}$ and Timer Interrupts are not tested when the I bit is set; therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$FFE-$FFF
$\overline{\text{INT}}$	2	$FFA-$FFB
Timer	3	$FF8-$FF9
SWI (or Other Instruction)	4	$FFC-$FFD

INPUT/OUTPUT

There are 19 input/output pins. (The $\overline{\text{INT}}$ pin may also be polled with branch instructions to provide an additional input pin.) All pins (Ports A, B, and C) are programmable as either inputs or outputs under software control of the corresponding data direction registers (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output or a logic 0 for input. On reset all DDRs are initialized to a logic 0 state to put the ports in the input mode. The port output registers are not initialized on reset and should be initialized before changing the DDR bits to avoid undefined levels. When programmed as outputs, all I/O pins read the latched output data, regardless of the logic levels at the output pin due to output loading; see Figure 17. When Port B is programmed for outputs, it is capable of sinking 10 milliamperes and sourcing one milliampere on each pin.

FIGURE 16 — TYPICAL INTERRUPT CIRCUITS

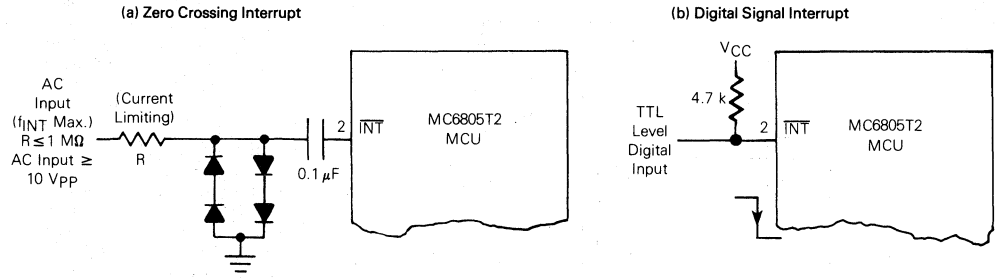
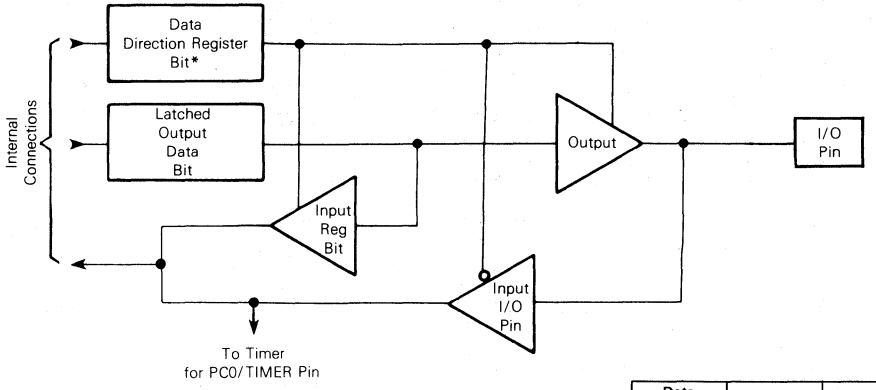


FIGURE 17 — TYPICAL PORT I/O CIRCUITRY



Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	3-State**	Pin

*DDR is a write-only register and reads as all 1s.

**Ports A (with CMOS drive disabled), B, and C are three-state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. Figure 18 provides some examples of port connections. The address map in Figure 5 gives the address of the data registers and DDRs. The register configuration is shown in Table 2.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read/modify/write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

The latched output data bit (see Figure 17) may always be written. Therefore, any write to a port writes 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).

PHASE LOCK LOOP

The PLL section consists of: a 14-bit binary variable divider, a fixed 10-stage divider, a digital phase and frequency comparator with a three-state output, and circuitry to avoid "backlash" effects in phase lock conditions.

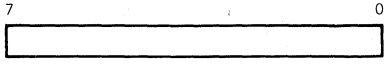
With a suitable high-frequency prescaler and an active integrator the user can easily establish a frequency synthesizer system driving a voltage controlled oscillator, as shown in Figure 19. The equations governing the PLL are given in Figure 20.

3

TABLE 2 — MCU REGISTER CONFIGURATION

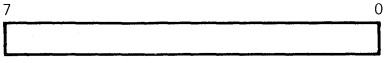
PORT DATA REGISTER

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002

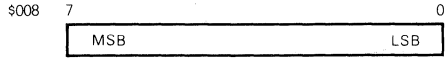


PORT DATA DIRECTION REGISTER (DDR)

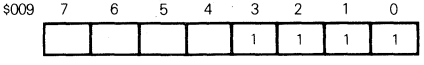
(1) Write Only; reads as all 1s
(2) 1 = Output, 0 = Input. Cleared to 0 by Reset.
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006



TIMER DATA REGISTER (TDR)



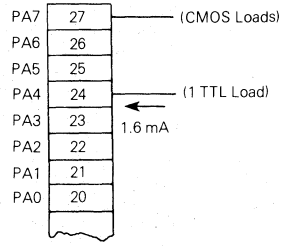
TIMER CONTROL REGISTER (TCR)



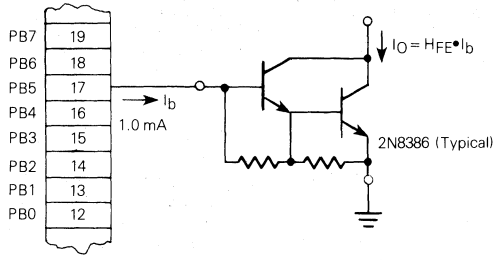
TCR Bits 0, 1, 2, and 3 read as 1s (not used).
TCR4 — Disable Timer: 1 = Timer Stopped. 0 = Timer Allowed to Count. Cleared to 0 by Reset.
TCR5 — External Timer Source: 1 = External, 0 = Internal $\phi 2$. Cleared to 0 by Reset.
TCR6 — Timer Interrupt Mask Bit: 1 = timer interrupt masked (disabled). Set to 1 by Reset.
TCR7 — Timer Interrupt Request Status Bit: Set when TDR goes to zero, must be cleared by software. Cleared to 0 by Reset.

FIGURE 18 — TYPICAL PORT CONNECTIONS

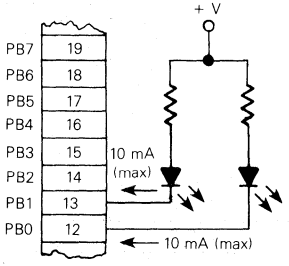
(a) Output Modes



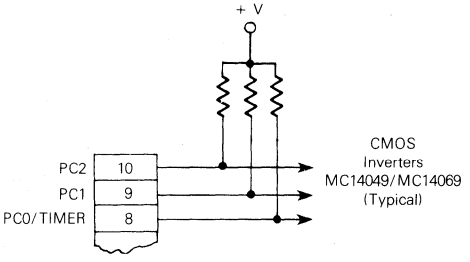
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

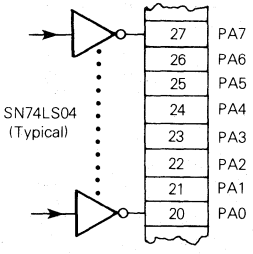


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

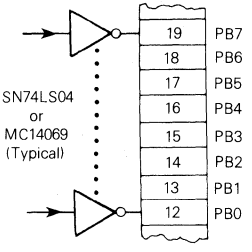


Port C, Bits 0-2 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors.

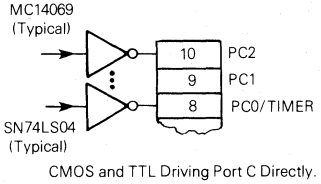
(b) Input Modes



TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly.



CMOS and TTL Driving Port C Directly.

VARIABLE DIVIDER

The variable divider is a 14-bit binary down counter which communicates with the CPU via two read/write registers located at address $00A, for the LS byte, and $00B, for the MS byte. The upper two bits in register $00B, always read as logic 1s. When the variable divider count has reached zero a preset pulse, f_{VAR}, is generated.

The f_{VAR} signal is applied to the phase comparator circuit together with the f_{REF} signal. The phase/frequency difference, between the two signals, results in an error signal output (ϕCOMP, pin 7) which controls the VCO frequency. In addition, the f_{VAR} pulse is also used to reload the 14-bit variable divider latch as shown in Figure 21.

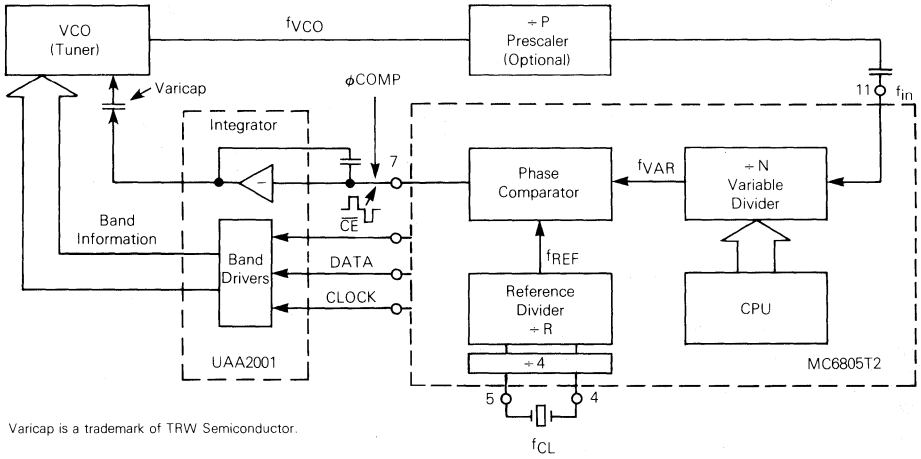
Data transfers from registers $00A and $00B to the latch occur outside the preset time and only during a write operation performed on register $00A. For example; a 6-bit data transfer to register $00B is only transferred to the variable divider if followed by a write operation to register $00A. For correct operation of the variable divider, the absolute value of the four lower significant bits of the 14-bit binary code (loaded into the 14-bit latch) must be less than or equal to the absolute value of the upper 10 bits. Figure 22 shows a typical manipulation of the 14-bit data to the registers.

The use of the 14-bit latch synchronizes the data transfer between two asynchronous systems, namely, the CPU and the variable divider.

At power-up reset both the variable divider and the contents of the PLL registers are set to logic 1s.

The variable frequency input pin, f_{in}, is self biased requiring an ac coupled signal with a minimum swing of 0.5 V. The input frequency range of f_{in} allows the device, together with a suitable prescaler, to cover a given frequency spectrum. For example, with a ÷64 prescaler the entire television frequency spectrum can be covered.

FIGURE 19 — PHASE LOCK LOOP AN AN RF FREQUENCY SYNTHESIZER



Varicap is a trademark of TRW Semiconductor.

FIGURE 20 — PRINCIPAL PLL EQUATIONS

For a system in lock:

$f_{VAR} = f_{REF}$

$f_{VAR} = f_{in} \div N$

$f_{in} = f_{VCO} \div P$

$F_{VCO} = f_{REF} \times P \times N$

Where: P = Prescaler division ratio
N = Variable Divider division ratio

Minimum frequency step =

$$\frac{\Delta f_{VCO}}{\Delta N} = f_{REF} \times P$$

Example for determining minimum frequency step:

$f_{CL} = 4.00$ MHz = Crystal frequency

$f_{CL/4} = 1.00$ MHz = Internal Oscillator frequency

$R = 2^{10}$ = Reference Divider ratio

$P = 64$ = Prescaler division ratio

$$f_{REF} = \frac{f_{CL}}{4 \times R} = \frac{4 \times 10^6}{4 \times 1024} = 976.5 \text{ Hz}$$

$$\frac{\Delta f_{VCO}}{\Delta N} = 976.5 \times 64 = 62.5 \text{ kHz}$$

FIGURE 21 — MC6805T2 PLL BLOCK DIAGRAM

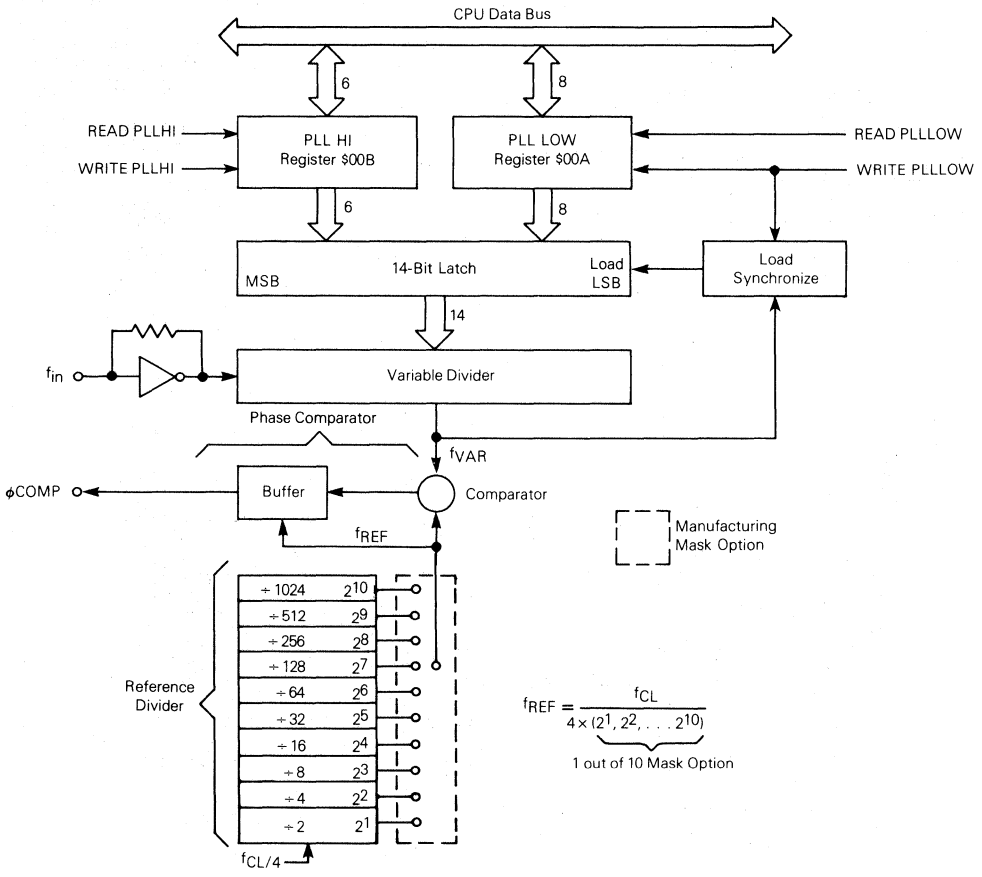


FIGURE 22 — TYPICAL FINE TUNE EXAMPLE

```
FTPLU   LDA     PLLLOW
        INCA                    check if LS byte = $FF (Reg $00A)
        BNE     TT1             if not increment only LS byte
        INC     PLLHI           increment MSB (Reg $00B) before LSB
TT1     INC     PLLLOW
        •
        •
        •
FTMIN   TST     PLLLOW          check if LS byte = $00
        BNE     TT2             if not decrement only LS byte
        DEC     PLLHI           decrement MSB before LSB
TT2     DEC     PLLLOW
        •
        •
        •
```

REFERENCE DIVIDER

This 10-stage binary counter generates a reference frequency, f_{REF}, which is compared to the output of the variable divider. The reference divider is mask programable, thus, allowing the user a choice of reference frequency, see Figure 21.

PHASE COMPARATOR

The phase comparator compares the frequency and phase of f_{VAR} and f_{REF}, and according to their phase relationship generates a three-level output (1, 0, Hi-Z), ϕCOMP, as shown in Figures 23 and 24. The output waveform is then integrated, amplified, and the resultant dc voltage is applied to the voltage controlled oscillator varicap.

In practice a linear characteristic around the steady-state region can not be achieved due to internal propagation delays. Thus, phase comparators exhibit non-linear characteristics and for systems which lock in phase, this results in a "backlash" effect – creating sidebands and FM distortion. To avoid this effect a very short pulse is injected periodically into the system. The loop, in turn, attempts to cancel this interference and in doing so brings the phase comparator to its linear zero, as shown in Figures 25 and 26.

A typical application, for a TV frequency synthesizer, is illustrated in Figure 27.

FIGURE 23 – PHASE COMPARATOR STATE DIAGRAM

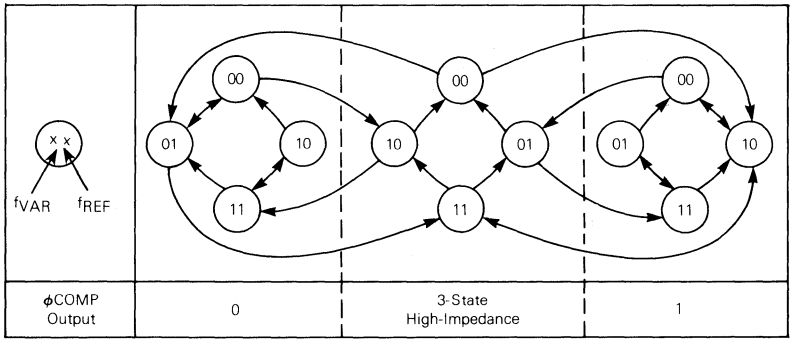


FIGURE 24 – PHASE COMPARATOR OUTPUT WAVEFORM

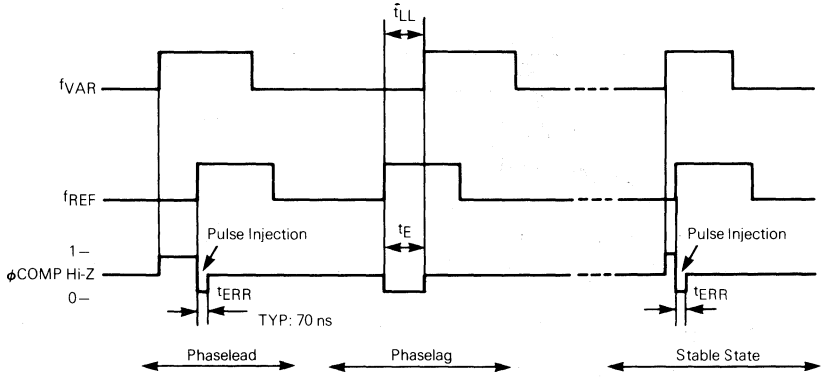


FIGURE 25 — PHASE COMPARATOR CHARACTERISTICS

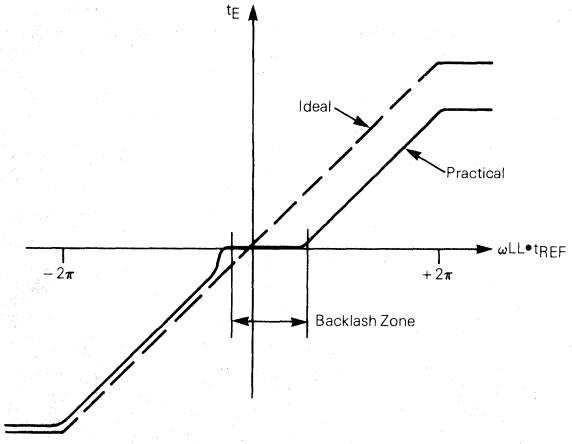


FIGURE 26 — PHASE COMPARATOR WITH PULSE INJECTION

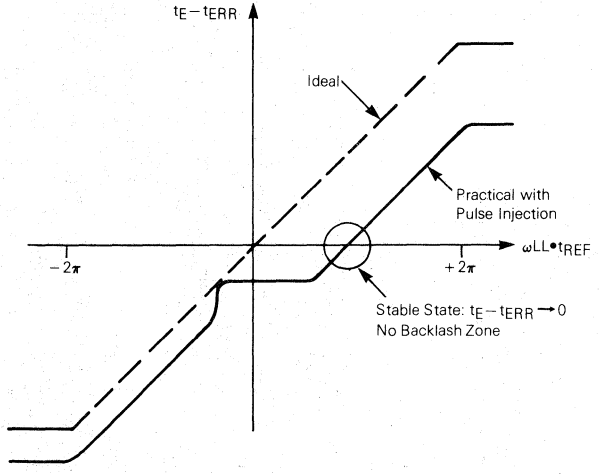
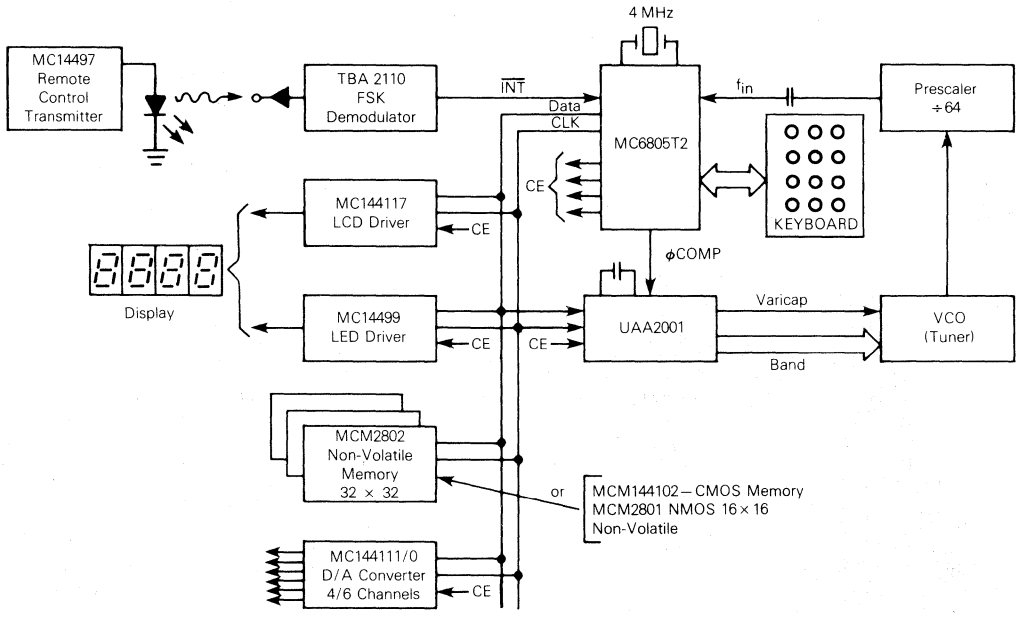


FIGURE 27 — A TYPICAL TV SYNTHESIZER APPLICATION

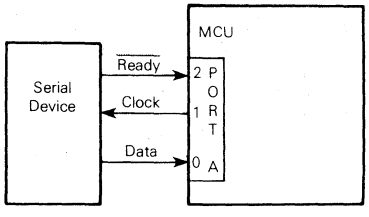


BIT MANIPULATION

The MCU has the ability to set or clear any single random-access memory or input/output bit (except the data direction registers, see Caution under **INPUT/OUTPUT**), with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. The carry bit equals the value of the bit referenced by BRSET and 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 RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines.

The coding example in Figure 28 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, LSB 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 bits in a RAM location.

FIGURE 28 — BIT MANIPULATION EXAMPLE




```
                    •
                    •
                    •
SELF      BRSET     2, PORTA, SELF
                    •
                    •
                    •
          BSET      1, PORTA
          BRCLR     0, PORTA, CONT
CONT      BCLR      1, PORTA
          ROR       RAMLOC
                    •
                    •
```

ADDRESSING MODES

The MC6805T2 has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family Users Manual.

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.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory (page zero) with a single 2-byte instruction. The address area includes the on-chip RAM and I/O registers and 128 bytes of ROM. Direct addressing is an effective use of both memory and time.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true; otherwise, control proceeds to the next instruction. The span of relative addressing is from − 126 to + 129 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.

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations (page zero). These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode the effective address is the sum of the contents of the unsigned 8-bit index register (X) and the unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE).

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset; except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended, the Motorola assembler determines the shortest form of indexed addressing.

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 page-zero address of the byte in which the specified bit is to be set or cleared. Thus, any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under **INPUT/OUTPUT**.

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, and condition (set or clear) is included in the opcode and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the value of the PC if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the Condition Code Register. See Caution under **INPUT/OUTPUT**.

INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MCU has a set of 59 basic instructions, which when combined with the 10 addressing modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 operand. Refer to Table 3.

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 (see Caution under **INPUT/OUTPUT**). The test for negative or zero (TST) instructions is included in the read/modify/write instructions though it does not perform the write. Refer to Table 4.

BRANCH INSTRUCTIONS

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 5.

BIT MANIPULATION INSTRUCTIONS

These instructions are used on any bit in the first 256 bytes of the memory (see Caution under **INPUT/OUTPUT**). One group either sets or clears. The other group performs the bit test branch operations. Refer to Table 6.

CONTROL INSTRUCTIONS

The control instructions control the MCU operations during program execution. Refer to Table 7.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 8.

OPCODE MAP SUMMARY

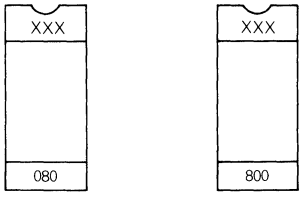
Table 9 is an opcode map for the instructions used on the MCU.

ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or on MDOS disk file.

To initiate a ROM pattern for the MCU it is necessary to first contact your local Motorola representative or distributor.

EPROMs – Two MCM2716 or one MCM2532 type EPROM(s), programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROM(s) must be clearly marked to indicate which EPROM(s) corresponds to which address space. The recommended marking procedure for two MCM2716 EPROMs is illustrated below.



XXX = Customer ID

After the EPROM(s) are marked they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

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 one blank EPROM from the data file used to create the custom mask to aid in the verification process.

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 5 volts. These RVUs are included in the mask charge and are not production parts. The RVUs thus are not guaranteed by Motorola Quality Assurance, and should be discarded after verification is completed.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS- compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The minimum MDOS system files as well as the absolute binary object file (filename, .LO type of file) from the M6805 cross assembler must be on the disk. An object file made from a memory dump using the ROLLOUT command is also acceptable. Consider submitting a source listing as well as the following files: filename, .LX(EXORciser loadable format) and filename, .SA (ASCII Source Code). These files will of course be kept confidential and are used 1) to speed up the process in-house if any problems arise, and 2) to speed up the user-to-factory interface if the user finds any software errors and needs assistance quickly from Motorola factory representatives.

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

TABLE 3 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	—	—		BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	—	—		BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 4 — READ/MODIFY/WRITE/ INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 5 — BRANCH INSTRUCTIONS

		Relative Addressing Mode		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 6 — BIT MANIPULATION INSTRUCTIONS

		Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0...7)	—	—	—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0...7)	—	—	—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0...7)	10 + 2 • n	2	7	—	—	—
Clear bit n	BCLR n (n = 0...7)	11 + 2 • n	2	7	—	—	—

TABLE 7 — CONTROL INSTRUCTIONS

		Inherent		
Function	Mnemonic	Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 8 — INSTRUCTION SET

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	↕
BRSET										X	•	•	•	•	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLL	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	↕	↕	↕
COM	X		X			X	X				•	•	↕	↕	1
CPX		X	X	X		X	X	X			•	•	↕	↕	↕
DEC	X		X			X	X				•	•	↕	↕	•
EOR		X	X	X		X	X	X			•	•	↕	↕	•
INC	X		X			X	X				•	•	↕	↕	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	↕	↕	•
LDX		X	X	X		X	X	X			•	•	↕	↕	•
LSL	X		X			X	X				•	•	↕	↕	↕
LSR	X		X			X	X				•	•	0	↕	↕
NEQ	X		X			X	X				•	•	↕	↕	↕
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	↕	↕	•
ROL	X		X			X	X				•	•	↕	↕	↕
RSP	X										•	•	•	•	•

Condition Code Symbols:

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ↕ Test and Set if True, Cleared Otherwise
- • Not Affected
- 0 Cleared

TABLE 8 — INSTRUCTION SET (Continued)

	Addressing Modes										Condition Code				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	^	^	^
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	^	^	●
STX			X	X		X	X	X			●	●	^	^	●
SUB		X	X	X		X	X	X			●	●	^	^	^
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	^	^	●
TXA	X										●	●	●	●	●

Condition Code Symbols:

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 1 Set

3

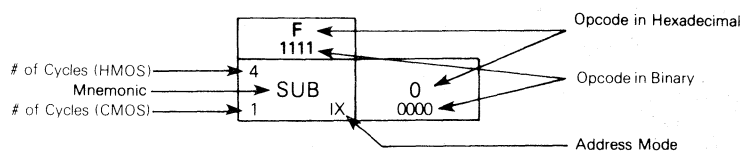
TABLE 9 — M6805 HMOS FAMILY OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



MC6805T2 MCU CUSTOM ORDERING INFORMATION

Date _____ Customer PO Number _____

Customer Company _____ Motorola Part Numbers

MC _____

Address _____

SC _____

City _____ State _____ Zip _____

Country _____

Phone _____ Extension _____

Customer Contact Person _____

Customer Part Number _____

OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information.

Timer Clock Source
- ☐ Internal $\phi 2$ clock (gated by PC0/TIMER)
- ☐ PC0/TIMER input pin (positive transition)

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only
- ☐ Open Drain

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

PLL Reference Divider
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)
- ☐ 2^8 (divide by 256)
- ☐ 2^9 (divide by 512)
- ☐ 2^{10} (divide by 1024)

Pattern Media (All other media requires prior factory approval.
- ☐ EPROMS (two MCM2716s or one MCM2532)
- ☐ Floppy Disk
- ☐ Other _____

Clock Freq. _____

Temp. Range _____ ☐ 0° to +70°C (Standard) ☐ −40° to +85°C*

*Requires prior factory approval

Marking Information (12 Characters Maximum)

Title _____

Signature _____

MOTOROLA

Advance Information

3

MC68HC05C4

8-BIT MICROCOMPUTER

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

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL

The MC68HC05C4 HCMOS Microcomputer is a member of the M68HC05 Family of low-cost single-chip microcomputers. This 8-bit microcomputer contains an on-chip oscillator, CPU, RAM, ROM, I/O, two serial interface sytems, and timer. The fully static design allows operation at frequencies down to dc, further reducing its already low-power consumption.

1.2 FEATURES

The following are some of the hardware and software highlights of the MC68HC05C4.

HARDWARE FEATURES

- HCMOS Technology
- 8-Bit Architecture
- Power Saving Stop and Wait Modes
- Fully Static Operation
- 176 Bytes of On-Chip RAM
- 4160 Bytes of On-Chip ROM
- 24 Bidirectional I/O Lines
- 2.1 MHz Internal Operating Frequency at 5 Volts; 1.0 MHz at 3 Volts
- Internal 16-Bit Timer Similar to MC6801 Timer
- Serial Communications Interface System
- Serial Peripheral Interface System
- Self-Check Mode
- External, Timer, Serial Communications Interface, and Serial Peripheral Interface Interrupts
- Master Reset and Power-On Reset
- Single 3- to 6-Volt Supply
- On-Chip Oscillator with RC or Crystal Mask Options
- 40-Pin Dual-In-Line Package
- Chip Carrier Also Available

SOFTWARE FEATURES

- Similar to MC6800
- 8 × 8 Unsigned Multiply Instruction
- Efficient Use of Program Space
- Versatile Interrupt Handling

SOFTWARE FEATURES (Continued)

- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power-Saving Standby Modes
- Upward Software Compatible with the M146805 CMOS Family
- Complete Development System Support on EXORciser and HDS-200

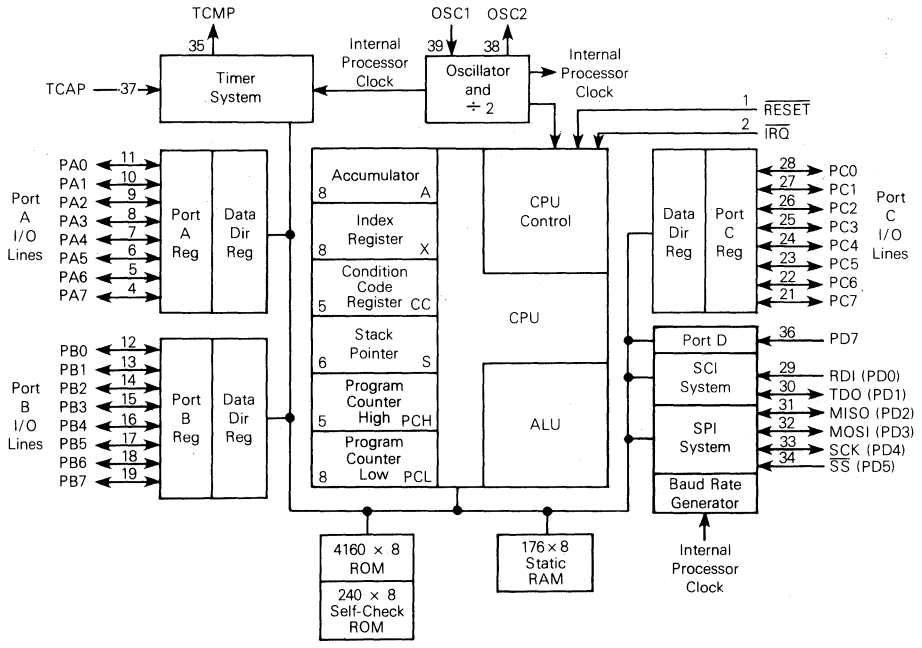


Figure 1-1. MC68HC05C4 Microcomputer Block Diagram

SECTION 2
FUNCTIONAL PIN DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, CPU REGISTERS, AND SELF-CHECK

This section provides a description of the functional pins, input/output programming, memory, CPU registers, and self-check.

2.1 FUNCTIONAL PIN DESCRIPTION

2.1.1 V_{DD} and V_{SS}

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

2.1.2 $\overline{IRQ}$ (Maskable Interrupt Request)

$\overline{IRQ}$ is a programmable option which provides two different choices of interrupt triggering sensitivity. These options are: 1) negative edge-sensitive triggering only, or 2) both negative edge-sensitive and level-sensitive triggering. In the latter case, either type of input to the $\overline{IRQ}$ pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When the $\overline{IRQ}$ pin goes low for at least on t_{ILIH}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

If the option is selected to include level-sensitive triggering, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See **INTERRUPTS** in Section 3 for more detail concerning interrupts.

2.1.3 $\overline{RESET}$

The $\overline{RESET}$ input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure. Refer to **RESETS** in Section 3 for a detailed description.

2.1.4 TCAP

The TCAP input controls the input capture feature for the on-chip programmable timer system. Refer to **INPUT CAPTURE REGISTER** in Section 4 for additional information.

2.1.5 TCMP

The TCMP pin (35) provides an output for the output compare feature of the on-chip timer system. Refer to **OUTPUT COMPARE REGISTER** in Section 4 for additional information.

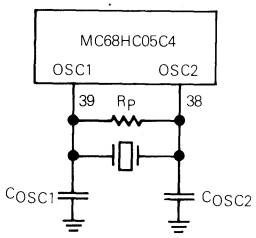
2.1.6 OSC1, OSC2

The MC68HC05C4 can be configured to accept either a crystal input or an RC network to control the internal oscillator. The internal clocks are derived by a divide-by-two of the internal oscillator frequency (f_{OSC}).

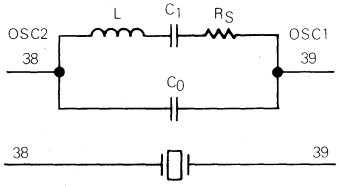
2.1.6.1 CRYSTAL. The circuit shown in Figure 2-1(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in 9.7 or 9.8 Control Timing. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to 9.5 or 9.6 for V_{DD} specifications.

	2 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

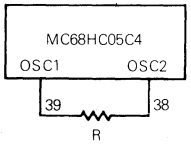
(a) Crystal Parameters



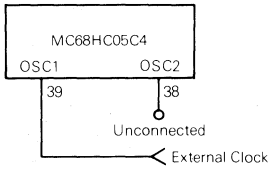
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit



(d) RC Oscillator Connections



(e) External Clock Source Connections

Figure 2-1. Oscillator Connections

2.1.6.2 RC. If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 2-1(d). The relationship between R and f_{OSC} is shown in Figure 2-2.

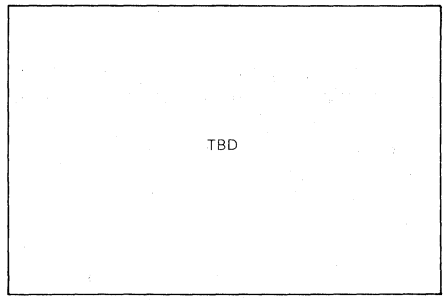


Figure 2-2. Typical Frequency vs Resistance For RC Oscillator Option Only

2.1.6.3 EXTERNAL CLOCK. An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 2-1(e). An external clock may be used with either the RC or crystal oscillator option. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OXOV} or t_{ILCH}.

2.1.7 PA0-PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. Refer to **INPUT/OUTPUT PROGRAMMING** paragraph below for a detailed description of I/O programming.

2.1.8 PB0-PB7

These eight lines comprise port B. The state of any pin is software programmable and all port B lines are configured as input during power-on or reset. Refer to **INPUT/OUTPUT PROGRAMMING** paragraph below for a detailed description of I/O programming.

2.1.9 PC0-PC7

These eight lines comprise port C. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset. Refer to **INPUT/OUTPUT PROGRAMMING** paragraph below for a detailed description of I/O programming.

2.1.10 PD0-PD5, PD7

These seven lines comprise port D, a fixed input port that is enabled during power-on. All enabled special functions (SPI and SCI) affect the pins on this port. Four of these lines, PD2/MISO, PD3/MOSI, PD4/SCK, and PD5/$\overline{SS}$, are used in the serial peripheral interface (SPI) discussed in

Section 6. Two of these lines, PD0/RDI and PD1/TDO, are used in the serial communications interface (SCI) discussed in Section 5. Refer to **2.2 INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming.

2.2 INPUT/OUTPUT PROGRAMMING

2.2.1 Parallel Ports

Ports A, B, and C may be programmed as an input or an output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each 8-bit port has an associated 8-bit data direction register. Any port A, port B, or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset, all DDRs are cleared, which configure all port A, B, and C pins as inputs. The data direction registers are capable of being written to or read by the processor. Refer to Figure 2-3 and Table 2-1. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

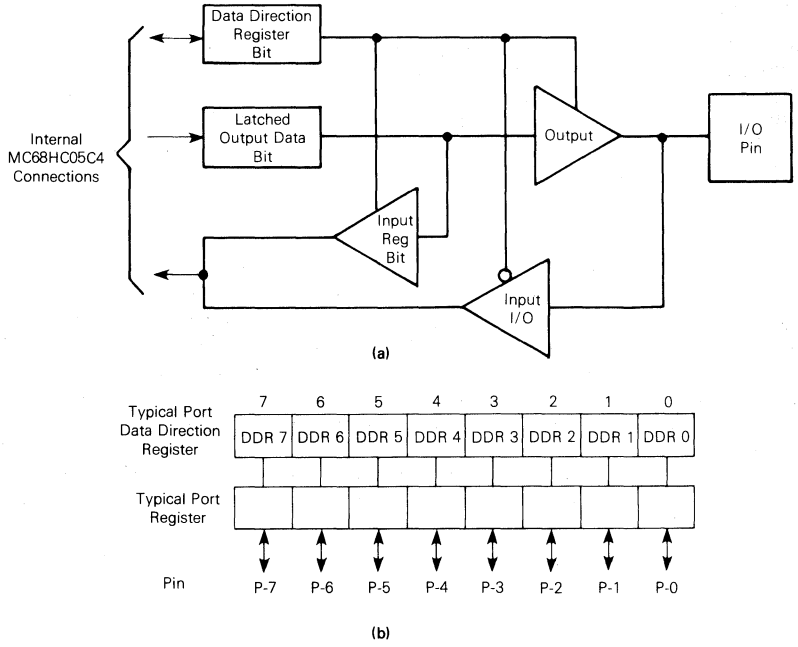


Figure 2-3. Typical Parallel Port I/O Circuitry

Table 2-1. I/O Pin Functions

R/$\overline{W}$*	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

2.2.2 Fixed Port

Port D is a 7-bit fixed input port (PD0-PD5, PD7) that continually monitors the external pins whenever the SPI or SCI systems are disabled. During power-on reset or external reset all seven bits become valid input ports because all special function output drivers are disabled. For example, with the serial communications interface (SCI) system enabled, (RE = TE = 1) PD0 and PD1 inputs will read zero. With the serial peripheral interface (SPI) system disabled (SPE = 0) PD2 through PD5 will read the state of the pin at the time of the read operation. No data register is associated with the port when it is used as an input.

2.2.3 Serial Port (SCI and SPI)

The serial communications interface (SCI) and serial peripheral interface (SPI) use the port D pins for their functions. The SCI function requires two of the pins (PD0-PD1) for its receive data input (RDI) and transmit data output (TDO) respectively, whereas the SPI function requires four of the pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), system clock (SCK), and slave select ($\overline{SS}$) respectively. Refer to **SECTION 5 SERIAL COMMUNICATIONS INTERFACE** and **SECTION 6 SERIAL PERIPHERAL INTERFACE** for a more detailed discussion.

2.3 MEMORY

As shown in Figure 2-4, the MCU is capable of addressing 8192 bytes of memory and I/O registers with its program counter. The MC68HC05C4 MCU has implemented 4601 bytes of these locations. The first 256 bytes of memory (page zero) include: 25 bytes of I/O features such as data ports, the port DDRs, timer, serial peripheral interface (SPI), and serial communication interface (SCI); 48 bytes of user ROM, and 176 bytes of RAM. The next 4096 bytes complete the user ROM. The self-check ROM (224 bytes) and self-check vectors (16 bytes) are contained in memory locations $1F00 through $1FEF. The 16 highest address bytes contain the user defined reset and the interrupt vectors. Seven bytes of the lowest 32 memory locations are unused and the 176 bytes of user RAM include up to 64 bytes for the stack. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

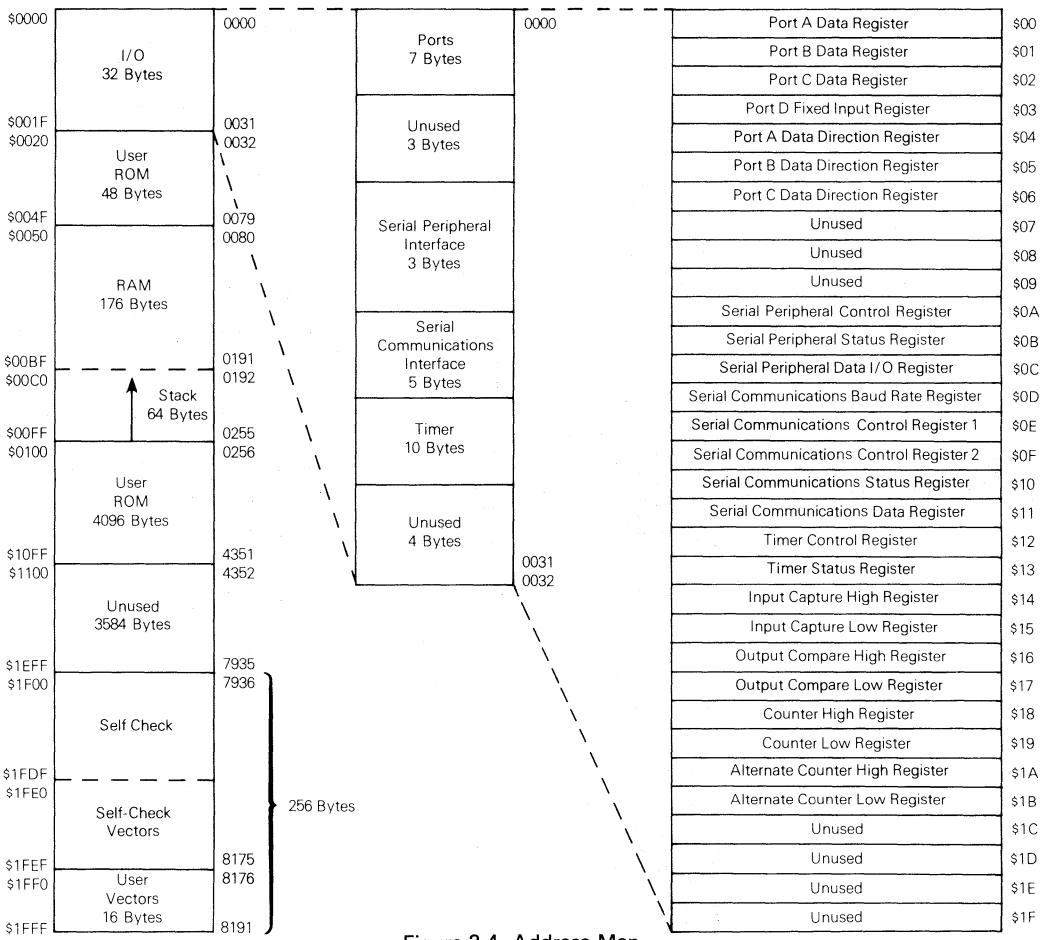


Figure 2-4. Address Map

2.4 CPU REGISTERS

The MC68HC05C4 CPU contains five registers, as shown in the programming model of Figure 2-5. The interrupt stacking order is shown in Figure 2-6.

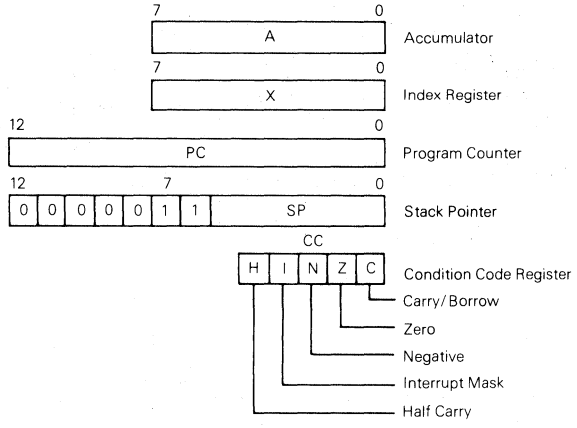
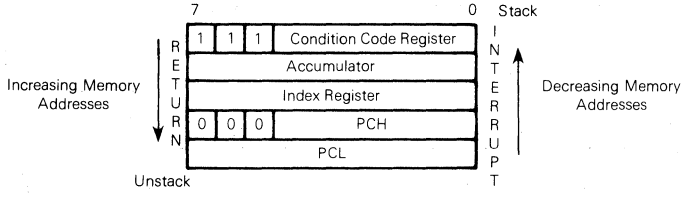


Figure 2-5. Programming Model



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 2-6. Stacking Order

2.4.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

2.4.2 Index Register (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

2.4.3 Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

2.4.4 Stack Pointer (SP)

The stack pointer is a 13-bit register containing the address of the next free locations on the push-down/pop-up stack. When accessing memory, the seven most significant bits are permanently configured to 0000011. These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($00FF). Nested interrupt and/or subroutines may use up to 64 (decimal) locations. When the 64 locations are exceeded, the stack pointer wraps around and points to its upper limit ($00FF), thus, losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

2.4.5 Condition Code Register (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. Each bit is explained in the following paragraphs.

2.4.5.1 HALF CARRY BIT (H). The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

2.4.5.2 INTERRUPT MASK BIT (I). When the I bit is set, all interrupts are disabled. Clearing this bit enables the interrupts. If an external interrupt occurs while the I bit is set, the interrupt is latched and is processed after the I bit is next cleared; therefore, no interrupts are lost because of the I bit being set. An internal interrupt can be lost if it is cleared while the I bit is set (refer to **SECTION 4 PROGRAMMABLE TIMER, SECTION 5 SERIAL COMMUNICATIONS INTERFACE,** and **SECTION 6 SERIAL PERIPHERAL INTERFACE** for more information).

2.4.5.3 NEGATIVE (N). When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

2.4.5.4 ZERO (Z). When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

2.4.5.5 CARRY/BORROW (C). Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

2.5 SELF-CHECK

The self-check capability of the MC68HC05C4 MCU provides an internal check to determine if the device is functional. Self-check is performed using the circuit shown in the schematic diagram of Figure 2-7. As shown in the diagram, port C pins PC0-PC3 are monitored (light emitting diodes are shown but other devices could be used) for the self-check results. The self-check mode is entered by applying a 9 Vdc input (through a 4.7 kilohm resistor) to the $\overline{IRQ}$ pin (2) and 5 Vdc input (through a 4.7 kilohm resistor) to the TCAP pin (37) and then depressing the reset switch to execute a reset. After reset, the following seven tests are performed automatically:

I/O — Functionally exercises ports A, B, and C
RAM — Counter test for each RAM byte
Timer — Tracks counter register and checks OCF flag
SCI — Transmission Test; checks for RDRF, TDRE, TC, and FE flags
ROM — Exclusive OR with odd ones parity result
SPI — Transmission test with check for SPIF, WCOL, and MODF flags
INTERRUPTS — Tests external, timer, SCI, and SPI interrupts.

Self-check results (using the LEDs as monitors) are shown in Table 2-2. The following subroutines are available to user programs and do not require any external hardware.

2.6 TIMER TEST SUBROUTINE

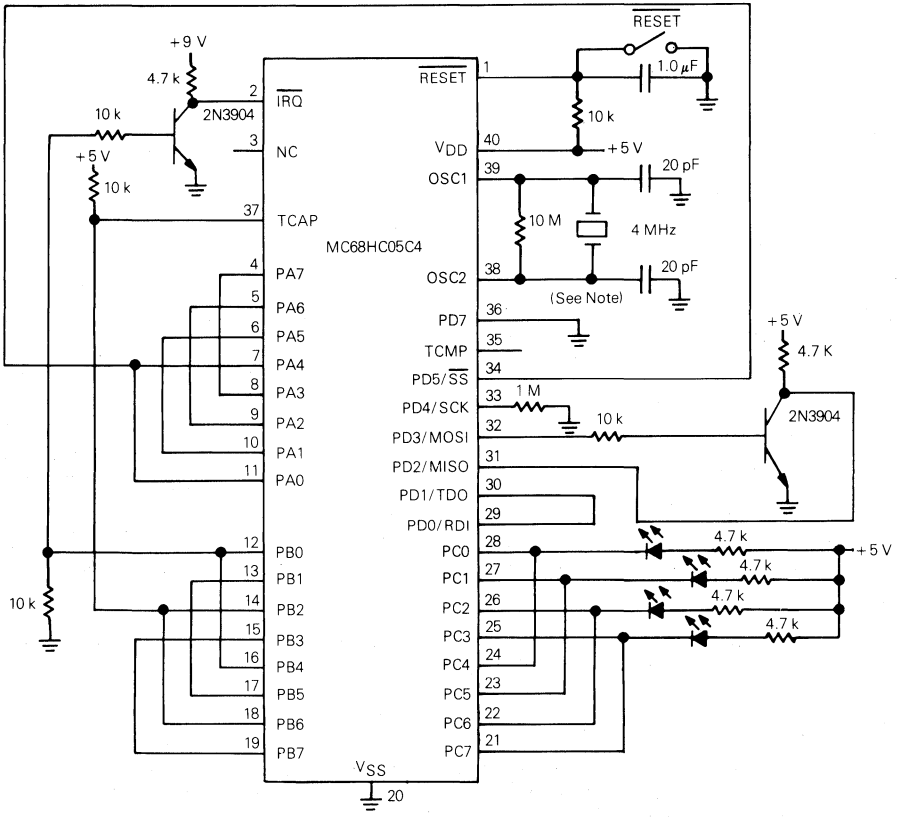
This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $1F0E. The output compare register is first set to the current timer state. Because the timer is free running and has only a divide-by-four prescaler, each timer count cannot be tested. The test reads the timer once every 10 counts (40 cycles) and checks for correct counting. The test tracks the counter until the timer wraps around, triggering the output compare flag in the timer status register. RAM locations $0050 and $0051 are overwritten. Upon return to the user's program, X = 40. If the test passed, A = 0.

2.7 ROM CHECKSUM SUBROUTINE

This subroutine returns with the Z bit cleared if any error is detected; otherwise, the Z bit is set.

This subroutine is called at location $1F93 with RAM location $0053 equal to $01 and A = 0. A short routine is set up and executed in RAM to compute a checksum of the entire ROM pattern. Upon return to the user's program, X = 0. If the test passed, A = 0. RAM locations $0050 through $0053 are overwritten.



NOTE: The RC Oscillator Option may also be used in this circuit.

Figure 2-7. Self-Check Circuit Schematic Diagram

Table 2-2. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad SPI
1	1	1	1	Bad Interrupts or $\overline{IRQ}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

0 Indicates LED on; 1 Indicates LED is off.

SECTION 3
RESETS, INTERRUPTS, AND LOW POWER MODES

3.1 RESETS

The MC68HC05C4 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 3-1.

3.1.1 $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one and one half t_{CYC}. The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.

3.1.2 Power-On Reset

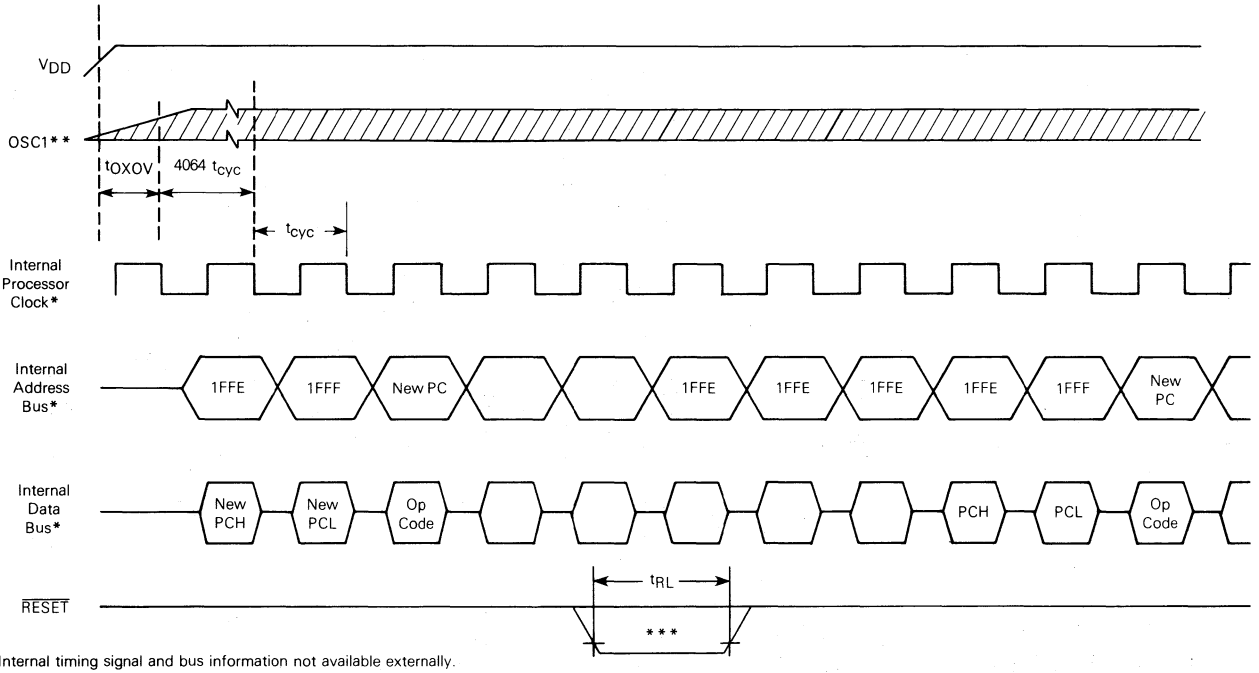
The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 4064 t_{CYC} delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 4064 t_{CYC} time out, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high.

Table 3-1 shows the actions of the two resets on internal circuits, but not necessarily in order of occurrence (X indicates that the condition occurs for the particular reset).

3.2 INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The MC68HC05C4 may be interrupted by one of five different methods: either one of four maskable hardware interrupts ($\overline{\text{IRQ}}$, SPI, SCI, or Timer) and one non-maskable software interrupt (SWI). Interrupts such as Timer, SPI, and SCI have several flags which will cause the interrupt. Generally, interrupt flags are located in read-only status registers, whereas their equivalent enable bits are located in associated control registers. The interrupt flags and enable bits are never contained in the same register. If the enable bit is a logic zero it blocks the interrupt from occurring but does not inhibit the flag from being set. Reset clears all enable bits to preclude interrupts during the reset procedure.

The general sequence for clearing an interrupt is a software sequence of first accessing the status register while the interrupt flag is set, followed by a read or write of an associated register. When any of these interrupts occur, and if the enable bit is a logic one, normal processing is suspended at the end of the current instruction execution. Interrupts cause the processor registers to be saved on



*Internal timing signal and bus information not available externally.
**OSC1 line is not meant to represent frequency. It is only used to represent time.
***The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 3-1. Power-On Reset and $\overline{\text{RESET}}$

Table 3-1. Reset Action on Internal Circuit

Condition	$\overline{\text{RESET}}$ Pin	Power-On Reset
Timer Prescaler reset to zero state	X	X
Timer counter configured to $FFFC	X	X
Timer output compare (TCMP) bit reset to zero	X	X
All timer interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts. The OLVL timer bit is also cleared by reset.	X	X
All data direction registers cleared to zero (input)	X	X
Configure stack pointer to $00FF	X	X
Force internal address bus to restart vector ($1FFE-$1FFF)	X	X
Set I bit in condition code register to a logic one	X	X
Clear STOP latch	X*	X
Clear external interrupt latch	X	X
Clear WAIT latch	X	X
Disable SCI (serial control bits TE=0 and RE=0). Other SCI bits cleared by reset include: TIE, TCIE, RIE, ILIE, RWU, SBK, RDRF, IDLE, OR, NF, and FE.	X	X
Disable SPI (serial output enable control bit SPE=0). Other SPI bits cleared by reset include: SPIE, MSTR, SPIF, WCOL, and MODF.	X	X
Set serial status bits TDRE and TC	X	X
Clear all serial interrupt enable bits (SPIE, TIE, and TCIE)	X	X
Place SPI system in slave mode (MSTR=0)	X	X
Clear SCI prescaler rate control bits SCP0-SCP1	X	X

*Indicates that timeout still occurs.

3

the stack (see Figure 2-6) and the interrupt mask (I bit) set to prevent additional interrupts. The appropriate interrupt vector then points to the starting address of the interrupt service routine (refer to Figure 2-4 for vector location). Upon completion of the interrupt service routine, the RTI instruction (which is normally a part of the service routine) causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 2-6.

NOTE

The interrupt mask bit (I bit) will be cleared if and only if the corresponding bit stored in the stack is zero.

A discussion of interrupts, plus a table listing vector addresses for all interrupts including reset, in the MC68HC05C4 is provided in Table 3-2.

Table 3-2. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{\text{RESET}}$	$1FFE-$1FFF
N/A	N/A	Software	SWI	$1FFC-$1FFD
N/A	N/A	External Interrupt	$\overline{\text{IRQ}}$	$1FFA-$1FFB
Timer Status	ICF	Input Capture	TIMER	$1FF8-$1FF9
	OCF	Output Compare		
	TOF	Timer Overflow		
SCI Status	TDRE	Transmit Buffer Empty	SCI	$1FF6-$1FF7
	TC	Transmit Complete		
	RDRF	Receiver Buffer Full		
	IDLE	Idle Line Detect		
	OR	Overrun		
SPI Status	SPIF	Transfer Complete	SPI	$1FF4-$1FF5
	MODF	Mode Fault		

3.2.1 Hardware Controlled Interrupt Sequence

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 3-2, and for STOP and WAIT are provided in Figure 3-3. A discussion is provided below.

(a) – A low input on the $\overline{\text{RESET}}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **RESETS** paragraph 3.1.

(b) STOP – The STOP instruction causes the oscillator to be turned off and the processor to "sleep" until an external interrupt ($\overline{\text{IRQ}}$) or reset occurs.

(c) WAIT – The WAIT instruction causes all processor clocks to stop, but leaves the Timer, SCI, and SPI clocks running. This "rest" state of the processor can be cleared by reset, an external interrupt ($\overline{\text{IRQ}}$), Timer interrupt, SPI interrupt, or SCI interrupt. There are no special wait vectors for these individual interrupts.

3.2.2 Software Interrupt (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask (I bit) in the condition code register. The interrupt service routine address is specified by the contents of memory location $1FFC and $1FFD.

3.2.3 External Interrupt

If the interrupt mask (I bit) of the condition code register has been cleared and the external interrupt pin ($\overline{\text{IRQ}}$) has gone low, then the external interrupt is recognized. When the interrupt is recognized, the current state of the CPU is pushed onto the stack and the I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FFA and $1FFB. Either a level-sensitive and negative edge-sensitive trigger, or a negative edge-sensitive only trigger are available as a mask option. Figure 3-4 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first method shows single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

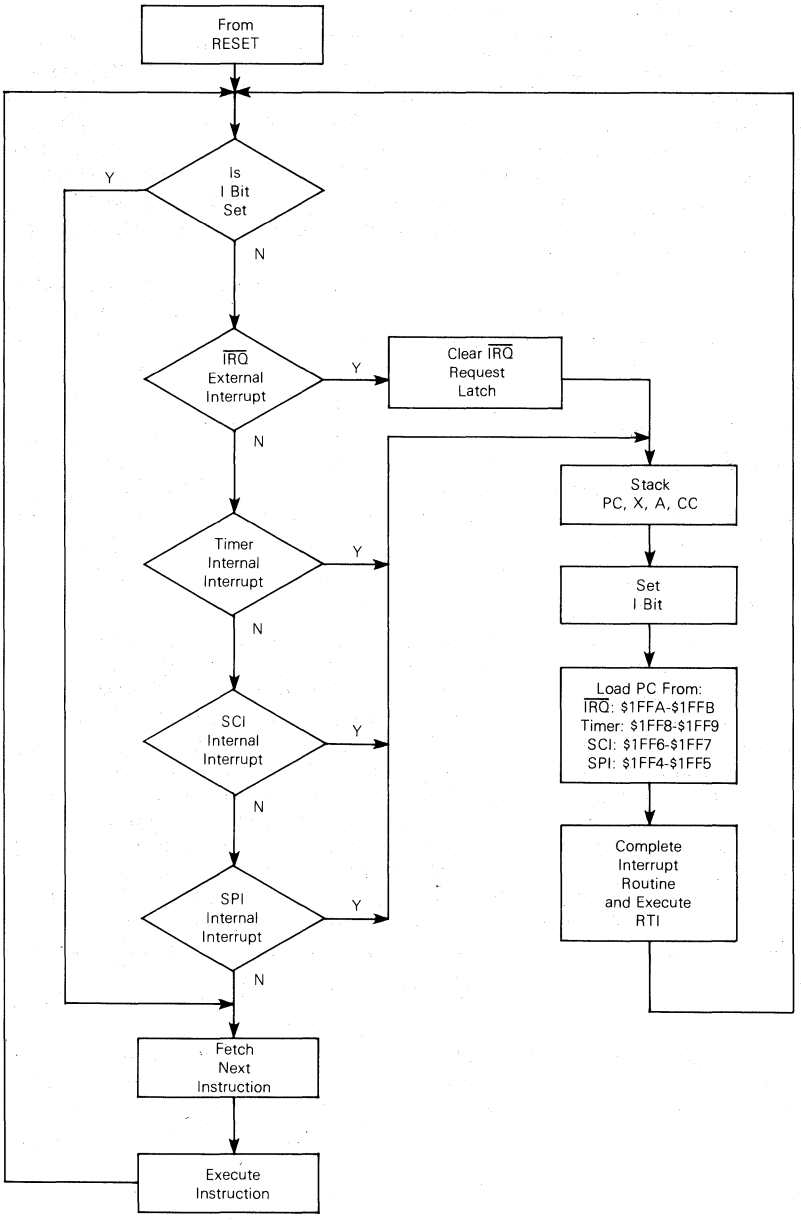


Figure 3-2. Hardware Interrupt Flowchart

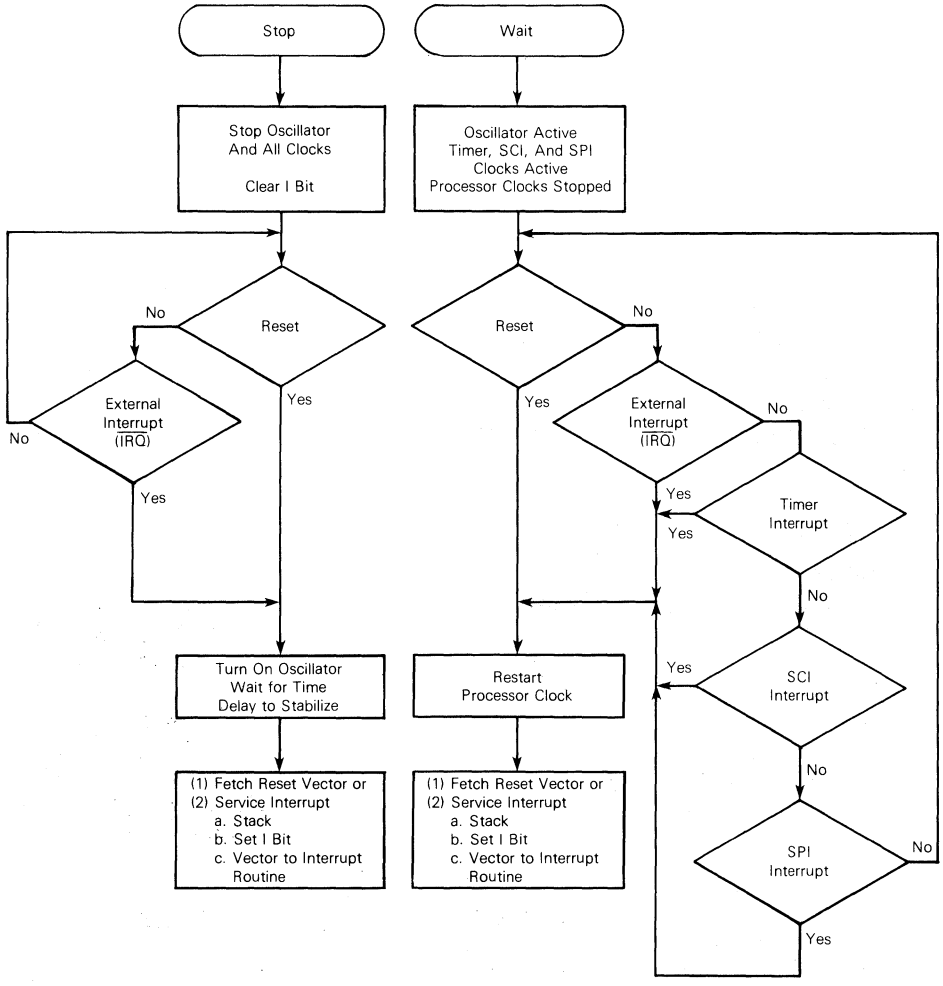
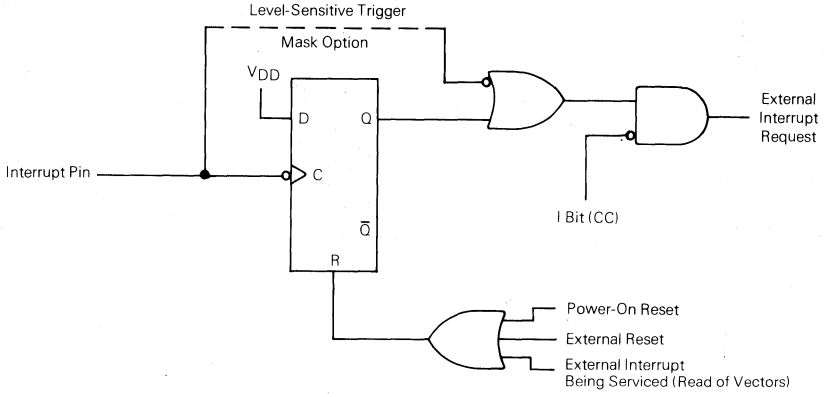
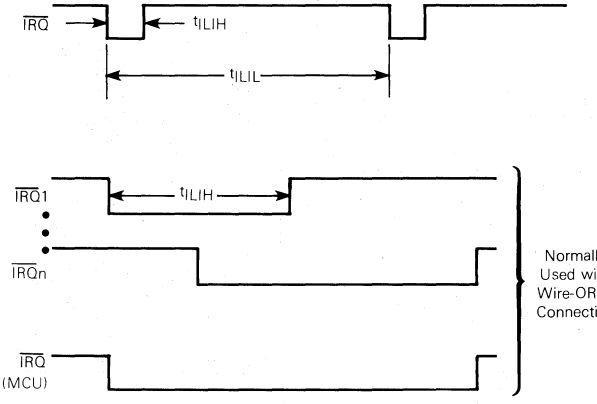


Figure 3-3. STOP/WAIT Flowcharts



(a) Interrupt Function Diagram



Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 21 t_{cyc} cycles.

Level-Sensitive Trigger Condition
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

(b) Interrupt Mode Diagram

Figure 3-4. External Interrupt

3.2.4 Timer Interrupt

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR, location $13) and all three will vector to the same interrupt service routine ($1FF8-$1FF9).

All interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) in the timer control register (TCR, location $12). Reset clears all enable bits, thus preventing an interrupt from occurring during the reset time period. The actual processor interrupt is generated only if the I bit in the condition code register is also cleared. When the interrupt is recognized, the current machine state is pushed onto the stack and I bit is set. This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FF8 and $1FF9. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to **SECTION 4 PROGRAMMABLE TIMER** for additional information about the timer circuitry.

3.2.5 Serial Communications Interface (SCI) Interrupts

An interrupt in the serial communications interface (SCI) occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the condition code register is clear and the enable bit in the serial communications control register 2 (location $0F) is enabled. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SCI interrupt causes the program counter to vector to memory location $1FF6 and $1FF7 which contains the starting address of the interrupt service routine. Software in the serial interrupt service routine must determine the priority and cause of the SCI interrupt by examining the interrupt flags and the status bits located in the serial communications status register (location $10). The general sequence for clearing an interrupt is a software sequence of accessing the serial communications status register while the flag is set followed by a read or write of an associated register. Refer to **SECTION 5 SERIAL COMMUNICATIONS INTERFACE** for a description of the SCI system and its interrupts.

3.2.6 Serial Peripheral Interface (SPI) Interrupts

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location $0B) is set, provided the I bit in the condition code register is clear and the enable bit in the serial peripheral control register (location $0A) is enabled. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location $1FF4 and $1FF5 which contains the starting address of the interrupt service routine. Software in the serial peripheral interrupt service routine must determine the priority and cause of the SPI interrupt by examining the interrupt flag bits located in the SPI status register. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to **SECTION 6 SERIAL PERIPHERAL INTERFACE** for a description of the SPI system and its interrupts.

3.3 LOW POWER MODES

3.3.1 STOP Instruction

The STOP instruction places the MC68HC05C4 in its lowest power consumption mode. In the STOP mode the internal oscillator is turned off, causing all internal processing to be halted; refer to Figure 3-3. During the STOP mode, the I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered and all input/output lines remain unchanged. This continues until an external interrupt ($\overline{IRQ}$) or reset is sensed at which time the internal oscillator is turned on. The external interrupt or reset causes the program counter to vector to memory location $1FFA and $1FFB or $1FFE and $1FFF which contains the starting address of the interrupt or reset service routine respectively.

3.3.2 WAIT Instruction

The WAIT instruction places the MC68HC05C4 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock remains active, and all CPU processing is stopped; however, the programmable timer, serial peripheral interface, and serial communications interface systems remain active. Refer to Figure 3-3. During the WAIT mode, the I bit in the condition code register is cleared to enable all interrupts. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged. This continues until any interrupt or reset is sensed. At this time the program counter vectors to the memory location ($1FF4 through $1FFF) which contains the starting address of the interrupt or reset service routine.

SECTION 4
PROGRAMMABLE TIMER

4.1 INTRODUCTION

The programmable timer, which is preceded by a fixed divide-by-four prescaler, can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 4-1 and timing diagrams are shown in Figures 4-2 through 4-5.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the condition code register should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur. This prevents interrupts from occurring between the time that the high and low bytes are accessed.

The programmable timer capabilities are provided by using the following ten addressable 8-bit registers (note the high and low represent the significance of the byte). A description of each register is provided below.

Timer Control Register (TCR) location $12,
Timer Status Register (TSR) location $13,
Input Capture High Register location $14,
Input Capture Low Register location $15,
Output Compare High Register location $16,
Output Compare Low Register location $17,
Counter High Register location $18,
Counter Low Register location $19,
Alternate Counter High Register location $1A, and
Alternate Counter Low Register location $1B.

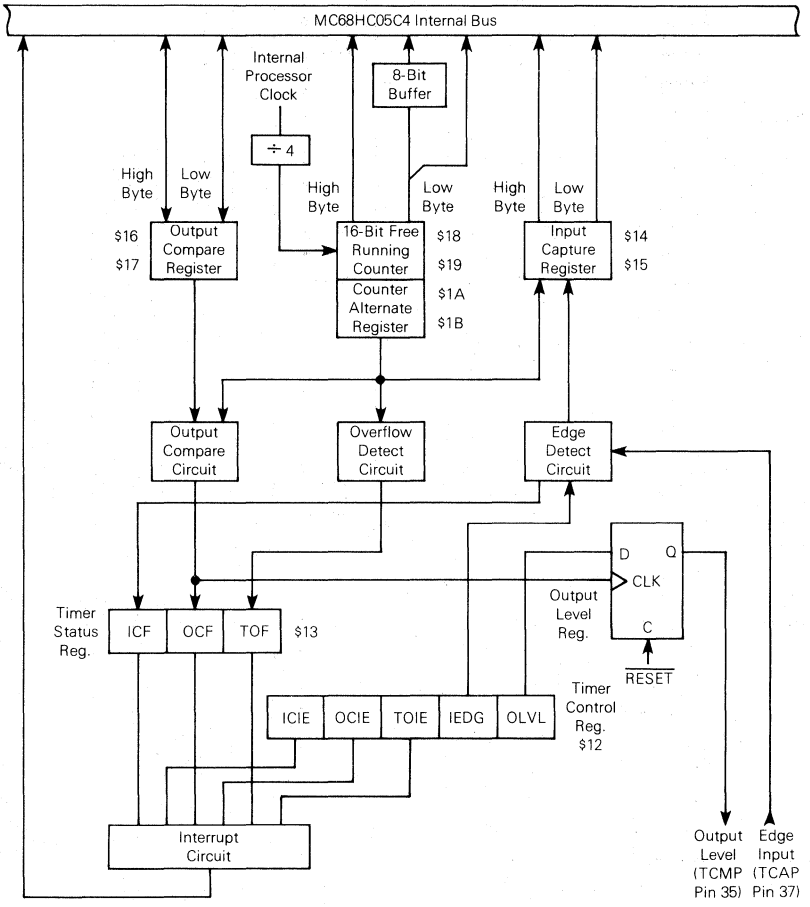
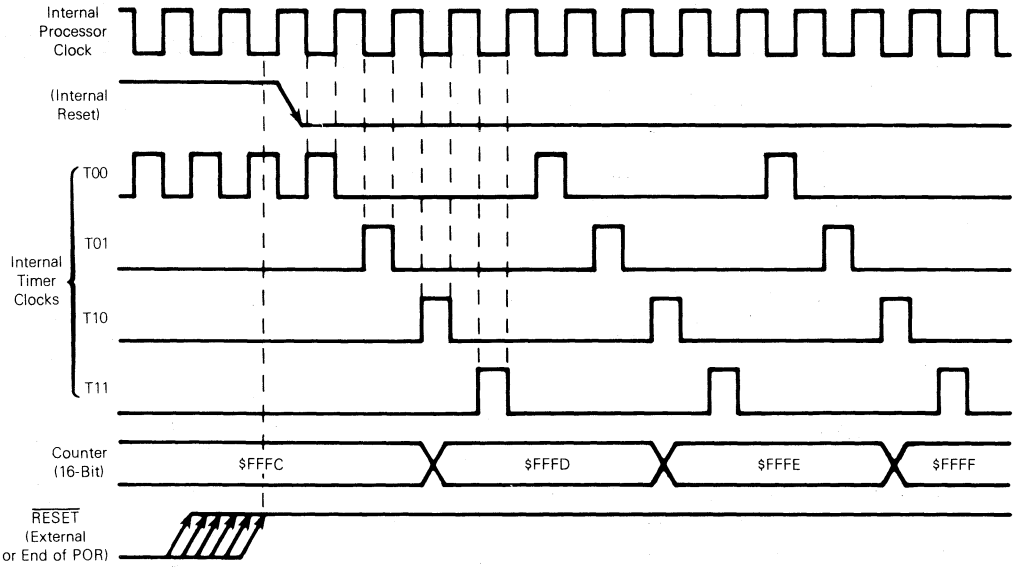
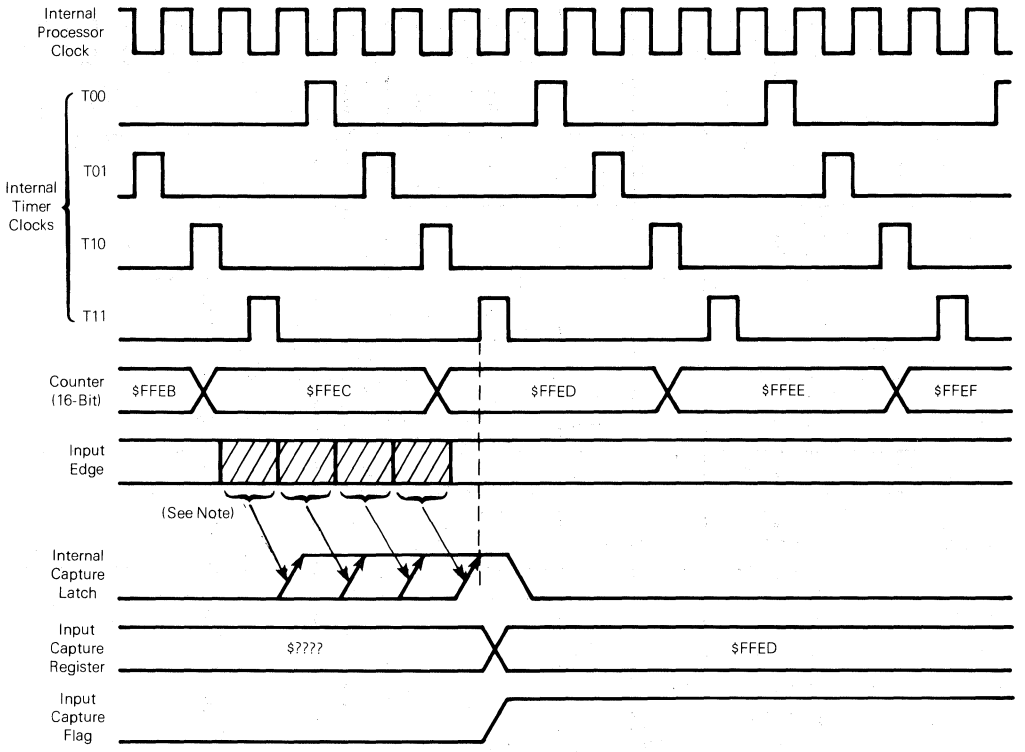


Figure 4-1. Programmable Timer Block Diagram



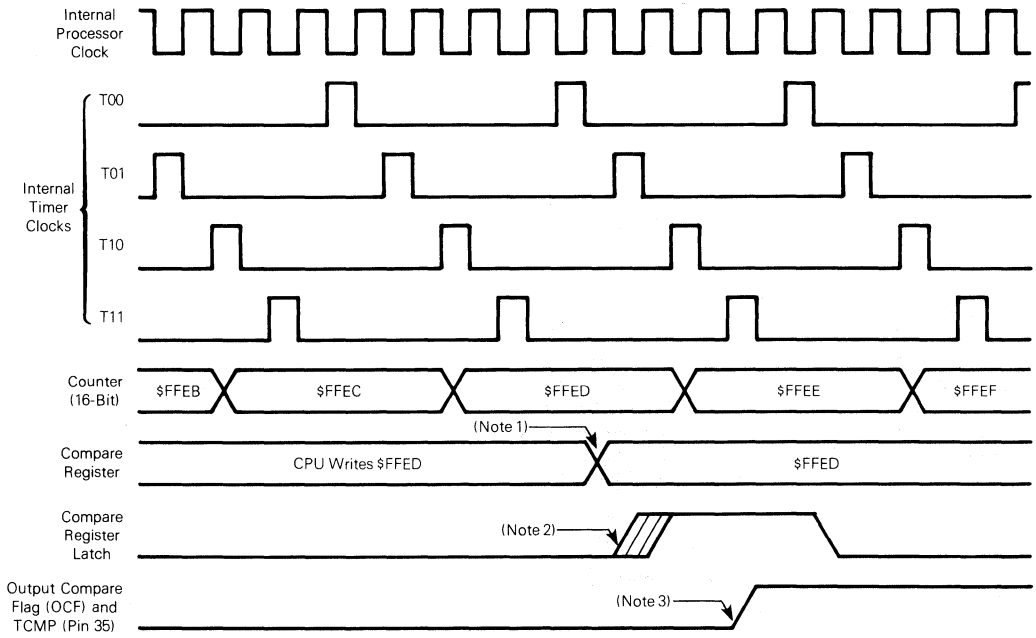
NOTE: The Counter Register and Timer Control Register are the only ones affected by $\overline{\text{RESET}}$.

Figure 4-2. Timer State Timing Diagram For Reset



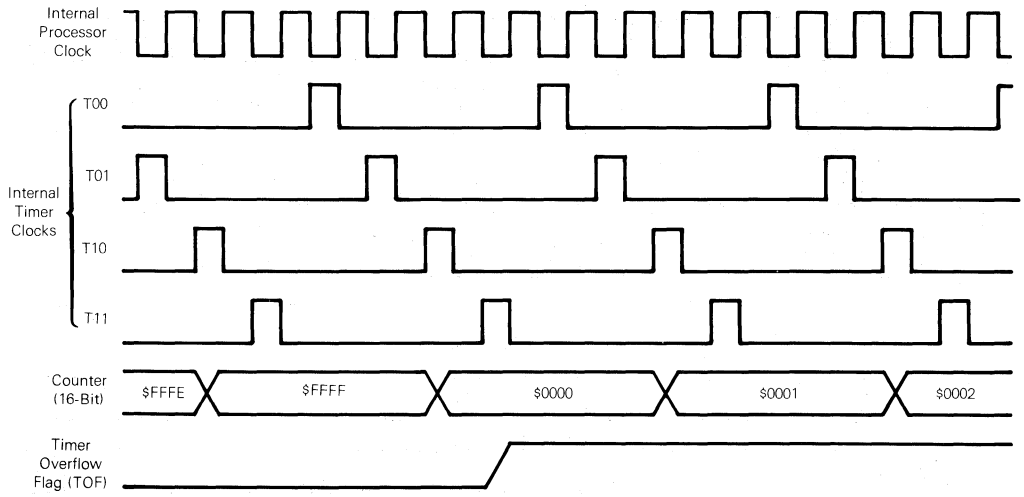
NOTE: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

Figure 4-3. Timer State Timing Diagram For Input Capture



NOTES: 1. The CPU write to the compare register may take place at any time, but a compare only occurs at timer state T01. Thus, a 4-cycle difference may exist between the write to the compare register and the actual compare.
2. Internal compare takes place during timer state T01.
3. OCF is set at the timer state T11 which follows the comparison match ($FFED in this example).

Figure 4-4. Timer State Timing Diagram For Output Compare



NOTE: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by a read of the timer status register during the internal processor clock high time followed by a read of the counter low register.

Figure 4-5. Timer State Diagram For Timer Overflow

4.2 COUNTER

The key element in the programmable timer is a 16-bit free running counter, or counter register, preceded by a prescaler which divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal processor clock is 2.0 MHz. The counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting its value.

The double byte free running counter can be read from either of two locations $18-$19 (called counter register at this location), or $1A-$1B (counter alternate register at this location). A read sequence containing only a read of the least significant byte of the free running counter ($19,$1B) will receive the count value at the time of the read. If a read of the free running counter or counter alternate register first addresses the most significant byte ($18,$1A) it causes the least significant byte ($19,$1B) to be transferred to a buffer. This buffer value remains fixed after the first most significant byte "read" even if the user reads the most significant byte several times. This buffer is accessed when reading the free running counter or counter alternate register least significant byte ($19 or $1B), and thus completes a read sequence of the total counter value. Note that in reading either the free running counter or counter alternate register, if the most significant byte is read, the least significant byte must also be read in order to complete the sequence.

The free running counter is configured to $FFFC during reset and is always a read-only register. During a power-on-reset (POR), the counter is also configured to $FFFC and begins running after the oscillator startup delay. Because the free running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free running counter repeats every 262,144 MPU internal processor clock cycles. When the counter rolls over from $FFFF to $0000, the timer overflow flag (TOF) bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

4.3 OUTPUT COMPARE REGISTER

The output compare register is a 16-bit register, which is made up of two 8-bit registers at locations $16 (most significant byte) and $17 (least significant byte). The output compare register can be used for several purposes such as, controlling an output waveform or indicating when a period of time has elapsed. The output compare register is unique in that all bits are readable and writable and are not altered by the timer hardware. Reset does not affect the contents of this register and if the compare function is not utilized, the two bytes of the output compare register can be used as storage locations.

The contents of the output compare register are compared with the contents of the free running counter once during every four internal processor clocks. If a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked (by the output compare circuit pulse) to an output level register. The values in the output compare register and the output level bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit, OCIE, is set.

After a processor write cycle to the output compare register containing the most significant byte ($16), the output compare function is inhibited until the least significant byte ($17) is also written. The user must write both bytes (locations) if the most significant byte is written first. A write made

only to the least significant byte ($17) will not inhibit the compare function. The free running counter is updated every four internal processor clock cycles due to the internal prescaler. The minimum time required to update the output compare register is a function of the software program rather than the internal hardware.

A processor write may be made to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

Because neither the output compare flag (OCF bit) or output compare register is affected by reset, care must be exercised when initializing the output compare function with software. The following procedure is recommended:

(1) Write the high byte of the output compare register to inhibit further compares until the low byte is written.
(2) Read the timer status register to arm the OCF if it is already set.
(3) Write the output compare register low byte to enable the output compare function with the flag clear.

The advantage of this procedure is to prevent the OCF bit from being set between the time it is read and the write to the output compare register. A software example is shown below.

```
B7    16    STA   OCMPHI   INHIBIT OUTPUT COMPARE
B6    13    LDA   TSTAT    ARM OCF BIT IF SET
BF    17    STX   OCMPLD   READY FOR NEXT COMPARE
```

4.4 INPUT CAPTURE REGISTER

The two 8-bit registers which make up the 16-bit input capture register are read-only and are used to latch the value of the free running counter after a defined transition is sensed by the corresponding input capture edge detector. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free running counter on the rising edge of the internal processor clock preceding the external transition (refer to timing diagram shown in Figure 4-3). This delay is required for internal synchronization. Resolution is affected by the prescaler allowing the timer to only increment every four internal processor clock cycles.

The free running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free running counter value which corresponds to the most recent input capture.

After a read of the most significant byte of the input capture register ($14), counter transfer is inhibited until the least significant byte ($15) of the input capture register is also read. This characteristic forces the minimum pulse period attainable to be determined by the time used in the capture software routine and its interaction with the main program. The free running counter increments every four internal processor clock cycles due to the prescaler.

A read of the least significant byte ($15) of the input capture register does not inhibit the free running counter transfer. Again, minimum pulse periods are ones which allow software to read the least significant byte ($15) and perform needed operations. There is no conflict between the read of the input capture register and the free running counter transfer since they occur on opposite edges of the internal processor clock.

4.5 TIMER CONTROL REGISTER (TCR)

The timer control register (TCR, location $12) is an 8-bit read/write register which contains five control bits. Three of these bits control interrupts associated with each of the three flag bits found in the timer status register (discussed below). The other two bits control: 1) which edge is significant to the input capture edge detector (i.e., negative or positive), and 2) the next value to be clocked to the output level register in response to a successful output compare. The timer control register and the free running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes it to a high. The timer control register is illustrated below followed by a definition of each bit.

7	6	5	4	3	2	1	0	
ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	$12

B7, ICIE — If the input capture interrupt enable (ICIE) bit is set, a timer interrupt is enabled when the ICF status flag (in the timer status register) is set. If the ICIE bit is clear, the interrupt is inhibited. The ICIE bit is cleared by reset.

B6, OCIE — If the output compare interrupt enable (OCIE) bit is set, a timer interrupt is enabled whenever the OCF status flag is set. If the OCIE bit is clear, the interrupt is inhibited. The OCIE bit is cleared by reset.

B5, TOIE — If the timer overflow interrupt enable (TOIE) bit is set, a timer interrupt is enabled whenever the TOF status flag (in the timer status register) is set. If the TOIE bit is clear, the interrupt is inhibited. The TOIE bit is cleared by reset.

B1, IEDG — The value of the input edge (IEDG) bit determines which level transition on pin 37 will trigger a free running counter transfer to the input capture register. Reset does not affect the IEDG bit.

0 = negative edge
1 = positive edge

B0, OLVL — The value of the output level (OLVL) bit is clocked into the output level register by the next successful output compare and will appear at pin 35. This bit and the output level register are cleared by reset.

0 = low output
1 = high output

4.6 TIMER STATUS REGISTER (TSR)

The timer status register (TSR) is an 8-bit register of which the three most significant bits contain read-only status information. These three bits indicate the following:

1. A proper transition has taken place at pin 37 with an accompanying transfer of the free running counter contents to the input capture register,
2. A match has been found between the free running counter and the output compare register, and
3. A free running counter transition from $FFFF to $0000 has been sensed (timer overflow).

The timer status register is illustrated below followed by a definition of each bit. Refer to timing diagrams shown in Figures 4-2, 4-3, and 4-4 for timing relationship to the timer status register bits.

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	0	0	0	0	0	$13

B7, ICF — The input capture flag (ICF) is set when a proper edge has been sensed by the input capture edge detector. It is cleared by a processor access of the timer status register (with ICF set) followed by accessing the low byte ($15) of the input capture register. Reset does not affect the input compare flag.

B6, OCF — The output compare flag (OCF) is set when the output compare register contents matches the contents of the free running counter. The OCF is cleared by accessing the timer status register (with OCF set) and then accessing the low byte ($17) of the output compare register. Reset does not affect the output compare flag.

B5, TOF — The timer overflow flag (TOF) bit is set by a transition of the free running counter from $FFFF to $0000. It is cleared by accessing the timer status register (with TOF set) followed by an access of the free running counter least significant byte ($19). Reset does not affect the TOF bit.

Accessing the timer status register satisfies the first condition required to clear any status bits which happen to be set during the access. The only remaining step is to provide an access of the register which is associated with the status bit. Typically, this presents no problem for the input capture and output compare functions.

A problem can occur when using the timer overflow function and reading the free running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if: 1) the timer status register is read or written when TOF is set, and 2) the least significant byte of the free running counter is read but not for the purpose of servicing the flag. The counter alternate register at address $1A and $1B contains the same value as the free running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

During STOP and WAIT instructions, the programmable timer functions as follows: during the wait mode, the timer continues to operate normally and may generate an interrupt to trigger the CPU out of the wait state; during the stop mode, the timer holds at its current state, retaining all data, and resumes operation from this point when an external interrupt is received.

SECTION 5
SERIAL COMMUNICATIONS INTERFACE (SCI)

5.1 INTRODUCTION

A full-duplex asynchronous serial communications interface (SCI) is provided with a standard NRZ format and a variety of baud rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. The serial data format is standard mark/space (NRZ) which provide one start bit, eight or nine data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

5.1.1 SCI Two Wire System Features

- Standard NRZ (mark/space) format.
- Advanced error detection method includes noise detection for noise duration of up to 1/16 bit time.
- Full-duplex operation (simultaneous transmit and receive).
- Software programmable for one of 32 different baud rates.
- Software selectable world length (eight or nine bit words).
- Separate transmitter and receiver enable bits.
- SCI may be interrupt driven.
- Four separate enable bits available for interrupt control.

5.1.2 SCI Receiver Features

- Receiver wake-up function (idle or address bit).
- Idle line detect.
- Framing error detect.
- Noise detect.
- Overrun detect.
- Receiver data register full flag.

5.1.3 SCI Transmitter Features

- Transmit data register empty flag.
- Transmit complete flag.
- Break send.

Any SCI two-wire system requires receive data in (RDI) and transmit data out (TDO).

5.2 DATA FORMAT

Receive data in (RDI) or transmit data out (TDO) is the serial data which is presented between the internal data bus and the output pin (TDO), and between the input pin (RDI) and the internal data bus. Data format is as shown for the NRZ in Figure 5-1 and must meet the following criteria:

1. A high level indicates a logic one and a low level indicates a logic zero.
2. The idle line is in a high (logic one) state prior to transmission/reception of a message.
3. A start bit (logic zero) is transmitted/received indicating the start of a message.
4. The data is transmitted and received least-significant-bit first.
5. A stop bit (high in the tenth or eleventh bit position) indicates the byte is complete.
6. A break is defined as the transmission or reception of a low (logic zero) for some multiple of the data format.

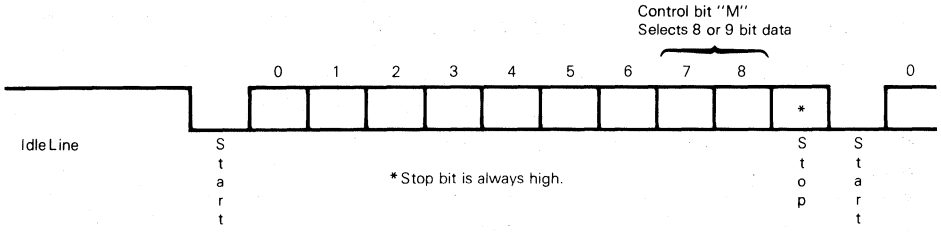


Figure 5-1. Data Format

5.3 WAKE-UP FEATURE

In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line returns to the idle state. An SCI receiver is re-enabled by an idle string of at least ten (or eleven) consecutive ones. Software for the transmitter must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

The user is allowed a second method of providing the wake-up feature in lieu of the idle string discussed above. This method allows the user to insert a logic one in the most significant bit of the transmit data word which needs to be received by all "sleeping" processors.

5.4 RECEIVE DATA IN

Receive data in is the serial data which is presented from the input pin via the SCI to the internal data bus. While waiting for a start bit, the receiver samples the input at a rate which is 16 times higher than the set baud rate. This 16 times higher-than-baud rate is referred to as the RT rate in Figures 5-2 and 5-3, and as the receiver clock in Figure 5-7. When the input (idle) line is detected low, it is tested for three more sample times (referred to as the start edge verification samples in Figure 5-2). If at least two of these three verification samples detect a logic low, a valid start bit is assumed to have been detected (by a logic low following the three start qualifiers) as shown in Figure 5-2; however, if in two or more of the verification samples a logic high is detected, the line is

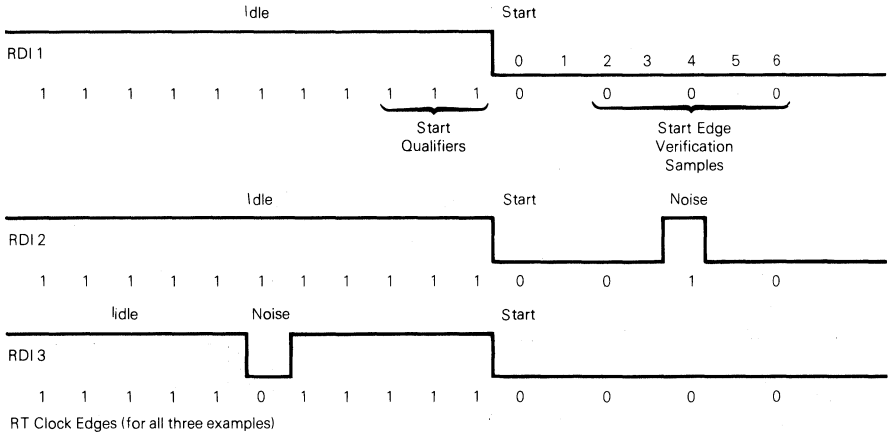


Figure 5-2. Examples of Start Bit Sampling Technique

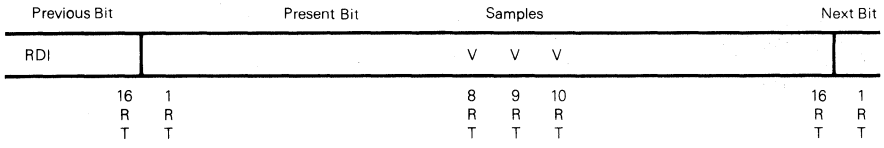


Figure 5-3. Sampling Technique Used on All Bits

assumed to be idle. (A noise flag is set if one of the three verification samples detects a logic high, thus a valid start bit could be assumed and a noise flag still set.) The receiver clock generator is controlled by the baud rate register (see Figures 5-6 and 5-7); however, the serial communications interface is synchronized by the start bit (independent of the transmitter).

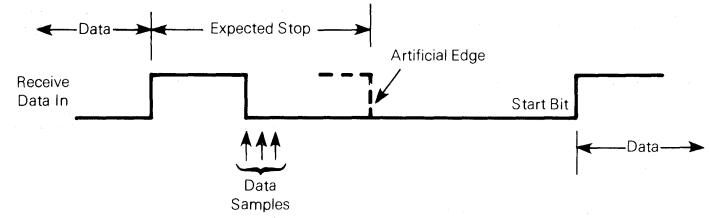
Once a valid start bit is detected, the start bit, each data bit, and the stop bit are sampled three times at RT intervals of 8RT, 9RT, and 10RT (1RT is the position where the bit is expected to start) as shown in Figure 5-3. The value of the bit is determined by voting logic which takes the value of the majority of samples (two or three out of three). A noise flag is set when all three samples on a valid start bit or a data bit or the stop bit do not agree. (As discussed above, a noise flag is also set when the start bit verification samples do not agree.)

5.5 START BIT DETECTION FOLLOWING A FRAMING ERROR

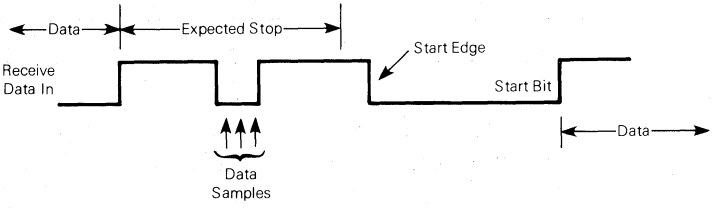
If there has been a framing error without detection of a break (10 zeros for 8-bit format or 11 zeros for 9-bit format), the circuit continues to operate as if there actually were a stop bit and the start

edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers (shown in Figure 5-2) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 5-4); therefore the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced and the receiver must actually receive a logic one bit before start. See Figure 5-5.



(a) Case 1, Receive Line Low During Artificial Edge



(b) Case 2, Receive Line High During Expected Start Edge

Figure 5-4. SCI Artificial Start Following A Framing Error

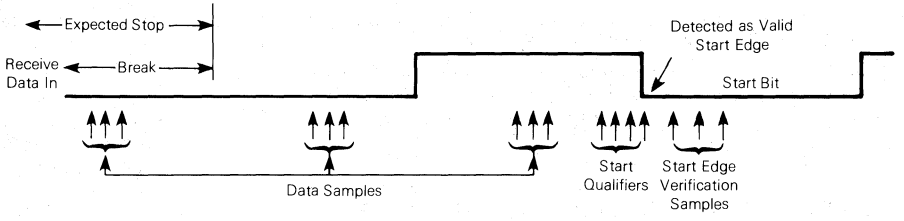


Figure 5-5. SCI Start Bit Following A Break

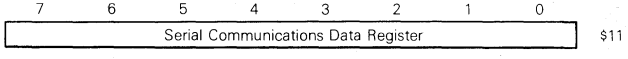
5.6 TRANSMIT DATA OUT (TDO)

Transmit data out is the serial data which is presented from the internal data bus via the SCI and then to the output pin. Data format is as discussed above and shown in Figure 5-1. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16 that of the receiver sample clock.

5.7 REGISTERS

There are five different registers used in the serial communications interface (SCI) and the internal configuration of these registers is discussed in the following paragraphs. A block diagram of the SCI system is shown in Figure 5-6.

5.7.1 Serial Communications Data Register (SCDAT)

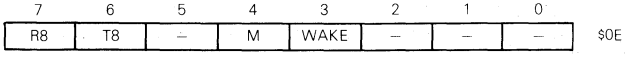


The serial communications data register performs two functions in the serial communications interface; i.e. it acts as the receive data register when it is read and as the transmit data register when it is written. Figure 5-6 shows this register as two separate registers, namely: the receive data register (RDR) and the transmit data register (TDR). As shown in Figure 5-6, the TDR (transmit data register) provides the parallel interface from the internal data bus to the transmit shift register and the receive data register (RDR) provides the interface from the receive shift register to the internal data bus.

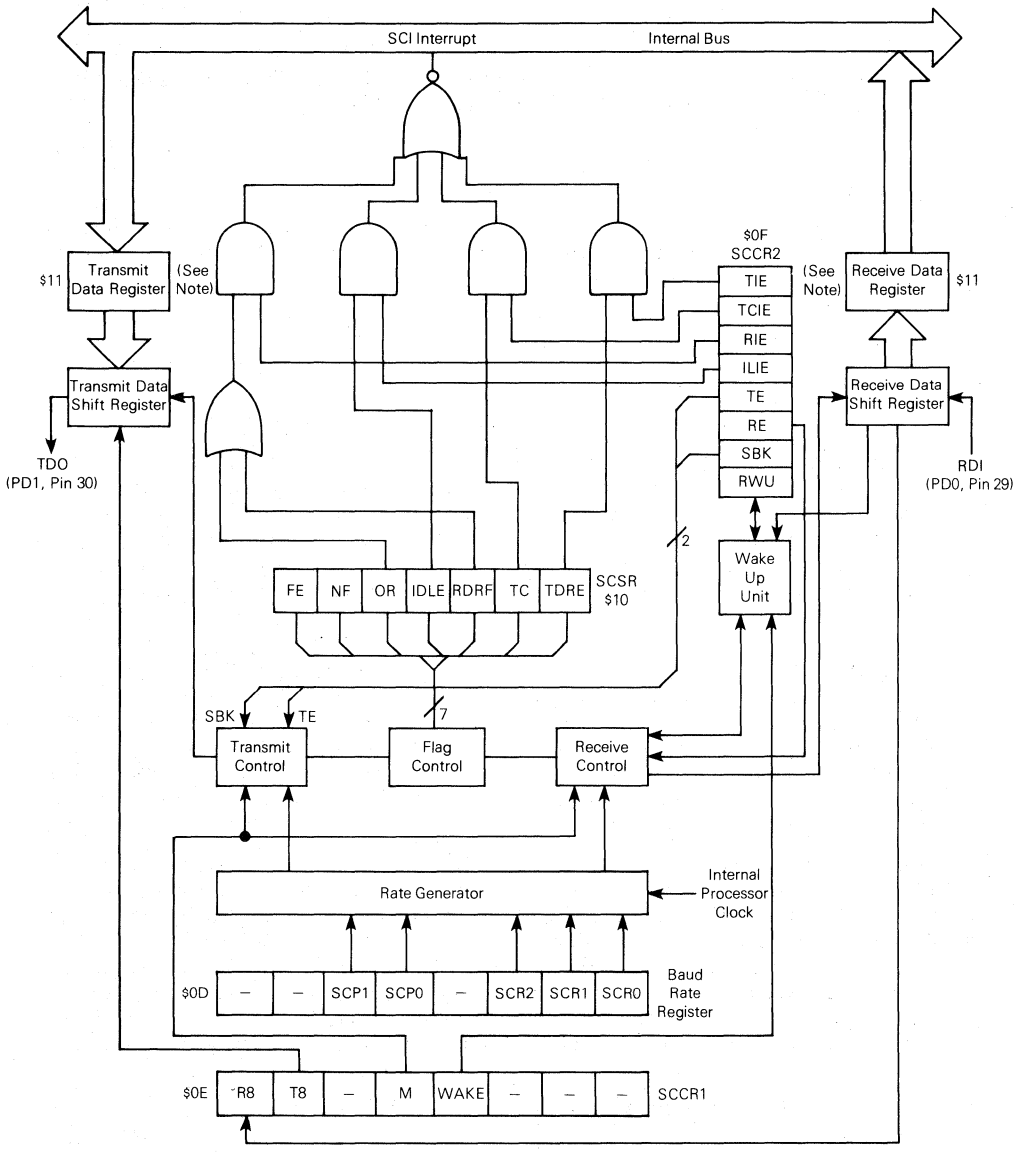
When SCDAT is read, it becomes the receive data register and contains the last byte of data received. The receive data register, represented above, is a read-only register containing the last byte of data received from the shift register for the internal data bus. The RDRF bit (receive data register full bit in the serial communications status register) is set to indicate that a byte has been transferred from the input serial shift register to the serial communications data register. The transfer is synchronized with the receiver bit rate clock (from the receive control) as shown in Figure 5-6. All data is received least-significant-bit first.

When SCDAT is written, it becomes the transmit data register and contains the next byte of data to be transmitted. The transmit data register, also represented above, is a write-only register containing the next byte of data to be applied to the transmit shift register from the internal data bus. As long as the transmitter is enabled, data stored in the serial communications data register is transferred to the transmit shift register (after the current byte in the shift register has been transmitted). The transfer from the SCDAT to the transmit shift register is synchronized with the bit rate clock (from the transmit control) as shown in Figure 5-6. All data is transmitted least-significant-bit first.

5.7.2 Serial Communications Control Register 1 (SCCR1)



The serial communications control register 1 (SCCR1) provides the control bits which: 1) determine the word length (either 8 or 9 bits), and 2) selects the method used for the wake-up feature. Bits 6 and 7 provide a location for storing the ninth bit for longer bytes.



NOTE: The Serial Communications Data Register (SCDAT) is controlled by the internal R/W signal. It is the transmit data register when written and receive data register when read.

Figure 5-6. Serial Communications Interface Block Diagram

B7, R8 — If the M bit is a one, then this bit provides a storage location for the ninth bit in the receive data byte. Reset does not affect this bit.

B6, T8 — If the M bit is a one, then this bit provides a storage location for the ninth bit in the transmit data byte. Reset does not affect this bit.

B4, M — The option of the word length is selected by the configuration of this bit and is shown below. Reset does not affect this bit.

0 = 1 start bit, 8 data bits, 1 stop bit
1 = 1 start bit, 9 data bits, 1 stop bit

B3, WAKE — This bit allows the user to select the method for receiver "wake up". If the WAKE bit is a logic zero, an idle line condition will "wake up" the receiver. If the WAKE bit is set to a logic one, the system acknowledges an address bit (most significant bit). The address bit is dependent on both the WAKE bit and the M bit level (table shown below). (Additionally, the receiver does not use the wake-up feature unless the RWU control bit in serial communications control register 2 is set as discussed below.) Reset does not affect this bit.

Wake	M	Method of Receiver "Wake-Up"
0	X	Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

5.7.3 Serial Communications Control Register 2 (SCCR2)

7	6	5	4	3	2	1	0	
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	$0F

The serial communications control register 2 (SCCR2) provides the control bits which: individually enable/disable the transmitter or receiver, enable the system interrupts, and provide the wake-up enable bit and a "send break code" bit. Each of these bits is described below. (The individual flags are discussed in the **5.7.4 Serial Communications Status Register**.)

B7, TIE — When the transmit interrupt enable bit is set, the SCI interrupt occurs provided TDRE is set (see Figure 5-6). When TIE is clear, the TDRE interrupt is disabled. Reset clears the TIE bit.

B6, TCIE — When the transmission complete interrupt enable bit is set, the SCI interrupt occurs provided TC is set (see Figure 5-6). When TCIE is clear, the TC interrupt is disabled. Reset clears the TCIE bit.

B5, RIE — When the receive interrupt enable bit is set, the SCI interrupt occurs provided OR is set or RDRF is set (see Figure 5-6). When RIE is clear, the OR and RDRF interrupts are disabled. Reset clears the RIE bit.

B4, ILIE — When the idle line interrupt enable bit is set, the SCI interrupt occurs provided IDLE is set (see Figure 5-6). When ILIE is clear, the IDLE interrupt is disabled. Reset clears the ILIE bit.

B3, TE — When the transmit enable bit is set, the transmit shift register output is applied to the TDO line. Depending on the state of control bit M in serial communications control register 1, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. If a transmission is in progress, and TE is written to a zero, then the transmitter will wait until after the present byte has been transmitted before placing the TDO pin in the idle high-impedance state. If the TE bit has been written to a zero and then set to a one before the current byte is transmitted, the transmitter will wait until that byte is transmitted and will then initiate transmission of a new preamble. After the preamble is transmitted, and provided the TDRE bit is set (no new data to transmit), the line remains idle (driven high while TE = 1); otherwise, normal transmission occurs. This function allows the user to "neatly" terminate a transmission sequence. After loading the last byte in the serial communications data register and receiving the interrupt from TDRE, indicating the data has been transferred into the shift register, the user should clear TE. The last byte will then be transmitted and the line will go idle (high impedance). Reset clears the TE bit.

B2, RE — When the receive enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all of the status bits associated with the receiver (RDRF, IDLE, OR, NF, and FE) are inhibited. Reset clears the RE bit.

B1, RWU — When the receiver wake-up bit is set, it enables the "wake up" function. The type of "wake up" mode for the receiver is determined by the WAKE bit discussed above (in the SCCR1). When the RWU bit is set, no status flags will be set. Flags which were set previously will not be cleared when RWU is set. If the WAKE bit is cleared, RWU is cleared after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. Under these conditions, RWU cannot be set if the line is idle. If the WAKE bit is set, RWU is cleared after receiving an address bit. The RDRF flag will then be set and the address byte will be stored in the receiver data register. Reset clears the RWU bit.

B0, SBK — When the send break bit is set the transmitter sends zeros in some number equal to a multiple of the data format bits. If the SBK bit is toggled set and clear, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or sending data. The actual number of zeros sent when SBK is toggled depends on the data format set by the M bit in the serial communications control register 1; therefore, the break code will be synchronous with respect to the data stream. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. Reset clears the SBK bit.

5.7.4 Serial Communications Status Register (SCSR)

7	6	5	4	3	2	1	0	
TDRE	TC	RDRF	IDLE	OR	NF	FE	–	$10

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt. In addition, a noise flag bit and a framing error bit are also contained in the SCSR.

B7, TDRE — The transmit data register empty bit is set to indicate that the contents of the serial communications data register have been transferred to the transmit serial shift register. If the TDRE bit is clear, it indicates that the transfer has not yet occurred and a write to the serial communications data register will overwrite the previous value. The TDRE bit is cleared by accessing the serial communications status register (with TDRE set), followed by writing to the serial communications data register. Data can not be transmitted unless the serial communications status register is accessed before writing to the serial communications data register to clear the TDRE flag bit. Reset sets the TDRE bit.

B6, TC — The transmit complete bit is set at the end of a data frame, preamble, or break condition if:

1. TE=1, TDRE=1, and no pending data, preamble, or break is to be transmitted; or
2. TE=0, and the data, preamble, or break (in the transmit shift register) has been transmitted.

The TC bit is a status flag which indicates that one of the above conditions has occurred. The TC bit is cleared by accessing the serial communications status register (with TC set), followed by writing to the serial communications data register. It does not inhibit the transmitter function in any way. Reset sets the TC bit.

B5, RDRF — When the receive data register full bit is set, it indicates that the receiver serial shift register is transferred to the serial communications data register. If multiple errors are detected in any one received word, the NF, FE, and RDRF bits will be affected as appropriate during the same clock cycle. The RDRF bit is cleared when the serial communications status register is accessed (with RDRF set) followed by a read of the serial communications data register. Reset clears the RDRF bit.

B4, IDLE — When the idle line detect bit is set, it indicates that a receiver idle line is detected (receipt of a minimum number of ones to constitute the number of bits in the byte format). The minimum number of ones needed will be 10 (M=0) or 11 (M=1). This allows a receiver that is not in the wake-up mode to detect the end of a message, detect the preamble of a new message, or to resynchronize with the transmitter. The IDLE bit is cleared by accessing the serial communications status register (with IDLE set) followed by a read of the serial communications data register. The IDLE bit will not be set again until

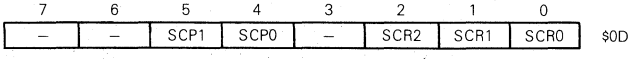
after an RDRF has been set; i.e., a new idle line occurs. The IDLE bit is not set by an idle line when the receiver "wakes up" from the wake-up mode. Reset clears the IDLE bit.

B3, OR — When the overrun error bit is set, it indicates that the next byte is ready to be transferred from the receive shift register to the serial communications data register when it is already full (RDRF bit is set). Data transfer is then inhibited until the RDRF bit is cleared. Data in the serial communications data register is valid in this case, but additional data received during an overrun condition (including the byte causing the overrun) will be lost. The OR bit is cleared when the serial communications status register is accessed (with OR set), followed by a read of the serial communications data register. Reset clears the OR bit.

B2, NF — The noise flag bit is set if there is noise on a "valid" start bit or if there is noise on any of the data bits or if there is noise on the stop bit. It is not set by noise on the idle line nor by invalid (false) start bits. If there is noise, the NF bit is not set until the RDRF flag is set. Each data bit is sampled three times as described above in RECEIVE DATA IN and shown in Figure 5-3. The NF bit represents the status of the byte in the serial communications data register. For the byte being received (shifted in) there will also be a "working" noise flag the value of which will be transferred to the NF bit when the serial data is loaded into the serial communications data register. The NF bit does not generate an interrupt because the RDRF bit gets set with NF and can be used to generate the interrupt. The NF bit is cleared when the serial communications status register is accessed (with NF set), followed by a read of the serial communications data register. Reset clears the NF bit.

B1, FE — The framing error bit is set when the byte boundaries in the bit stream are not synchronized with the receiver bit counter (generated by a "lost" stop bit). The byte is transferred to the serial communications data register and the RDRF bit is set. The FE bit does not generate an interrupt because the RDRF bit is set at the same time as FE and can be used to generate the interrupt. Note that if the byte received causes a framing error and it will also cause an overrun if transferred to the serial communications data register, then the overrun bit will be set, but not the framing error bit, and the byte will not be transferred to the serial communications data register. The FE bit is cleared when the serial communications status register is accessed (with FE set) followed by a read of the serial communications data register. Reset clears the FE bit.

5.7.5 Baud Rate Register



The baud rate register provides the means for selecting different baud rates which may be used as the rate control for the transmitter and receiver. The SCP0-SCP1 bits function as a prescaler for the

SCR0-SCR2 bits. Together, these five bits provide multiple, baud rate combinations for a given crystal frequency.

B5, SCP1
B4, SCP0

These two bits in the baud rate register are used as a prescaler to increase the range of standard baud rates controlled by the SCR0-SCR2 bits. A table of the prescaler internal processor clock division versus bit levels is provided below. Reset clears SCP1-SCP0 bits (divide-by-one).

SCP1	SCP0	Internal Processor Clock Divide By
0	0	1
0	1	3
1	0	4
1	1	13

B2, SCR2
B1, SCR1
B0, SCR0

These three bits in the baud rate register are used to select the baud rates of both the transmitter and receiver. A table of baud rates versus bit levels is shown below. Reset does not affect the SCR2-SCR0 bits.

SCR2	SCR1	SCR0	Prescaler Output Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The diagram of Figure 5-7 and Tables 5-1 and 5-2 illustrate the divided chain used to obtain the baud rate clock (transmit clock). Note that there is a fixed rate divide-by-16 between the receive clock (RT) and the transmit clock (Tx). The actual divider chain is controlled by the combined SCP0-SCP1 and SCR0-SCR2 bits in the baud rate register as illustrated. All divided frequencies shown in the first table represent the final transmit clock (the actual baud rate) resulting from the internal processor clock division shown in the "divide-by" column only (prescaler division only). The second table illustrates how the prescaler output can be further divided by action of the SCI select bits (SCR0-SCR2). For example, assume that a 9600 Hz baud rate is required with a 2.4576 MHz external crystal. In this case the prescaler bits (SCP0-SCP1) could be configured as a divide-by-one or a divide-by-four. If a divide-by-four prescaler is used, then the SCR0-SCR2 bits must be configured as a divide-by-two. This results in a divide-by-128 of the internal processor clock to produce a 9600 Hz baud rate clock. Using the same crystal, the 9600 baud rate can be obtained with a prescaler divide-by-one and the SCR0-SCR2 bits configured for a divide-by-eight.

NOTE

The crystal frequency is internally divided-by-two to generate the internal processor clock.

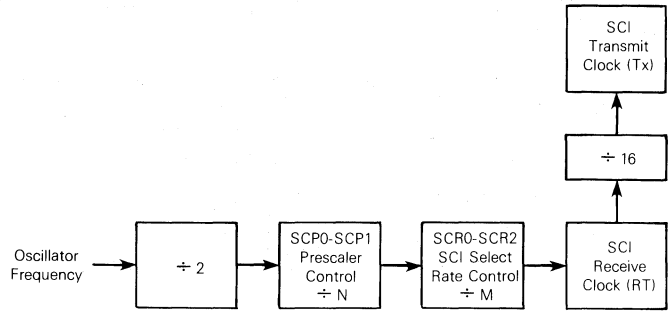


Figure 5-7. Rate Generator Division

Table 5-1. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock*	Crystal Frequency MHz				
1	0	Divided By	4.194304	4.0	2.4576	2.0	1.8432
0	0	1	131.072 kHz	125.000 kHz	76.80 kHz	62.50 kHz	57.60 kHz
0	1	3	43.691 kHz	41.666 kHz	25.60 kHz	20.833 kHz	19.20 kHz
1	0	4	32.768 kHz	31.250 kHz	19.20 kHz	15.625 kHz	14.40 kHz
1	1	13	10.082 kHz	9600 Hz	5.907 kHz	4800 Hz	4430 Hz

*The clock in the "Clock Divided By" column is the internal processor clock.

NOTE: The divided frequencies shown in Table 5-1 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs.

Table 5-2. Transmit Baud Rate Output For a Given Prescaler Output

SCR Bits			Divide	Representative Highest Prescaler Baud Rate Output				
2	1	0	By	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	0	1	131.072 kHz	32.768 kHz	76.80 kHz	19.20 kHz	9600 Hz
0	0	1	2	65.536 kHz	16.384 kHz	38.40 kHz	9600 Hz	4800 Hz
0	1	0	4	32.768 kHz	8.192 kHz	19.20 kHz	4800 Hz	2400 Hz
0	1	1	8	16.384 kHz	4.096 kHz	9600 Hz	2400 Hz	1200 Hz
1	0	0	16	8.192 kHz	2.048 kHz	4800 Hz	1200 Hz	600 Hz
1	0	1	32	4.096 kHz	1.024 kHz	2400 Hz	600 Hz	300 Hz
1	1	0	64	2.048 kHz	512 Hz	1200 Hz	300 Hz	150 Hz
1	1	1	128	1.024 kHz	256 Hz	600 Hz	150 Hz	75 Hz

NOTE: Table 5-2 illustrates how the SCI select bits can be used to provide lower transmitter baud rates by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receiver clock is 16 times higher in frequency than the actual baud rate.

SECTION 6
SERIAL PERIPHERAL INTERFACE (SPI)

6.1 INTRODUCTION AND FEATURES

6.1.1 Introduction

The serial peripheral interface (SPI) is an interface built into the MC68HC05C4 MCU which allows several MC68HC05C4 MCUs, or MC68HC05C4 plus peripheral devices, to be interconnected within a single "black box" or on the same printed circuit board. In a serial peripheral interface (SPI), separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured in one containing one master MCU and several slave MCUs, or in a system in which an MCU is capable of being either a master or a slave.

Figure 6-1 illustrates two different system configurations. Figure 6-1a represents a system of five different MCUs in which there are one master and four slaves (0, 1, 2, 3). In this system four basic lines (signals) are required for the MOSI (master out slave in), MISO (master in slave out), SCK (serial clock), and $\overline{SS}$ (slave select) lines. Figure 6-1b represents a system of five MCUs in which three can be master or slave and two are slave only.

6.1.2 Features

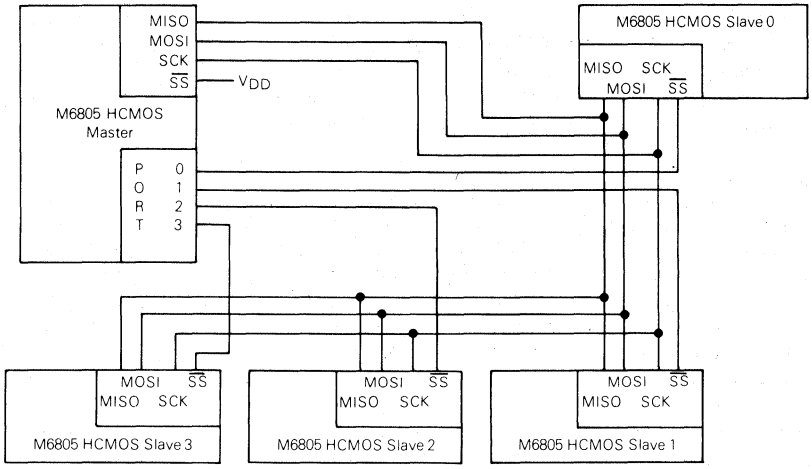
- Full duplex, three-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-Master mode fault protection capability

6.2 SIGNAL DESCRIPTION

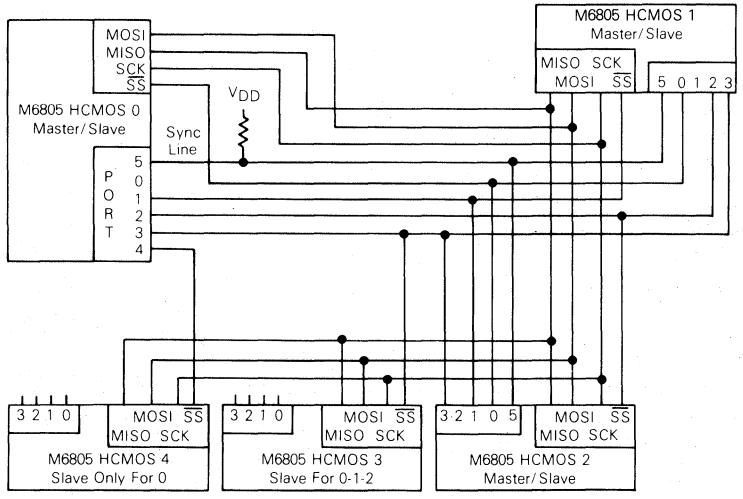
The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) discussed above are described in the following paragraphs. Each signal function is described for both the master and slave mode.

6.2.1 Master Out Slave In (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device. In this manner data is transferred serially from a master to a slave on this line; most



a. Single Master, Four Slaves



b. Three Master/Slave, Two Slaves

Figure 6-1. Master-Slave System Configuration

significant bit first, least significant bit last. The timing diagrams of Figure 6-2 summarize the SPI timing diagram shown in Section 9, and show the relationship between data and clock (SCK). As shown in Figure 6-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

NOTE

Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location $0A). When a device is operating as a master, the MOSI pin is an output because the program in firmware sets the MSTR bit to a logic one.

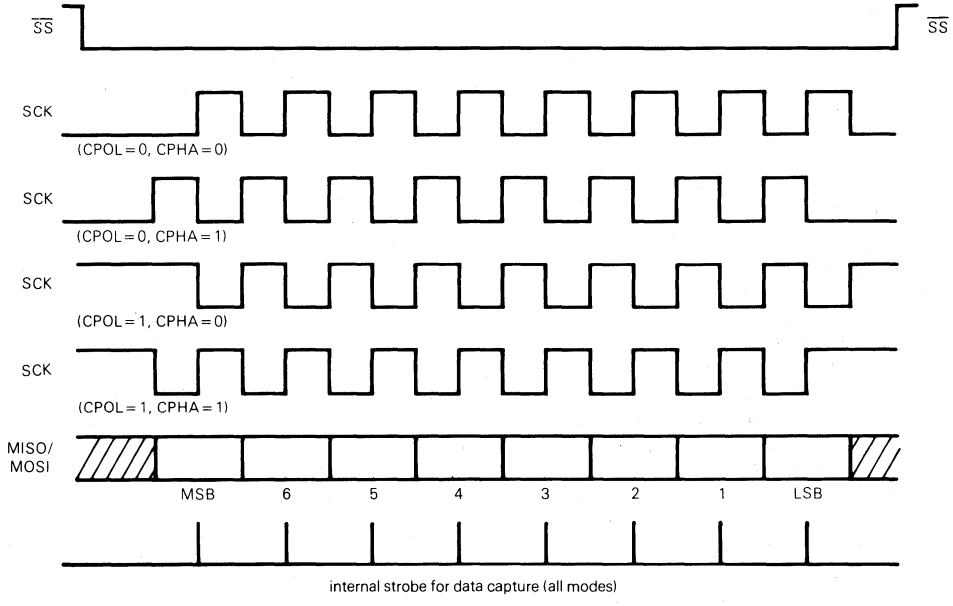


Figure 6-2. Data Clock Timing Diagram

6.2.2 Master In Slave Out (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master; i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of Figure 6-2 shows the relationship between data and clock (SCK). As shown in Figure 6-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

NOTE

The slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial peripheral status register (SPSR, location $0B) is used to signify that the I/O operation is complete.

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $0A) is set to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic level of the $\overline{SS}$ pin; i.e., if $\overline{SS} = 1$ then the MISO pin is placed in the high-impedance state, whereas, if $\overline{SS} = 0$ the MISO pin is an output for the slave device.

6.2.3 Slave Select ($\overline{SS}$)

The slave select ($\overline{SS}$) pin is a fixed input (PD5, pin 34), which receives an active low signal that is generated by the master device to enable slave device(s) to accept data. To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurrence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle. Figure 6-2 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are: 1) with CPHA = 1 or 0, the first bit of data is applied to the MISO line for transfer, and 2) when CPHA = 0 the slave device is prevented from writing to its data register. Refer to the WCOL status flag in the serial peripheral status register (location $0B) description for further information on the effects that the $\overline{SS}$ input and CPHA control bit have on the I/O data register. A high level $\overline{SS}$ signal forces the MISO (master in slave out) line to the high-impedance state. Also, SCK and the MOSI (master out slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. The master device will become a slave device any time its $\overline{SS}$ signal input is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system. When the $\overline{SS}$ line is detected low, it clears the MSTR control bit (serial peripheral control register, location $0A). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port D SPI pins become inputs). The MODF flag bit in the serial peripheral status register (location $0B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a

software error; however, a system could be configured which would contain a default master which would automatically "take-over" and restart the system.

6.2.4 Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (location $0A) discussed below. Refer to Figure 6-2 for timing.

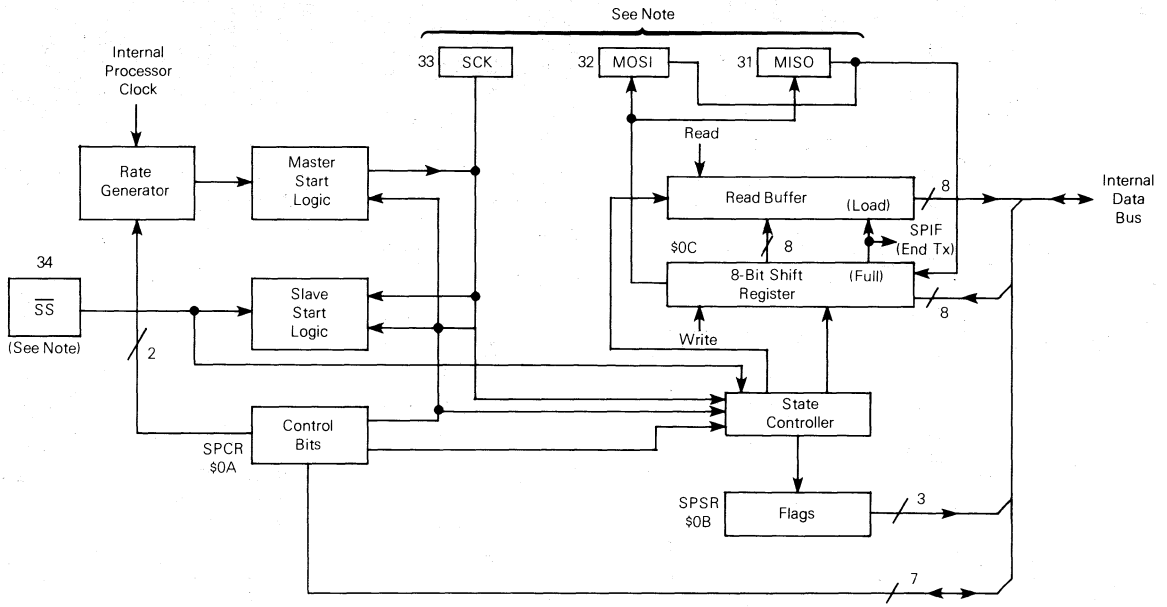
The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the serial peripheral control register (location $0A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface. Timing is shown in Figure 6-2.

6.3 FUNCTIONAL DESCRIPTION

A block diagram of the serial peripheral interface (SPI) is shown in Figure 6-3. In a master configuration, the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the state controller as well as the 8-bit shift register. As a master device, data is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{SS}$ pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Figure 6-4 illustrates the MOSI, MISO, and SCK master-slave interconnections. Note that in Figure 6-4 the master $\overline{SS}$ pin is tied to a logic high and the slave $\overline{SS}$ pin is a logic low. Figure 6-1 provides a larger system connection for these same pins. Note that in Figure 6-1, all $\overline{SS}$ pins are connected to a port pin of a master/slave device. In this case any of the devices can be a slave.



NOTE: The $\overline{SS}$, SCK, MOSI, and MISO are external pins which provide the following functions:

a. MOSI – Provides serial output to slave unit(s) when device is configured as a master. Receives serial input from master unit when device is configured as a slave unit.

b. MISO – Receives serial input from slave unit(s) when device is configured as a master. Provides serial output to master when device is configured as a slave unit.

c. SCK – Provides system clock when device is configured as a master unit. Receives system clock when device is configured as a slave unit.

d. $\overline{SS}$ – Provides a logic low to select a slave device for a transfer with a master device.

Figure 6-3. Serial Peripheral Interface Block Diagram

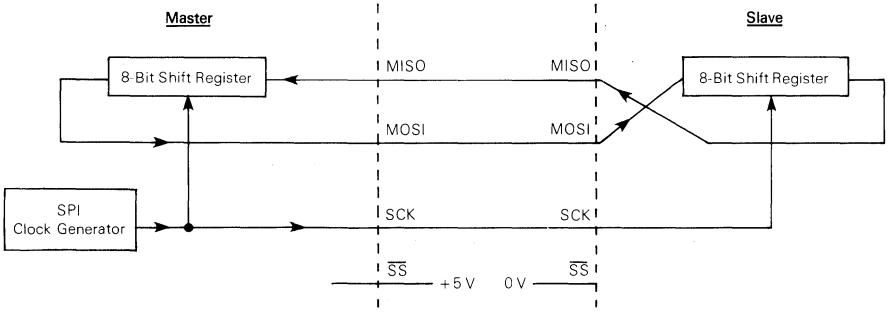


Figure 6-4. Serial Peripheral Interface Master-Slave Interconnection

6.4 REGISTERS

There are three registers in the serial parallel interface which provide control, status, and data storage functions. These registers which include the serial peripheral control register (SPCR, location $0A), serial peripheral status register (SPSR, location $0B), and serial peripheral data I/O register (SPDR, location $0C) are described below.

6.4.1 Serial Peripheral Control Register (SPCR)

7	6	5	4	3	2	1	0	
SPIE	SPE	–	MSTR	CPOL	CPHA	SPR1	SPR0	$0A

The serial peripheral control register bits are defined as follows:

B7, SPIE — When the serial peripheral interrupt enable bit is high, it allows the occurrence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and/or MODF) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

B6, SPE — When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in. Because the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

B4, MSTR — The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident. The MSTR bit is cleared by reset; therefore, the device is always placed in the slave mode during reset.

B3, CPOL — The clock polarity bit controls the normal or steady state value of the clock when data is not being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by reset. Refer to Figure 6-2.

B2, CPHA — The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master and slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relation. The CPHA bit in general selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by reset. Refer to Figure 6-2.

B1, SPR1
B0, SPR0 — These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master; however they have no effect in the slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

6.4.2 Serial Peripheral Status Register (SPSR)

7	6	5	4	3	2	1	0	
SPIF	WCOL	–	MODF	–	–	–	–	$0B

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows:

B7, SPIF — The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle that SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register. While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

B6, WCOL

The function of the write collision status bit is to notify the user that an attempt was made to write the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by 1) a read of the serial peripheral data register prior to the SPIF bit being set, or 2) a read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur in both the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer. The signal on the $\overline{SS}$ pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero. When CPHA is a logic zero, data is latched with the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device.

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will be receiving a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the msb onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

A special case of WCOL occurs in the slave device. This happens when the master device starts a transfer sequence (an edge or SCK for CPHA = 1; or an active $\overline{SS}$ transition for CPHA = 0) at the same time the slave device CPU is writing to its serial peripheral interface data register. In this case it is assumed that the data byte written (in the slave device serial peripheral interface) is lost and the contents of the slave device read buffer becomes the byte that is transferred. Because the master device receives back the last byte transmitted, the master device can detect that a fatal WCOL occurred.

Since the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurrence. This helps alleviate the user from a strict real-time programming effort. The WCOL bit is cleared by reset.

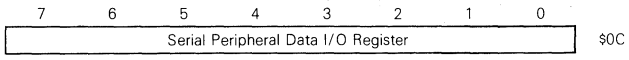
B4, MODF

The function of the mode fault flag is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set. The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

1. MODF is set and SPI interrupt is generated if SPIE = 1.
2. The SPE bit is forced to a logic zero. This blocks all output drive from the device, disables the SPI system.
3. The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

6.4.3 Serial Peripheral Data I/O Register (SPDR)



The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices. A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle that the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the WCOL and SPIF status bits to understand the limits on using the serial peripheral data I/O register.

6.5 SERIAL PERIPHERAL INTERFACE (SPI) SYSTEM CONSIDERATIONS

There are two types of SPI systems; single master system and multi-master systems. Figure 6-1 illustrates both of these systems and a discussion of each is provided below.

Figure 6-1a illustrates how a typical single master system may be configured, using an M6805 HCMOS family device as the master and four M6805 HCMOS family devices as slaves. As shown, the MOSI, MISO, and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock, the slave devices all receive it. Since the M6805 HCMOS master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines. The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO line. For example, three slave devices, enabled for a transfer, are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation. To ensure that proper data transmission is occurring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous

byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. A system of this type is shown in Figure 6-1b. An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

SECTION 7
EFFECTS OF STOP AND WAIT MODES ON THE TIMER AND SERIAL SYSTEMS

7.1 INTRODUCTION

The STOP and WAIT instructions have different effects on the programmable timer, serial communications interface (SCI), and serial peripheral interface (SPI) systems. These different effects are discussed separately below.

7.2 STOP MODE

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing including the operation of the programmable timer, serial communications interface, and serial peripheral interface. The only way for the MCU to "wake up" from the stop mode is by receipt of an external interrupt (logic low on $\overline{IRQ}$ pin) or by the detection of a reset (logic low on $\overline{RESET}$ pin or a power-on reset). The effects of the stop mode on each of the MCU systems (Timer, SCI, and SPI) are described separately.

7.2.1 Timer During Stop Mode

When the MCU enters the stop mode, the timer counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the stop mode is exited by an interrupt (if exited by reset the counter is forced to $FFFC). If the stop mode is exited by an external low on the $\overline{IRQ}$ pin, then the counter resumes from its stopped value as if nothing had happened. Another feature of the programmable timer, in the stop mode, is that if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuitry is armed. This action does not set any timer flags or "wake up" the MCU, but when the MCU does "wake up" there will be an active input capture flag (and data) from that first valid edge which occurred during the stop mode. If the stop mode is exited by an external reset (logic low on $\overline{RESET}$ pin), then no such input capture flag or data action takes place even if there was a valid input capture edge (at the TCAP pin) during the MCU stop mode.

7.2.2 SCI During Stop Mode

When the MCU enters the stop mode, the baud rate generator which drives the receiver and transmitter is shut down. This essentially stops all SCI activity. The receiver is unable to receive and transmitter is unable to transmit. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. When the stop mode is exited, that particular transmission resumes (if the exit is the result of a low input to the $\overline{IRQ}$ pin). Since the previous transmission resumes after an $\overline{IRQ}$ interrupt stop mode exit, the user should ensure that the SCI transmitter is in the idle state when the STOP instruction is executed. If the receiver is receiving data when the STOP instruction is

executed, received data sampling is stopped (baud rate generator stops) and the rest of the data is lost. For the above reasons, all SCI transactions should be in the idle state when the STOP instruction is executed.

7.2.3 SPI During Stop Mode

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, thus the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits the stop mode (provided it is an exit resulting from a logic low on the $\overline{IRQ}$ pin). If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the stop mode, no flags are set until a logic low $\overline{IRQ}$ input results in an MCU "wake up". Caution should be observed when operating the SPI (as a slave) during the stop mode because none of the protection circuitry (write collision, mode fault, etc.) is active.

It should also be noted that when the MCU enters the stop mode all enabled output drivers (TDO, TCMP, MISO, MOSI, and SCK ports) remain active and any sourcing currents from these outputs will be part of the total supply current required by the device.

7.3 WAIT MODE

When the MCU enters the wait mode, the CPU clock is halted. All CPU action is suspended; however, the timer, SCI, and SPI systems remain active. In fact an interrupt from the timer, SCI, or SPI (in addition to a logic low on the $\overline{IRQ}$ or $\overline{RESET}$ pins) causes the processor to exit the wait mode. Since the three systems mentioned above operate as they do in the normal mode, only a general discussion of the wait mode is provided below.

The wait mode power consumption depends on how many systems are active. The power consumption will be highest when all the systems (timer, TCMP, SCI, and SPI) are active. The power consumption will be the least when the SCI and SPI systems are disabled (timer operation cannot be disabled in the wait mode). If a non-reset exit from the wait mode is performed (i.e., timer overflow interrupt exit), the state of the remaining systems will be unchanged. If a reset exit from the wait mode is performed all the systems revert to the disabled reset state.

SECTION 8
INSTRUCTION SET AND ADDRESSING MODES

8.1 INSTRUCTION SET

The MCU has a set of 62 basic instructions. 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.

All of the instructions used in the M146805 CMOS Family are used in the MC68HC05C4 MCU, plus an additional one; the multiply (MUL) instruction. This instruction allows for unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high order product is then stored in the index register and the low order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation: X:A←X*A

Description: Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register.

Condition Codes:

- H: Cleared
- I: Not affected
- N: Not affected
- Z: Not affected
- C: Cleared

Source Form(s): MUL

Addressing Mode	Cycles	Bytes	Opcode
Inherent	11	1	$42

8.1.1 Register/Memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 8-1.

8.1.2 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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 8-2.

Table 8-1. Register/Memory Instructions

		Addressing Modes																		
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnem.	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

Table 8-2. Read-Modify-Write Instructions

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11	—	—	—	—	—	—	—	—	—	—	—	—

8.1.3 Branch Instructions

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between − 127 and + 128 to the current program counter. Refer to Table 8-3.

Table 8-3. Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

8.1.4 Bit Manipulation Instructions

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space except for ROM, port D data location ($03), serial peripheral status register ($0B), serial communications status register ($10), timer status register ($13), and timer input capture register ($14-$15). All port registers, port DDRs, timer, two serial systems, on-chip RAM, and 48 bytes of ROM reside in the first 256 bytes (page zero). An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to Table 8-4.

Table 8-4. Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

8.1.5 Control Instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8-5.

Table 8-5. Control Instructions

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

8.1.6 Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 8-6.

8.1.7 Opcode Map

Table 8-7 is an opcode map for the instructions used on the MCU.

8.2 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 8-7 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the *M6805 HMOS/M146805 CMOS Family Microcomputer/Microprocessor User's Manual.*

Table 8-6. Instruction Set

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			↕	•	↕	↕	↕
ADD		X	X	X		X	X	X			↕	•	↕	↕	↕
AND		X	X	X		X	X	X			•	•	↕	↕	•
ASL	X		X			X	X				•	•	↕	↕	↕
ASR	X		X			X	X				•	•	↕	↕	↕
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	↕	↕	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	↕
BRSET										X	•	•	•	•	↕
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	↕	↕	↕

COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
MUL	X										0	●	●	●	0
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	Λ	Test and Set if True Cleared Otherwise
I	Interrupt Mask	●	Not Affected
N	Negate (Sign Bit)	?	Load CC Register From Stack
Z	Zero	0	Cleared
C	Carry/Borrow	1	Set

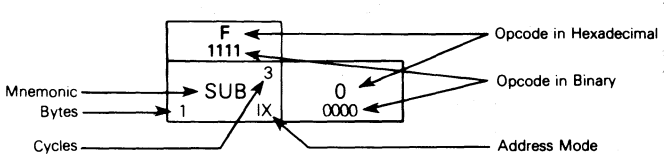
Table 8-7. MC68HC05C4 HCMOS Instruction Set Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DTR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



8.2.1 Inherent

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

8.2.2 Immediate

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

EA = PC + 1; PC ← PC + 2

8.2.3 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 lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers, and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

EA = (PC + 1); PC ← PC + 2

Address Bus High ← 0; Address Bus Low ← (PC + 1)

8.2.4 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

EA = (PC + 1):(PC + 2); PC ← PC + 3

Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

8.2.5 Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

EA = X; PC ← PC + 1

Address Bus High ← 0; Address Bus Low ← X

8.2.6 Indexed, 8-Bit Offset

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in a n element table. All instructions are two bytes. The content of the index register (X) is not changed. The content of

(PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC←PC + 2
Address Bus High←K; Address Bus Low←X + (PC + 1)

where:

K = The carry from the addition of X + (PC + 1)

8.2.7 Indexed, 16-Bit Offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8- or 16-bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC←PC + 3
Address Bus High←(PC + 1) + K;
Address Bus Low←X + (PC + 2)

where:

K = The carry from the addition of X + (PC + 2)

8.2.8 Relative

Relative addressing is only used in branch instructions. In relative addressing, the content of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC←EA if branch taken;
otherwise, EA = PC←PC + 2

8.2.9 Bit Set/Clear

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified in the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

EA = (PC + 1); PC←PC + 2
Address Bus High←0; Address Bus Low←(PC + 1)

8.2.10 Bit Test and Branch

Bit test and branch is a combination of direct addressing, bit set/clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is

added to the PC if the specified bit is set or cleared in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC ← PC + 3

SECTION 9
ELECTRICAL SPECIFICATIONS

9.1 INTRODUCTION

This section contains the electrical specifications and associated timing information for the MC68HC05C4.

9.2 MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS}−0.5 to V_{DD}+0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range	T_A	−55 to +125	°C
Storage Temperature Range	T_{stg}	−65 to +150	°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 $V_{SS} \le (V_{in} \text{ or } V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

9.3 THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Plastic		100	
Chip Carrier		100	

V_{DD} = 4.5 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD6	3.26 kΩ	2.38 kΩ	50 pF
PD1-PD4	1.9 kΩ	2.26 kΩ	200 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA0-PA7, PB0-PB7, PC0-PC7, PD6	10.91 kΩ	6.32 kΩ	50 pF
PD1-PD4	6 kΩ	6 kΩ	200 pF

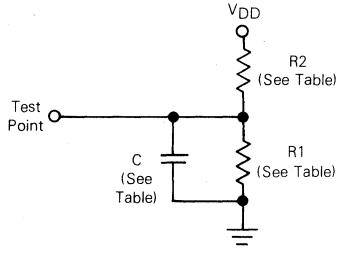


Figure 9-1. Equivalent Test Load

9.4 POWER CONSIDERATIONS

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

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

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins – User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

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

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

Solving equations 1 and 2 for K gives:

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

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

3

9.5 DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55°C to +125°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	– –	0.1 –	V V
Output High Voltage (I_{Load} = 0.8 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP (I_{Load} = 1.6 mA) PD1-PD4	V_{OH} V_{OH}	$V_{DD}-0.8$ $V_{DD}-0.8$	– –	– –	V V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	–	–	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	–	V_{DD}	V
Input Low Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	$0.2 \times V_{DD}$	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 500 ns, (V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V) RUN WAIT (See Note) STOP (See Note)	 I_{DD} I_{DD} I_{DD}	 – – –	 5 1.5 1.0	 TBD TBD TBD	 mA mA μA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	–	–	±10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	–	–	±1	μA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0-PD5, PD7	 C_{out} C_{in}	 – –	 – –	 12 8	 pF pF

NOTE: Measured under the following conditions:

1. All ports are configured as input, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
2. No load on TCMP, C_L = 20 pF on OSC2.
3. OSC1 is a square wave with V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
4. TE = RE = SPE = 0

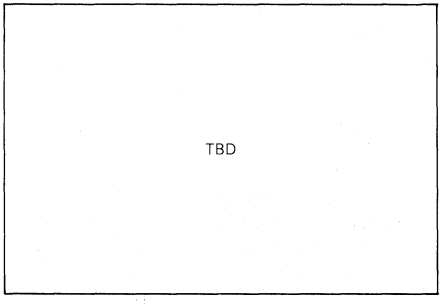


Figure 9-2. Typical Operating Current vs Internal Frequency

9.6 DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55°C to 125°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL}	–	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	–	V
Output High Voltage					
(I_{Load} = 0.2 mA) PA0-PA7, PB0-PB7, PC0-PC7, TCMP	V_{OH}	$V_{DD}-0.3$	–	–	V
(I_{Load} = 0.4 mA) PD1-PD4	V_{OH}	$V_{DD}-0.3$	–	–	V
Output Low Voltage					
(I_{Load} = 0.4 mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4, TCMP	V_{OL}	–	–	0.3	V
Input High Voltage					
PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	–	V_{DD}	V
Input Low Voltage					
PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PD7, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	$0.2 \times V_{DD}$	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1000 ns, (V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V)					
RUN	I_{DD}	–	<4	TBD	mA
WAIT (See Note)	I_{DD}	–	<1.5	TBD	mA
STOP (See Note)	I_{DD}	–	<30	TBD	μA
I/O Ports Hi-Z Leakage Current					
PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD4	I_{IL}	–	<1	±10	μA
Input Current					
$\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0, PD5, PD7	I_{in}	–	<1	±1	μA
Capacitance					
Ports (as input or output)	C_{out}	–	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TCAP, OSC1, PD0-PD5, PD7	C_{in}	–	–	8	pF

NOTE: Measured under the following conditions:

1. All ports are configured as input, V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V.
2. No load on TCMP, C_L = 20 pF on OSC2.
3. OSC1 is a square wave with V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V.
4. TE = RE = SPE = 0.

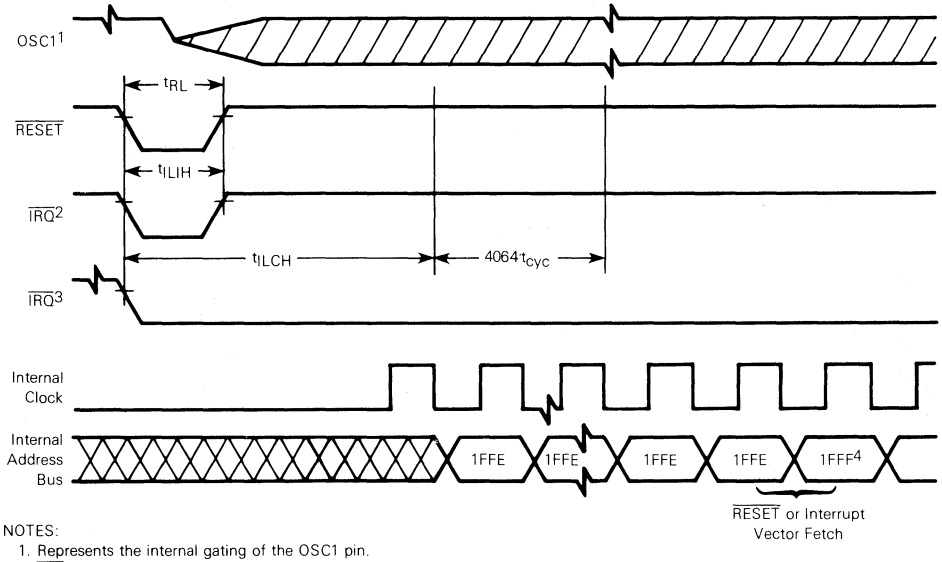
9.7 CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55 to +125°C)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 – dc	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal ($f_{osc} \div 2$) External Clock ($f_{osc} \div 2$)	 f_{op} f_{op}	 – dc	 2.1 2.1	 MHz MHz
Cycle Time (See Figure 3-1)	t_{cyc}	480	–	ns
Crystal Oscillator Startup Time (See Figure 3-1)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 9-3)	t_{ILCH}	–	100	ms
$\overline{RESET}$ Pulse Width (See Figure 3-2)	t_{RL}	1.5	–	t_{cyc}
Timer Resolution** Input Capture Pulse Width (See Figure 9-4) Input Capture Pulse Period (See Figure 9-4)	 t_{TRESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 ***	 – – –	 t_{cyc} ns t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	125	–	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	–	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.



NOTES:
1. Represents the internal gating of the OSC1 pin.
2. $\overline{IRQ}$ pin edge-sensitive mask option.
3. $\overline{IRQ}$ pin level and edge-sensitive mask option.
4. $\overline{RESET}$ vector address shown for timing example.

Figure 9-3. Stop Recovery Timing Diagram

9.8 CONTROL TIMING (V_{DD} = 3.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55 to +125°C)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation				
Crystal Option	f_{osc}	–	2.0	MHz
External Clock Option	f_{osc}	dc	2.0	MHz
Internal Operating Frequency				
Crystal ($f_{osc} \div 2$)	f_{op}	–	1.0	MHz
External Clock ($f_{osc} \div 2$)	f_{op}	dc	1.0	MHz
Cycle Time (See Figure 3-1)	t_{cyc}	1000	–	ns
Crystal Oscillator Startup Time (See Figure 3-1)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 9-3)	t_{ILCH}	–	100	ms
$\overline{RESET}$ Pulse Width – Excluding Power-Up (See Figure 3-1)	t_{RL}	1.5	–	t_{cyc}
Timer				
Resolution**	t_{RESL}	4.0	–	t_{cyc}
Input Capture Pulse Width (See Figure 9-4)	t_{TH}, t_{TL}	250	–	ns
Input Capture Pulse Period (See Figure 9-4)	t_{TLTL}	***	–	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (See Figure 3-4)	t_{ILIH}	250	–	ns
Interrupt Pulse Period (See Figure 3-4)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	–	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.

*** The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.

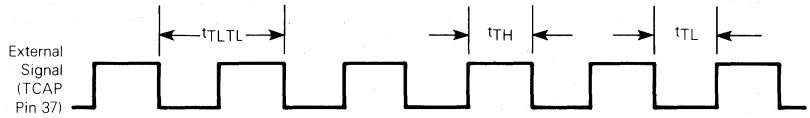


Figure 9-4. Timer Relationships

9.9 SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-5)

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55 to +125°)

Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	1.05	MHz
	Slave	$f_{op(s)}$	dc	2.1	MHz
1	Enable Lead Time				
	Master	$t_{lead(m)}$	*	–	ns
	Slave (CPHA = 0)	$t_{lead(S0)}$	240	–	ns
	Slave (CPHA = 1)	$t_{lead(S1)}$	100	–	ns
2	Enable Lag Time				
	Master	$t_{lag(m)}$	*	–	ns
	Slave (CPHA = 0)	$t_{lag(S0)}$	0.0	–	ns
	Slave (CPHA = 1)	$t_{lag(S1)}$	125	–	ns
3	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	TBD	–	ns
	Slave	$t_{w(SCKH)s}$	TBD	–	ns
4	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	TBD	–	ns
	Slave	$t_{w(SCKL)s}$	TBD	–	ns
5	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	100	–	ns
	Slave	$t_{su(s)}$	100	–	ns
6	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	100	–	ns
	Slave	$t_{h(s)}$	100	–	ns
7	Access Time				
	Slave	t_a	–	TBD	ns
8	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	–	TBD	ns
9	Data Valid				
	Master (Before Capture Edge)	$t_{v(B)m}$	TBD	–	ns
	Slave (After Enable Edge)**	$t_{v(B)s}$	–	200	ns
10	Data Valid				
	Master (After Capture Edge)	$t_{v(A)}$	TBD	–	ns
11	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{rm}	–	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{rs}	–	2.0	µs
12	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{fm}	–	100	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{fs}	–	2.0	µs
13	Output Data Hold (After Enable Edge)				
	Master	$t_{ho(m)}$	0	–	ns
	Slave	$t_{ho(s)}$	0	–	ns

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.

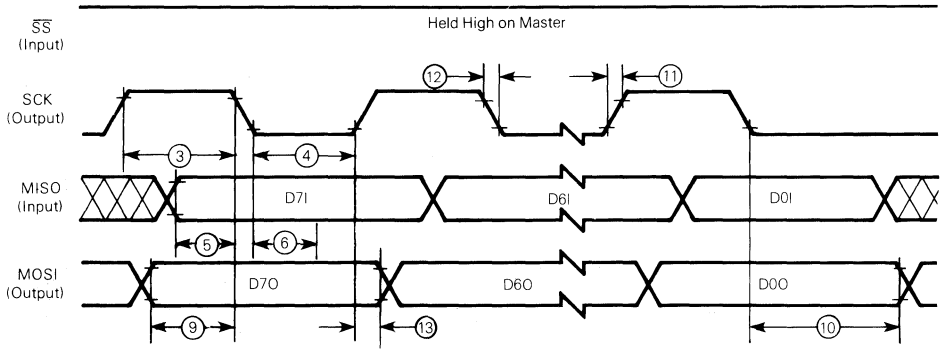
9.10 SERIAL PERIPHERAL INTERFACE (SPI) TIMING (Figure 9-5)

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −55 to +125°C)

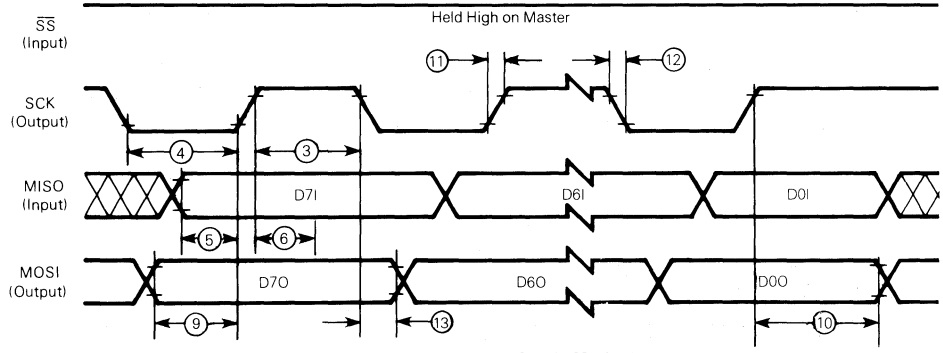
Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency				
	Master	$f_{op(m)}$	dc	0.5	MHz
	Slave	$f_{op(s)}$	dc	1.0	MHz
1	Enable Lead Time				
	Master	$t_{lead(m)}$	*	—	ns
	Slave (CPHA = 0)	$t_{lead(S0)}$	500	—	ns
	Slave (CPHA = 1)	$t_{lead(S1)}$	200	—	ns
2	Enable Lag Time				
	Master	$t_{lag(m)}$	*	—	ns
	Slave (CPHA = 0)	$t_{lag(S0)}$	0.0	—	ns
	Slave (CPHA = 1)	$t_{lag(S1)}$	250	—	ns
3	Clock (SCK) High Time				
	Master	$t_{w(SCKH)m}$	TBD	—	µs
	Slave	$t_{w(SCKH)s}$	TBD	—	ns
4	Clock (SCK) Low Time				
	Master	$t_{w(SCKL)m}$	TBD	—	µs
	Slave	$t_{w(SCKL)s}$	TBD	—	ns
5	Data Setup Time (Inputs)				
	Master	$t_{su(m)}$	200	—	ns
	Slave	$t_{su(s)}$	200	—	ns
6	Data Hold Time (Inputs)				
	Master	$t_{h(m)}$	200	—	ns
	Slave	$t_{h(s)}$	200	—	ns
7	Access Time				
	Slave	t_a	—	TBD	ns
8	Disable Time (Hold Time to High-Impedance State)				
	Slave	t_{dis}	—	TBD	ns
9	Data Valid				
	Master (Before Capture Edge)	$t_{v(B)m}$	TBD	—	ns
	Slave (After Enable Edge)**	$t_{v(B)s}$	—	400	ns
10	Data Valid				
	Master (After Capture Edge)	$t_{v(A)}$	TBD	—	ns
11	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{rm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{rs}	—	2.0	µs
12	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)				
	SPI Outputs (SCK, MOSI, MISO)	t_{fm}	—	200	ns
	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)	t_{fs}	—	2.0	µs
13	Output Data Hold (After Enable Edge)				
	Master	$t_{ho(m)}$	0	—	ns
	Slave	$t_{ho(s)}$	0	—	ns

* Signal production depends on software.

** Assumes 200 pF load on all SPI pins.



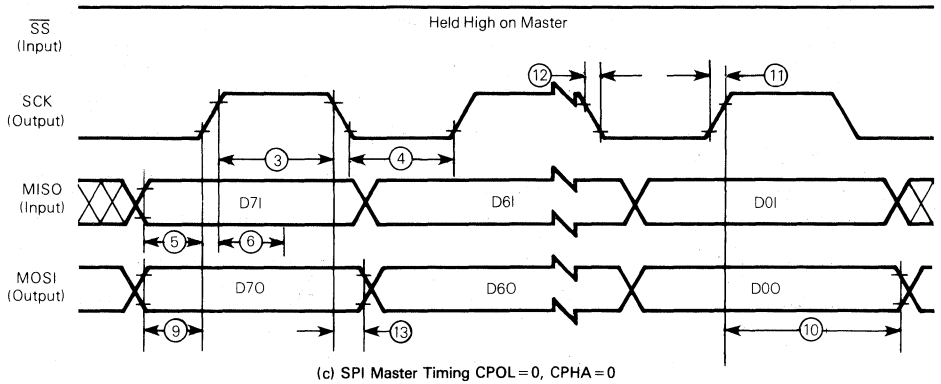
(a) SPI Master Timing CPOL=0, CPHA=1



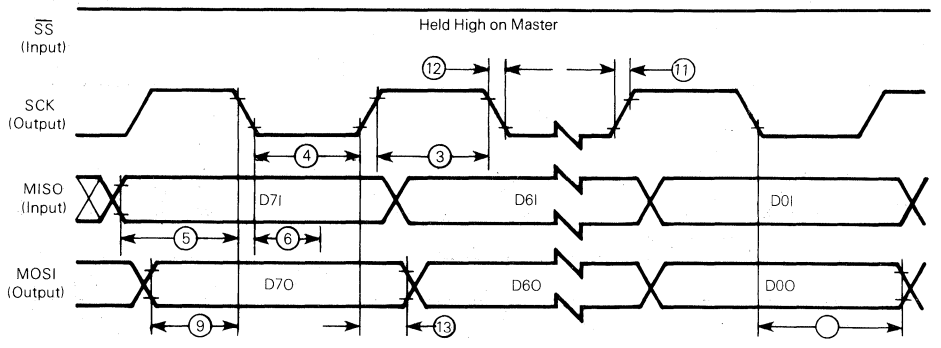
(b) SPI Master Timing CPOL=1, CPHA=1

NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.

Figure 9-5. Timing Diagrams



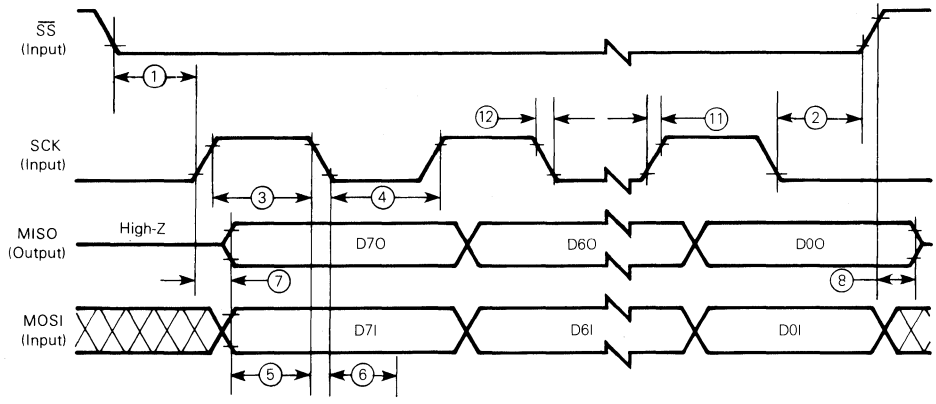
(c) SPI Master Timing CPOL = 0, CPHA = 0



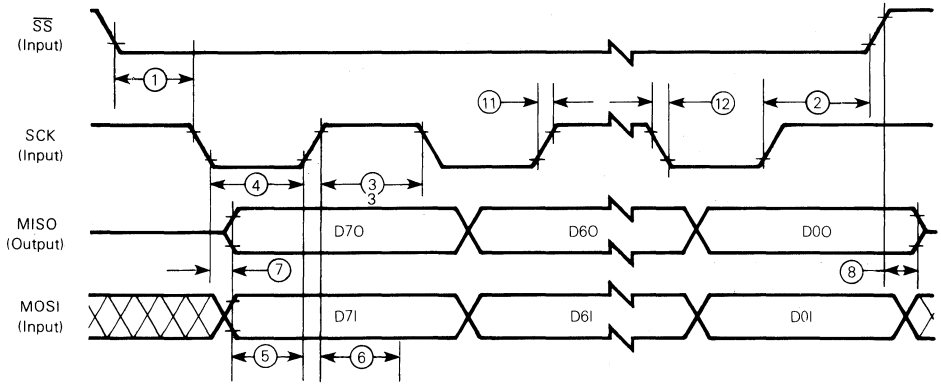
(d) SPI Master Timing CPOL = 1, CPHA =

NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.

Figure 9-5. Timing Diagrams (Continued)



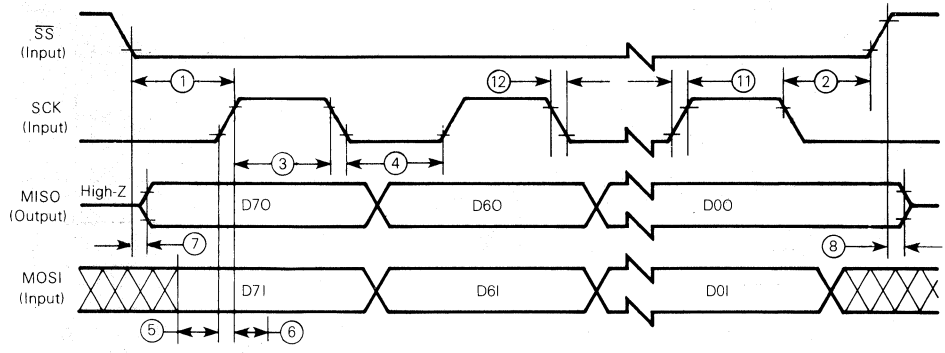
(e) SPI Slave Timing CPOL = 0, CPHA = 1



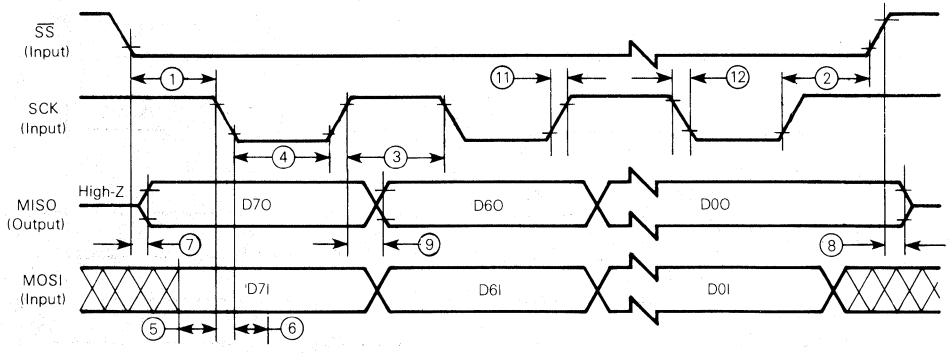
(f) SPI Slave Timing CPOL = 1, CPHA = 1

NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.

Figure 9-5. Timing Diagrams (Continued)



(g) SPI Slave Timing CPOL = 0, CPHA = 0



(h) SPI Slave Timing CPOL = 1, CPHA = 0

NOTE: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.

Figure 9-5. Timing Diagrams (Continued)

SECTION 10
ORDERING INFORMATION

10.1 INTRODUCTION

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

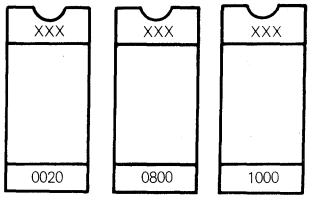
EPROM(s), MCM2716 or MCM2532
MDOS, disk file

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

10.1.1 EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure 10-1 illustrates the markings for the three MCM2716 EPROMs required to emulate the MC68HC05C4 MCU.

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.



XXX = Customer ID

Figure 10-1. EPROM Marking Example

10.1.2 MDOS Disk File

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.**

10.2 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 contractural agreement for creation of the customer mask. If desired, Motorola will program a blank MCM2716 or MCM2532 EPROM (**supplied by the customer**) from the data file used to create the custom mask to aid in the verification process.

10.3 ROM VERIFICATION UNITS

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 (25°) and five volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

10.4 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 and company name. The floppies are not returned by Motorola as they are used for archival storage. The minimum MDOS system files as well as the absolute binary object file (filename.LO type of file) from the M6805 cross assembler 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.

OPTION LIST

Select the options for the MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Customer Name_____

Address_____

City_____ State_____ Zip_____

Phone (_____) _____ Extension_____

Contact Ms/Mr_____

Customer Part Number_____

Pattern Media
- ☐ 2532 EPROM
- ☐ 2716 EPROM
- ☐ MDOS Disk File
- ☐ (Note) _____

Note: Other Media require prior factory approval.

Signature_____

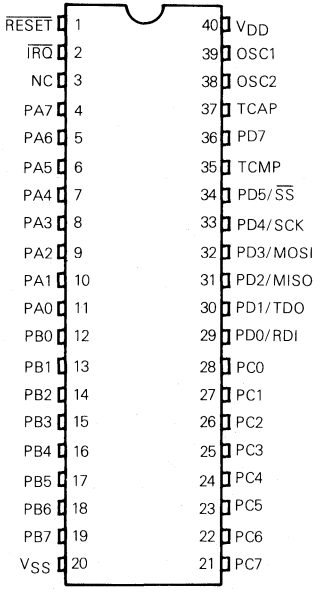
Title_____

Figure 10-2. Ordering Form

SECTION 11
MECHANICAL DATA

This section contains the pin assignment and package dimension diagrams for the MC68HC05C4 microcomputer.

11.1 PIN ASSIGNMENT



NOTE: Pin assignments are the same for both the dual in-line and chip carrier package.

MOTOROLA

MC68HC11A4

Product Preview

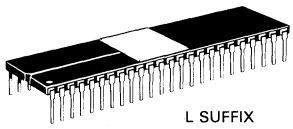
MC68HC11A4 8-BIT MICROCOMPUTER

The HCMOS MC68HC11A4 is an advanced microcomputer (MCU) containing highly sophisticated on-chip peripheral functions. An improved instruction set provides additional capability while maintaining compatibility with the other members of the M6801 Family. The fully static design allows operation at frequencies down to dc, further reducing its already low power consumption. Features available in addition to the normal M6801 features include:

- 4K Bytes of ROM
- 512 Bytes of EEPROM
- 256 Bytes of RAM (All Saved During Standby)
- Enhanced 16-Bit Timer System
  - Four Stage Programmable Prescaler
  - Three Input Capture Functions
  - Five Output Compare Functions
- An 8-Bit Pulse Accumulator Circuit
- An Enhanced Non-Return-To-Zero Serial Communications Interface (SCI)
- A New Serial Peripheral Interface
- Eight Channel 8-Bit A/D Converter
- A Real Time Interrupt Circuit
- A Computer Operating Properly (COP) Watchdog System

HCMOS
(HIGH-DENSITY HIGH-PERFORMANCE SILICON GATE)

MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE

Also Available In
52-Pin Quad Pack

3

PIN ASSIGNMENT

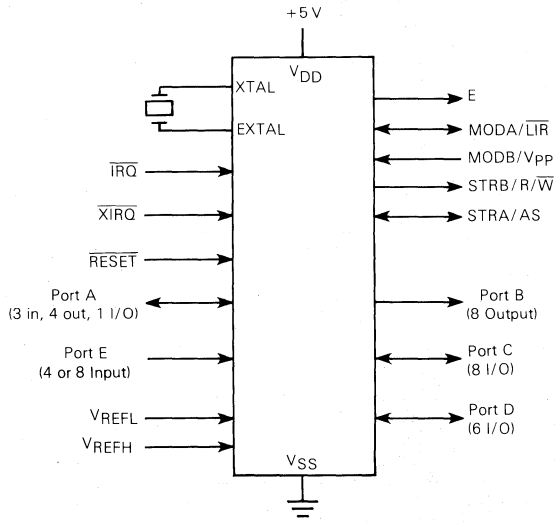
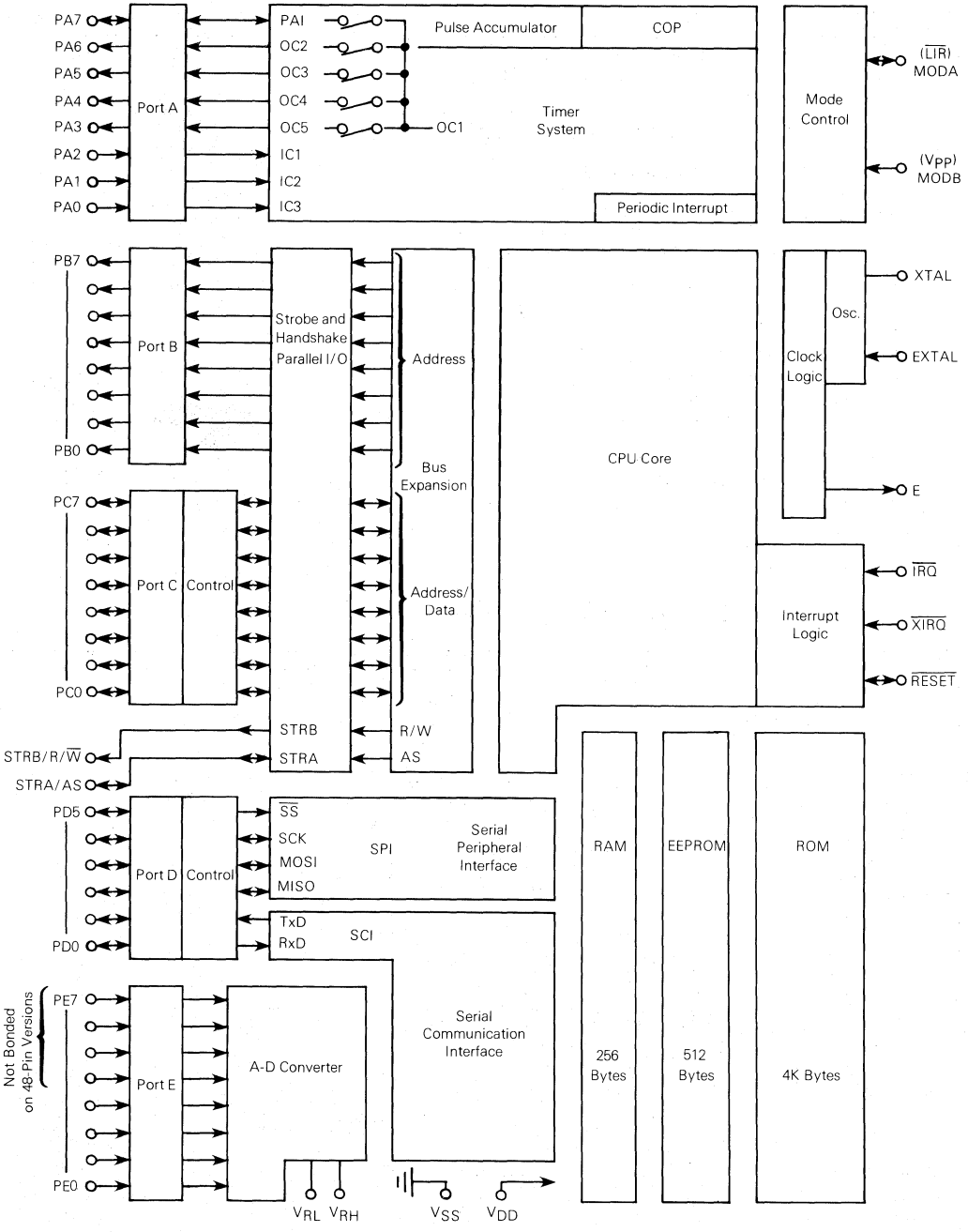


FIGURE 1 — MC68HC11A4 BLOCK DIAGRAM



GENERAL DESCRIPTION

The MC68HC11A4 is a single-chip microcomputer that utilizes HCMOS techniques to provide the low-power characteristics and high noise immunity of CMOS plus the high-speed operation of HMOS. On chip memory systems include a 4K byte ROM, 512 bytes of electrically erasable programmable ROM (EEPROM), and 256 bytes of static RAM. The MC68HC11A4 microcomputer also provides highly sophisticated, on-chip peripheral functions including: an 8-channel analog-to-digital converter, a serial communications interface (SCI) subsystem, and a serial peripheral interface (SPI) subsystem.

New design techniques are used to provide a 2 MHz nominal bus rate. The timer system is expanded to provide three input capture lines, five output compare lines, and a real time interrupt circuit. This gives the MC68HC11A4 one of the most comprehensive timer systems found on a single-chip microcomputer. Other features of the MC68HC11A4 include: a pulse accumulator which can be used to count external events (event counting mode) or measure an external period (input gates accumulation of internal clock — E/64); a computer operating properly (COP) watchdog system which helps protect against software failures; a programmable clock monitor system which causes generation of a system reset in case the clock is lost or running too slow; and an illegal opcode detection circuit which provides an unmaskable interrupt if an illegal opcode fetch is detected.

OPERATING MODES

The MC68HC11A4 MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip (mode 0) and expanded multiplexed (mode 1), and the special operating modes are bootstrap and special test. The levels required on the MODA and MODB pins for mode selection are discussed in **FUNCTIONAL PIN DESCRIPTION.** The characteristics of the operating modes are discussed below.

SINGLE-CHIP MODE (MODE 0)

In the single-chip mode the MCU functions as a self-contained microcomputer and has no external address or data bus. In this mode the MCU provides maximum use of its pins for on-chip peripheral functions, and all address and data activity occurs within the MCU. As discussed in **FUNCTIONAL PIN DESCRIPTION,** when MODA = 0 and MODB = V_{DD} the single chip mode is selected during reset.

EXPANDED MULTIPLEXED MODE (MODE 1)

In this mode, two I/O ports plus two additional I/O lines become address, data, and control (AS and R/$\overline{W}$) to allow the MCU to address up to 64K bytes of address space. High order address bits are output on the port B pins. Low order address bits and the data bus are time multiplexed on the eight port C pins. Port D bit 6 becomes the address strobe (AS) control output which is used in demultiplexing the low order address from the data at port C. The R/$\overline{W}$ control pin (port D, pin 7) is used to control the direction of data transfers on the port C bus. Refer to **FUNCTIONAL PIN DESCRIPTION** and **INPUT/OUTPUT PORTS** for additional information regarding address strobe, read/write, port B, and port C.

BOOTSTRAP MODE

The bootstrap mode is considered a special mode as distinguished from the normal operating single-chip mode. In the bootstrap mode, all vectors are fetched from the 128 byte on-chip boot loader ROM. This is a very versatile mode since there are essentially no limitations on the special purpose program that is boot loaded into the internal RAM. The boot loader is contained in 128 bytes of ROM which is enabled as internal memory space at $BF80-$BFFF. The boot loader contains a small program which reads a 256 byte program into on-chip RAM ($0000-$00FF). After the character for address $00FF is received, control is automatically passed to that program at memory address $0000 and the MCU operates in the single-chip mode.

In the bootstrap mode, the serial receive logic is initialized by software in the boot loader ROM. This allows the program control of the serial communications interface (SCI) baud and word format.

During initialization of the special bootstrap mode, a special control bit is configured to permit access to a number of special test control bits which allows for self testing of the MCU in the bootstrap mode. Also, since the mode control bits can be written to, the operating mode of the MCU may be changed from the special bootstrap mode (which is a single-chip mode by default) to expanded multiplexed mode under program control.

TEST MODE

The test mode is considered a special mode as distinguished from the normal operating expanded multiplexed mode; however, it is considered as operating in the expanded multiplexed mode since external memory may be addressed. The reset vector is fetched from external memory space $BFFE-$BFFF; therefore, program control may be vectored to an external test program.

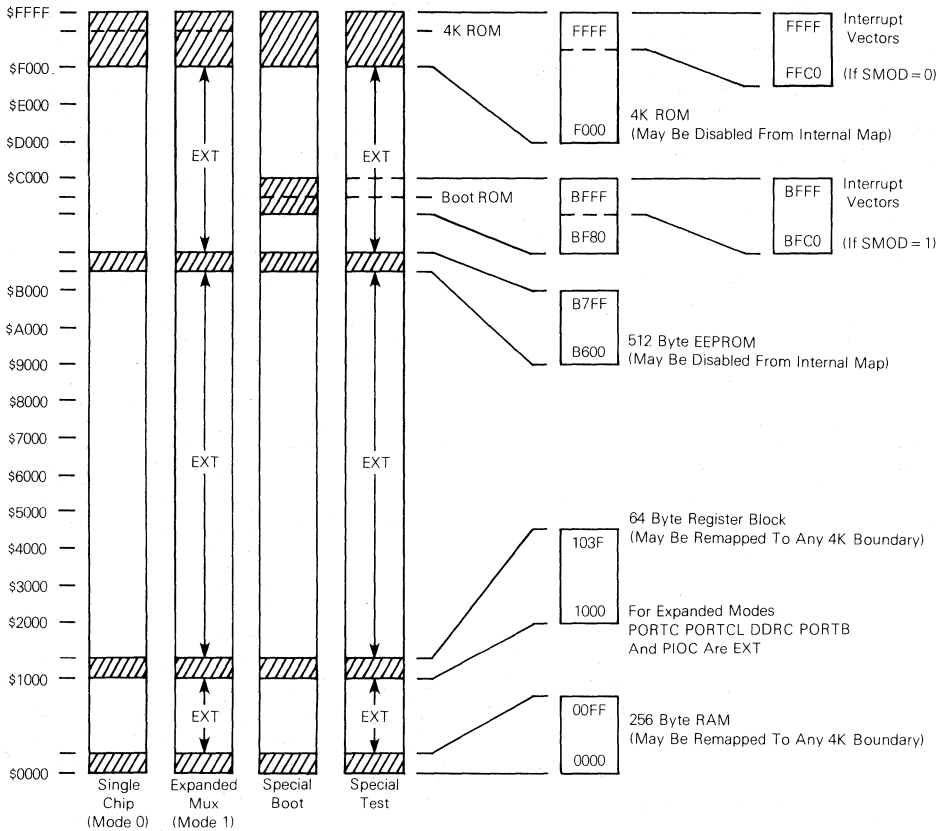
The test mode is primarily intended as the main production test mode at the time of manufacture; however, it may also be used to program calibration or personality data into the internal EEPROM (electrically erasable programmable read only memory) of the MC68HC11A4. During initialization of the test mode, a special control bit is configured to permit access to a number of special test control bits. Also, since the mode control bits can be written to in the test mode, the operating mode of the MCU may be changed from the special test mode (which is an expanded multiplexed mode by default) to the single-chip mode under program control.

MEMORY

Composite memory maps for each MC68HC11A4 mode of operation are shown in Figure 2. These modes include single-chip, expanded multiplexed, special boot, and special test.

In the single-chip mode (mode 0) of Figure 2, the MC68HC11A4 does not generate external addresses. The actual internal memory locations are shown in the shaded areas of Figure 2 and the contents of these shaded areas are

FIGURE 2 — MC68HC11A4 MEMORY MAPS



shown on the right side of the diagram. For example: memory locations $0000 through $00FF contain the 256 bytes allocated to RAM; memory locations $1000 through $103F are allocated for a 64-byte register block; memory locations $B600 through $B7FF are allocated for a 512-byte EEPROM (electrically eraseable programmable read only memory); and memory locations $F000 through $FFFF are allocated for 4K bytes of ROM (memory locations $FFC0 through $FFFF are reserved for the interrupt and reset vectors).

The expanded multiplexed mode (mode 1) memory locations shown in Figure 2 are basically the same as for the single-chip mode; however, the memory locations between the shaded areas (designated EXT) are for externally addressed memory and I/O.

The special bootstrap mode memory locations are similar to the single-chip memory locations except that a special bootstrap program is addressed at memory locations $BF80 through $BFFF. The special bootstrap program controls the process of boot loading a 256 byte program through a serial port into internal RAM.

The special test mode memory locations are similar to the expanded multiplex mode except the interrupt vectors are at external memory locations.

CENTRAL PROCESSING UNIT

The central processing unit (CPU) of the MC68HC11A4 is basically an extension of the MC6801 CPU. In addition to being able to execute all MC6800 and MC6801 instructions, the MC68HC11A4 uses a 4-page opcode map to allow execution of 91 new opcodes. As in the MC6801, the CPU of the MC68HC11A4 is implemented independently from the I/O,

memory, or on-chip peripheral configurations. Consequently, this CPU can be treated as an independent processor communicating with these internal subsystems when operating in the single-chip mode. However, when the MC68HC11A4 is operating in the extended multiplexed mode, it is capable of addressing external memory and peripherals in addition to communicating with the on-chip subsystems.

The MC68HC11A4 CPU has seven registers available to the programmers as shown in Figure 3. The interrupt stacking order is shown in Figure 4. The seven registers are discussed below.

FIGURE 3 – MC68HC11A4 PROGRAMMING MODEL

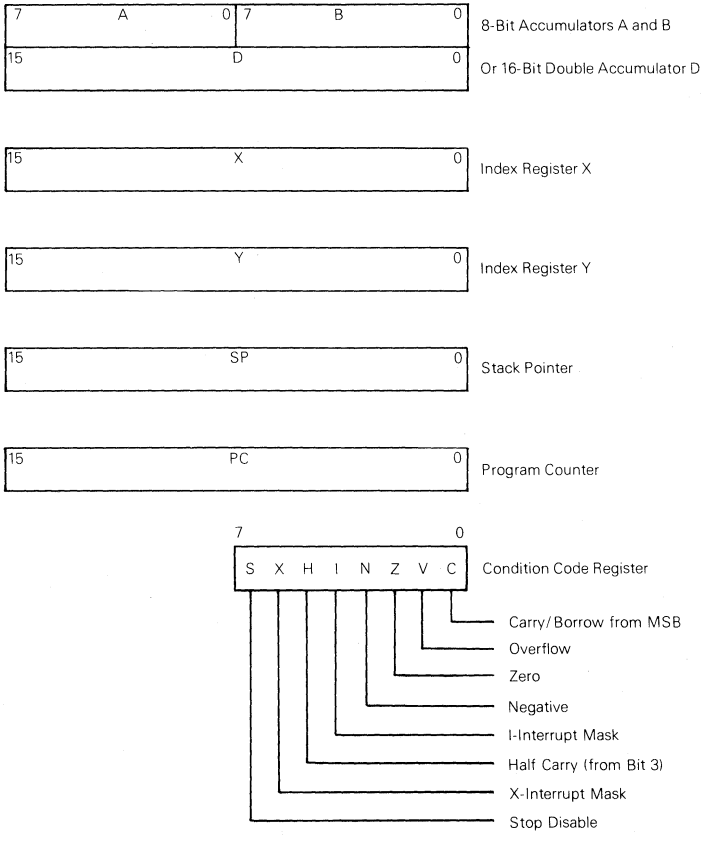
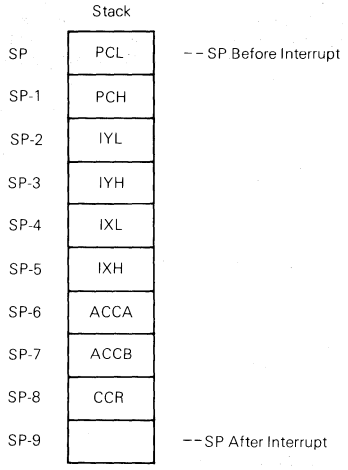


FIGURE 4 — INTERRUPT STACKING ORDER



ACCUMULATOR A AND B

Accumulator A and accumulator B are general purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. As in the MC6801, these two accumulators can be concatenated into a single double-byte accumulator called the D accumulator.

INDEX REGISTER X (IX)

The 16-bit IX register is used during indexed modes of addressing. It provides a 16-bit indexing value which may be added to an 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage area.

INDEX REGISTER Y (IY)

The 16-bit IY register is also used during indexed modes of addressing similar to the IX register; however, most instructions using the IY register require an extra byte of machine code and a cycle of execution time since they are two byte opcodes. The IY register can also be used as a counter or as a temporary storage in the same manner as the IX register.

STACK POINTER (SP)

The stack pointer (SP) is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack (a push), the SP is decremented; whereas, each time a byte is removed from the stack (a pull) the SP is incremented. The address contained in the SP also indicates the location at which the accumulators (A and B), IX, and IY can be stored during certain instructions.

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next instruction to be executed.

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit signifies the results of the instruction just executed. These bits can be individually tested by a program and a specific action can be taken as a result of the test. Each individual condition code register bit is explained below.

Carry/Borrow (C)

The C bit is set if there was a carry or borrow out of the arithmetic logic unit (ALU) during the last arithmetic operation. The C bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if there was an arithmetic overflow as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

The zero bit is set if the result of the last arithmetic, logic, or data manipulation was zero; otherwise, the Z bit is cleared.

Negative (N)

The negative bit is set if the result of the last arithmetic, logic, or data manipulation was negative (b7 of result equal to a logic one); otherwise, the N bit is cleared.

I Interrupt Mask (I)

The I interrupt mask bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

The half carry bit is set to a logic one when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction; otherwise, the H bit is cleared.

X Interrupt Mask (X)

The X interrupt mask bit is set only by hardware (Reset or $\overline{\text{XIRQ}}$); and it is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

The stop disable bit is set to disable the STOP instruction, and cleared to enable the STOP instruction. The S bit is program controlled. The STOP instruction is treated as no operation (NOP) if the S bit is set.

FUNCTIONAL PIN DESCRIPTION

The below pin descriptions do not include the I/O ports. They are discussed separately under **INPUT/OUTPUT PORTS.**

V_{DD} AND V_{SS}

Power is supplied to the MC68HC11A4 using these two pins. V_{DD} is power input (+5 V) and V_{SS} is the power return path.

$\overline{RESET}$

This active low bi-directional control pin is used as an input to initialize the MC68HC11A4 to a known start-up state, and as an open-drain output to indicate an internal failure has been detected in either the clock monitor or computer operating properly (COP) circuit.

XTAL, EXTAL

These two inputs provide for an interface with either a crystal input or a CMOS compatible clock to control the MC68HC11A4 internal clock generator circuitry. The frequency applied to these pins should be four times the desired internal clock rate since an internal divide-by-four circuit determines the actual E-clock rate. When a crystal is used, a 25 picofarad capacitor should be connected between V_{SS} and each of these two pins (XTAL and EXTAL); however, if a CMOS compatible external clock is used, the signal should be connected to the EXTAL pin and the XTAL pin should be left disconnected.

E

The E pin provides an output for the internally generated E-clock which can be used as a timing reference. The frequency of the E output is actually one fourth that of the input frequency at the XTAL and EXTAL pins. In general when the E pin is low, an internal process is taking place and, when high, data is being accessed. This output becomes inactive during the power-saving wait mode if the MC68HC11A4 is operating in the single-chip or bootstrap modes (see **MODA/$\overline{LIR}$, MODB/V_{PP}** description below).

$\overline{IRQ}$

The $\overline{IRQ}$ pin provides a means for requesting asynchronous interrupts to the MC68HC11A4. The $\overline{IRQ}$ interrupt input is program selectable with a choice of either negative edge-sensitive or level-sensitive triggering. The $\overline{IRQ}$ interrupt input is always configured to level-sensitive triggering during reset. The $\overline{IRQ}$ pin requires an external resistor to V_{DD}. The MCU completes the current instruction before responding to an interrupt request on the $\overline{IRQ}$ pin.

If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

$\overline{XIRQ}$

The $\overline{XIRQ}$ pin provides a means for requesting asynchronous non-maskable interrupts to the MC68HC11A4, after a power-on reset. During reset (including power-on reset), the X bit in the condition code register is set and the $\overline{XIRQ}$ interrupt is masked to preclude interrupts on this line until MCU operation is stabilized. The X bit may then be cleared by program control (using the transfer accumulator A instruction, TAP); however, the X bit can only be set again by reset or by recognition of a hardware $\overline{XIRQ}$ interrupt. Once the X bit in the condition code register is cleared, an interrupt on the $\overline{XIRQ}$ pin will be serviced as soon as the MCU completes the current instruction. When an $\overline{XIRQ}$ interrupt is recognized, on-chip hardware automatically sets the X bit. The X bit can be cleared either as part of interrupt routine by the TAP instruction (nested interrupt) or by the return from interrupt instruction. The $\overline{XIRQ}$ pin requires an external resistor to V_{DD}.

The $\overline{XIRQ}$ input may also be used to return the MCU to normal operation from the low-power stop mode by applying a low level to the $\overline{XIRQ}$ pin. If the X bit in the condition code register is cleared and the MCU is in the stop mode, a low input on the $\overline{XIRQ}$ brings the MCU out of the stop mode and operation resumes with the stacking operation leading to service of the $\overline{XIRQ}$ request. If the X bit is set and the MCU is in the stop mode, a low input on the $\overline{XIRQ}$ brings the MCU out of the stop mode and operation resumes with the program instruction following the STOP instruction.

MODA/$\overline{LIR}$, MODB/V_{PP}

These pins have alternate functions, MODA and MODB controlling one function, V_{PP} controlling an alternate function, and $\overline{LIR}$ used for an alternate function.

MODA, MODB

During reset these two pins are used to control the two basic operating modes of the MC68HC11A4 plus two special operating modes. The modes versus MODA and MODB inputs are shown in the table below.

MODB	MODA	Mode Selected
V_{DD}	0	Single-Chip (Mode 0)
V_{DD}	1	Expanded Multiplexed (Mode 1)
#	0	Special Bootstrap
#	1	Special Test

1 = Logic High
0 = Logic Low
# = 1.4 Times V_{DD} (or Higher)

V_{PP}

In addition to the MODA function, the MODA/V_{PP} pin is also used to supply the programming voltage for programming the internal EEPROM. Changing the voltage applied to this pin after reset has no affect on mode selection.

$\overline{LIR}$

In addition to the MODA function, the MODA/$\overline{LIR}$ pin provides an output as an aid in debugging once reset is completed. The LIR pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle (opcode fetch). Some MC68HC11A4 opcodes are two consecutive bytes long including a page 2 (PG2), page 3 (PG3), or page 4 (PG4) prebyte. For these instructions $\overline{LIR}$ goes low for only the first (prebyte) opcode byte fetch.

NOTE

The $\overline{LIR}$ output will not go low for at least two E-clock cycles after reset because of the reset vector fetch.

V_{REFL}, V_{REFH}

These two pins provide the reference voltage for the analog-to-digital converter.

R/$\overline{W}$/STRB

This pin provides two different functions depending on the operating mode. In single-chip mode the pin provides the STRB (output strobe) function and in the expanded multiplexed mode it provides the R/$\overline{W}$ (read-write) function.

In the single-chip mode the STRB pin acts as a programmable strobe. This strobe can also be used to provide a data acknowledge (handshake) to a parallel I/O device.

In the expanded multiplexed mode the R/$\overline{W}$ (read/write) is used to control the direction of transfers on the external data bus. A low level (write) on the R/$\overline{W}$ pin enables the data bus output drivers to the external data bus. A high level (read) on this pin forces the output drivers to a high-impedance state and data is read from the external bus.

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides the STRA (input strobe) function and in the expanded multiplexed mode it provides the AS (address strobe) function.

In the single-chip mode, the STRA pin acts as a programmable input strobe. This input is also used with STRB and port C for full handshake modes of parallel I/O.

In the expanded multiplexed mode the AS (address strobe) output may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT PORTS

There are five 8-bit ports on the MC68HC11A4 MCU. All of these ports serve more than one purpose depending on the mode configuration of the MCU. A summary of the pins versus function and mode is provided in Table 1 and discussed in the following paragraphs. Because the port functions are controlled by the particular mode selected, each port is discussed for its function(s) during the mode of operation.

SINGLE-CHIP MODE

In the single-chip mode the MC68HC11A4 functions as a monolithic microcomputer without external address or data buses. In this mode, four of these ports (A, B, C, D) are configured as parallel I/O data ports. Port E can be used for general purpose static inputs and/or analog-to-digital converter channel inputs.

Port A

In all operating modes (including the single-chip mode) port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PAI) or a fifth output compare function (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of the free running counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PAI), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function, as described above, may be used as a general purpose input or output line.

Port B

In the single-chip mode, all of the port B pins are general purpose output pins. During MCU read cycles the levels sensed at the input side of the port B output drivers is read. Port B may also be used in a simple strobed output mode where the STRB (port D bit 7) pulses each time port B is written.

Port C

In the single-chip mode, all port C pins are general purpose input/output pins. Port C inputs can be latched by the STRA input (at port D bit 6). Port C may also be used in full handshake modes of parallel I/O where the STRA input and STRB output act as handshake control lines.

Port D

In the single-chip mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 are used as handshake control signals for ports B and C.

Bit 0 is the receive data input (RxD) for the serial communication interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI.

Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select ($\overline{SS}$) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (STRA) and 7 (STRB) are discussed in **FUNCTIONAL PIN DESCRIPTION.**

TABLE 1 — PORT SIGNAL SUMMARY

Port-Bit	Single-Chip Modes 0 and Bootstrap Mode	Expanded Multiplexed Mode 1 and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/SS	PD5/SS
D-6	STRA	AS
D-7	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4 ##	PE4/AN4 ##
E-5	PE5/AN5 ##	PE5/AN5 ##
E-6	PE6/AN6 ##	PE6/AN6 ##
E-7	PE7/AN7 ##	PE7/AN7 ##

- not bonded in 48-pin variations

Port E

In all operating modes (including the single-chip mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to pins.

EXPANDED MULTIPLEXED MODE

In the expanded multiplexed mode, the MC68HC11A4 has the capability of accessing a 64K byte address space. The total address space includes the same on-chip memory address as for single-chip mode plus external peripheral devices. In this mode ports B, C, and bits 6 and 7 of port D are configured as a memory expansion bus.

Port A

In all operating modes (including the expanded multiplexed mode), port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PAI) or a fifth output compare functon (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of a free running

counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PAI), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function as described above, may be used as a general purpose input or output line.

Port B

In the expanded multiplexed mode, all of the port B pins act as high order address output pins. During each MCU cycle, bits 8 through 15 of the address are output on the PB0-PB7 lines respectively.

Port C

In the expanded multiplexed mode, all port C pins are configured as multiplexed address/data pins. During the address portion of each MCU cycle, bits 0 through 7 of the address are output on the PC0-PC7 lines. During the data portion of each MCU cycle (E high), bits 0 through 7 (D0-D7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

Port D

In the expanded multiplexed mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 act as expansion bus control lines AS and R/W respectively.

Bit 0 is the receive data input (RxD) for the serial communications interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI.

Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select ($\overline{SS}$) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (AS) and 7 (R/$\overline{W}$) are discussed in **FUNCTIONAL PIN DESCRIPTION.**

Port E

In all operating modes (including the expanded multiplexed mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to external pins.

BOOTSTRAP MODE

In the bootstrap mode all I/O port pins function the same as in the single-chip mode. Operational differences are discussed in **OPERATING MODES.**

TEST MODE

In the test mode all I/O port pins function the same as in the expanded multiplexed mode. Operational differences are discussed in **OPERATING MODES.**

INTERRUPTS

The MC68HC11A4 MCU interrupts can be generated by any of four different basic methods: (1) by presenting the appropriate external signal; (2) by enabling interrupts from the programmable timer output compare or input capture, serial communication interface, serial peripheral interface timer overflow, pulse accumulator, or parallel I/O; (3) by executing a software interrupt (SWI) instruction; or (4) by detection of an illegal opcode.

The program may also be interrupted by: (1) detection of a timeout in the computer operating properly (COP) circuit, (2) clock monitor detects loss of the E-clock or a low frequency E-clock, or (3) by a reset. The above three methods of interrupting the program result in fetching a reset vector rather than an interrupt vector; however, they do interrupt the program.

When an external or internal (hardware) interrupt occurs, the interrupt is not serviced until the current instruction being executed is completed. Until the current instruction is complete, the interrupt is considered pending. After completion of current instruction execution, unmasked interrupts may be serviced in accordance with an established fixed hardware priority circuit; however, one I bit related interrupt source may be elevated to the highest I bit priority position in the circuit.

Seventeen hardware interrupts and one software interrupt (excluding reset type interrupts) can be generated from all of the possible sources. The interrupts can be divided into two basic categories, maskable and non-maskable. In the MC68HC11A4 fifteen of the interrupts can be masked using the condition code register I bit. In addition to being maskable by the I bit in the condition code register, all of the on-chip interrupt sources are individually maskable by control bits. The software interrupt (SWI instruction) is a non-maskable instruction rather than a maskable interrupt source. The last interrupt (external input to the $\overline{XIRQ}$ pin) is considered as a non-maskable interrupt because once enabled, it cannot be masked by software; however it is masked during reset and upon receipt of an interrupt at the $\overline{XIRQ}$ pin. Table 2 provides a list of each interrupt, its vector location in ROM, and the actual condition code register bit that masks it. A discussion of the various interrupts is provided below.

TABLE 2 — INTERRUPT VECTOR ASSIGNMENTS

Vector Address	Interrupt Source	Masked By
FFC0, C1	Reserved	—
FFD4, D5	Reserved	—
FFD6, D7	SCI Serial System	I Bit
FFD8, D9	SPI Serial Transfer Complete	I Bit
FFDA, DB	Pulse Accumulator Input Edge	I Bit
FFDC, DD	Pulse Accumulator Overflow	I Bit
FFDE, DF	Timer Overflow	I Bit
FFE0, E1	Timer Output Compare 5	I Bit
FFE2, E3	Timer Output Compare 4	I Bit
FFE4, E5	Timer Output Compare 3	I Bit
FFE6, E7	Timer Output Compare 2	I Bit
FFE8, E9	Timer Output Compare 1	I Bit
FFEA, EB	Timer Input Capture 3	I Bit
FFEC, ED	Timer Input Capture 2	I Bit
FFEE, EF	Timer Input Capture 1	I Bit
FFF0, F1	Real Time Interrupt	I Bit
FFF2, F3	$\overline{IRQ}$ (External Pin or Parallel I/O)	I Bit
FFF4, F5	$\overline{XIRQ}$ Pin (Pseudo Non-maskable Interrupt)	X Bit
FFF6, F7	SWI	None
FFF8, F9	Illegal Op-Code Trap	None
FFFA, FB	COP Failure (Reset)	None
FFFC, FD	COP Clock Monitor Fail (Reset)	None
FFFE, FF	$\overline{RESET}$	None

TIMER INTERRUPTS

The timer system provides nine of the fifteen interrupt possibilities: five output compare interrupts, three input capture interrupts, and a timer overflow interrupt.

The timer contains five 16-bit output compare registers which are program controlled and may be loaded with a number between $0000-$FFFF. The value in each output compare register is then compared to a 16-bit comparator, which is loaded from the timer free running counter, during each clock cycle. If a match is found between the 16-bit comparator value and the output compare register value, the corresponding output compare flag is set. When the output compare flag is set, a corresponding output compare interrupt may be generated and/or an external output may be generated at the corresponding port A pin(s). Port A outputs PA3 through PA7 are used as output pins for output compare functions OC1 through OC5.

In addition to the five output compare interrupts, the timer also provides for three input capture interrupts. The timer contains three 16-bit latch registers which are used to latch the value of the free running counter (in the timer) when an input capture edge is applied to the corresponding PA0-PA2 pin. The value of the free running counter is latched into the corresponding input capture register and an internal interrupt may be generated. The interrupt routine can then read the storage register and determine the time at which the input capture was detected.

The timer may also provide an interrupt when the free running counter changes value from $FFFF to $0000 (overflow). The 16-bit free running counter repeats this change once for every 65,536 inputs from a prescaler circuit. The prescaler is programmable for either divide-by-1, divide-by-4, divide-by-8, or divide-by-16 of the MCU E-clock. Thus, the prescaler extends the actual range of the free running counter and the time between timer overflow interrupts from 2^{16} to 2^{256} E-clock inputs to the prescaler.

REAL TIME INTERRUPTS

The real time interrupt is a maskable interrupt that occurs periodically at a rate of $E/2^{13}$, $E/2^{14}$, $E/2^{15}$, or $E/2^{16}$.

EXTERNAL INTERRUPTS

Two external interrupts are accessable using the $\overline{IRQ}$ and the $\overline{XIRQ}$ pins. The $\overline{IRQ}$ interrupt is a maskable interrupt while the $\overline{XIRQ}$ interrupt is considered a non-maskable interrupt; however, the $\overline{XIRQ}$ interrupt is masked during reset and immediately following receipt of an $\overline{XIRQ}$ interrupt signal. These interrupts are controlled by the I and X bits in the condition code register as discussed in **CENTRAL PROCESSING UNIT.**

SOFTWARE INTERRUPT (SWI)

The software interrupt is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the condition code register set). The SWI instruction is executed similar to other maskable interrupts in that it sets the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once it is fetched no other interrupt can be honored until the first instruction in the SWI service routine is completed.

SERIAL PERIPHERAL INTERFACE (SPI) INTERRUPT

A serial peripheral interface (SPI) interrupt is generated when a serial data transfer between the MC68HC11A4 and an external device has been completed. This interrupt is masked if the condition code register I bit is set.

SERIAL COMMUNICATIONS INTERFACE (SCI) INTERRUPT

A serial communications interface (SCI) interrupt is generated if any one of the following occurs in the SCI:

1. Transmit data register is empty
2. Transmission of data is complete
3. Receive data register is full or an overflow occurred in the receive data register
4. Idle line detected by receiver.

The SCI interrupt is masked if the condition code register I bit is set.

PULSE ACCUMULATOR INTERRUPT

The pulse accumulator contains an 8-bit counter which is program controlled to either count input pulses (event counting) at PA7 or to count internal E/64 clocks subject to an enable signal at PA7 (gated time accumulation). When the counter has an overflow from $FF to $00 a pulse accumulator overflow interrupt is generated provided the I bit in the condition code register is clear.

When the input to the pulse accumulator is a gate input at PA7 for counting internal E/64 clocks, the trailing edge of the gate signal (end of counting cycle) can generate an interrupt. This pulse accumulator input edge interrupt is generated provided the I bit in the condition code register is clear. Refer to **PULSE ACCUMULATOR** for more information.

PARALLEL I/O INTERRUPT

The parallel I/O subsystem can generate an interrupt which uses the same vector as the $\overline{IRQ}$ interrupt. The purpose of sharing the $\overline{IRQ}$ vector is to allow external emulation of the parallel I/O subsystem in expanded multiplexed modes.

RESETS

The MC68HC11A4 MCU has four possible types of reset: an active low external reset pin ($\overline{RESET}$), a power-on reset function, a computer operating properly (COP) watchdog timer reset, and a clock monitor reset.

$\overline{RESET}$ PIN

The $\overline{RESET}$ pin is used to reset the MCU to provide an orderly software startup procedure. To request an external reset, the $\overline{RESET}$ pin must be held low for eight E_{cyc} (two E_{cyc} if internal resets are not used).

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in power supply voltage. There is no provision for power-down reset. If the external $\overline{RESET}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{RESET}$ goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MC68HC11A4 MCU contains a watchdog timer which will time itself out if not reset within a specific time by a program reset sequence. If for any reason the COP watchdog timer is allowed to timeout, it generates an MCU reset which is functionally similar to pulling the $\overline{RESET}$ pin low.

A control bit, which is implemented in an EEPROM cell of the system configuration register, is used to enable (or disable) the COP reset function. When this bit is clear, the COP reset function is disabled; if set, the COP reset is enabled.

CLOCK MONITOR RESET

The MC68HC11A4 MCU contains a clock monitor circuit which measures the E-clock input frequency. If the E clock input rate is high enough, then the clock monitor does not time out. However, if the E clock signal is lost, or its frequency falls below 200 kHz, then an MCU reset is generated which is functionally similar to pulling the $\overline{RESET}$ pin low.

A read-write control bit, which is implemented in the system configuration options register, is used to enable (or disable) the clock monitor reset. When this bit is clear, the clock monitor reset function is disabled; when set, the clock monitor reset is enabled.

STOP AND WAIT

The MC68HC11A4 MCU contains two programmable low-power operating modes; stop and wait. In the wait mode, the on-chip oscillator remains active together with other functions discussed below. In the stop mode, all clocks including the crystal oscillator are stopped.

WAI (WAIT) INSTRUCTION

The WAI instruction places the MC68HC11A4 MCU in a low power consumption (wait) mode. In the wait mode, the internal clock remains active, and the MCU enters one of four different variations of the wait mode. These variations, which depend upon the I bit in the condition register and whether or not the COP circuit is required in the system, include: (1) only the CPU turned off; (2) CPU and the E clock output buffer turned off; (3) CPU and timer system turned off; or (4) CPU, E output, and timer system all off.

During the wait mode, the CPU registers are stacked and processing is suspended until a qualified interrupt is detected. The actual qualified interrupt type is dependent upon which of the wait mode variations is selected. The qualified interrupt(s) required to bring the MCU out of the wait mode for each of the wait mode variations is shown below. In all cases, reset brings the MCU out of the wait mode; however, as in all resets, the system is reset and the start of MCU operation is determined by the reset vector.

Wait Mode Variation	Qualified Interrupt
Only CPU Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$, Any Internal Interrupt
CPU and E Clock Output Buffers Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$, Any Internal Interrupt
CPU and Timer System Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$
CPU, E Clock Output Buffers, and Timer System Turned Off	$\overline{IRQ}$, $\overline{XIRQ}$

STOP INSTRUCTION

The STOP instruction places the MC68HC11A4 MCU in its lowest power consumption mode provided the S bit in the condition code register is clear. In the stop mode all clocks including the internal oscillator are stopped, causing all internal processing to be halted. To exit the stop mode and resume normal processing, a low level must be applied to one of the external interrupt pins ($\overline{IRQ}$ or $\overline{XIRQ}$) or to the $\overline{RESET}$ pin. If an external interrupt is used at the $\overline{IRQ}$ input, it is only effective if the I bit in the condition code register is clear. If an external interrupt is applied at the $\overline{XIRQ}$ input, the MCU exits from the stop mode regardless of the state of the X bit in condition code register; however, the actual recovery sequence differs depending on the X bit. If the X bit is clear, the MCU starts up with the stacking sequence leading to normal service of the XIRQ request. If the X bit is set, then processing will continue with the instruction immediately following the STOP instruction and no $\overline{XIRQ}$ interrupt service routine is requested. As in the wait mode, a low input to the $\overline{RESET}$ pin will always result in an exit from the stop mode and the start of MCU operation is determined by the reset vector.

Since the oscillator is stopped in the stop mode, a restart delay may be required to allow for oscillator stabilization when exiting from the stop mode. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, a control bit within the MCU may be used (cleared) to bypass the delay. If the delay bypass control bit is clear then the $\overline{RESET}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

PROGRAMMABLE TIMER SYSTEM

The timer system in the MC68HC11A4 uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free running counter. The free running counter is clocked by the output of a programmable prescaler (divide-by-1, 4, 8, or 16) which is in turn clocked by the MCU E clock. Functions available within the MC68HC11A4 timer include: three input capture functions and five output compare functions.

The capabilities of the programmable timer are obtained using the following registers:

1. Prescaler (divide-by-1, 4, 8, or 16)
2. Free Running Counter (16-bit)
3. Input Capture (three 16-bit registers)
4. Output Compare (five 16-bit registers)
5. Main Timer Control and Status Registers

PRESCALER AND FREE RUNNING COUNTER

The key element in the timer system is a 16-bit free running counter with its associated programmable prescaler (divide-by-1, 4, 8, or 16). The free running counter is clocked by the output of the prescaler which is in turn clocked by the E clock. The free running counter can be read by software at any time without affecting its value since it is clocked and read on opposite half cycles of the MPU E clock. The free running counter is cleared to $0000 during reset and is a read-only register (except in the test or bootstrap mode where this feature is used in factory testing).

The 16-bit free running counter repeats every 65,536 counts (prescaler output) and when the count changes from $FFFF to $0000 a timer overflow flag bit is set. Setting the timer overflow flag bit also generates an internal interrupt if the overflow interrupt enable bit is set.

Input Capture Functions

There are three separate 16-bit read-only input capture registers which are not affected by reset. Each of these registers is used to latch the value of the free running counter when a selected transition at an external pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Output Compare Functions

There are five separate 16-bit read/write output compare registers which are initialized to $FFFF at reset. The value written into the output compare register is compared to the free running counter value during each MCU E clock cycle. If a match is found between the two values, the particular output compare flag bit is set and an interrupt is generated provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For OC1, the output action to be taken, when a match is found, is controlled by a 5-bit mask register and a 5-bit data register. The 5-bit mask register specifies which timer port outputs are to be affected and the 5-bit data register specifies the data to be placed on the affected output pins. For OC2 through OC5, one specific timer output is affected as controlled by four 2-bit fields in a timer control register. Specific actions include: (1) timer disconnect from output pin logic, (2) toggle output compare line, (3) clear output compare line to zero, or (4) set output compare line to one.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes depending on the state of a control bit. These include the event counting mode or the gated time accumulation mode.

The pulse accumulator control register contains four bits which enable and configure the pulse accumulator system. One bit enables the counter. One bit determines whether the

PA7/PAI pin will be an input or an output. A third bit specifies the event counting mode or the gated time accumulation mode, and the fourth bit determines which edge of the PAI input is the active one. The 8-bit counter counts from $00 to $FF and when it overflows from $FF to $00 a flag bit is set. This results in a hardware interrupt provided the pulse accumulator overflow interrupt enable bit is set.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external (PAI) pin input (PA7). In the gated time accumulation mode, the 8-bit counter is clocked to increasing values by the MCU E clock (divided-by-64) provided the proper gating signal is applied to an external (PAI) pin input (PA7).

SERIAL COMMUNICATIONS INTERFACE (SCI)

The serial communications interface (SCI) allows the MC68HC11A4 to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI in the MC68HC11A4 is provided with a standard NRZ format with a variety of baud rates. The baud rate is derived from the crystal clock circuit and interface with peripheral devices is accomplished using port D pins. PD0 for receive data (RxD) and PD1 for transmit data (TxD).

BAUD RATE GENERATION

The actual baud rate generation circuit contains a programmable prescaler and divider which is clocked by the MCU E clock. A programmable baud rate register is used to provide the various divide ratios used in the baud rate generator prescaler and divider. This scheme of baud rate generation allows for selection of many different standard baud rates, all of which are controlled by the crystal oscillator.

DATA FORMAT

Receive data (RxD) in or transmit data (TxD) out is the serial data which is presented between the input pin (PD0) and the internal data bus, and between the internal data bus and the output pin (PD1). The data format requires:

1. An idle line which is in the high state (logic one) prior to transmission/reception of a message.
2. A start bit (logic zero) which is transmitted/received indicating the start of a message.
3. Data is transmitted and received least-significant bit first.
4. A stop bit (logic one in the tenth or eleventh bit position) indicates the byte is complete.
5. A break is defined as the transmission or reception of a logic zero for some multiple of the data format.

The data format word length may consist of either ten or eleven bits. Selection of the word length is controlled by a single bit in a control register within the SCI. If this control bit is clear, the data contains a start bit, eight data bits, and a stop bit. If this control bit is set, there is a start bit, nine data bits, and a stop bit.

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This is referred to as a double buffered system in that besides the character being shifted out serially, another character is already waiting to be loaded into the serial shift register. The output of the transmit serial shift-register is applied to the TxD output pin (PD1) as long as a transmit enable bit is set.

RECEIVE OPERATION

Receive data in (RxD) is serial data which is presented to the input pin (PD0). An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. In this manner the data input can be selectively sampled to detect receive data and then verify that the data is valid. Data is received in a serial shift register and is transferred to a parallel register as a complete byte. This is referred to as a double buffered system in that besides the character already in the parallel register, another is being shifted in serially.

WAKE-UP FEATURE

The wake-up feature allows a receiver(s) to "sleep" until a specific action takes place. In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of a message. This wake-up feature allows uninterested MPUs to ignore incoming messages. The MC68HC11A4 SCI permits this wake-up feature by either of two methods: idle line wake-up or address mark wake-up.

In idle line wake-up, all receivers wake up whenever an idle line is detected; however, if a receiver does not recognize its address in the first frame of a message it may ignore the rest of the message by invoking the wake-up feature. In this wake-up method, transmitter software must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

In the address mark wake-up, all serial frames consist of seven (or eight) information bits plus a most-significant bit (MSB) which is used to indicate an address frame if the MSB is a logic one. The first frame of each message is an address frame which wakes up all receivers in the system. All receivers evaluate this marked address frame to determine which receiver(s) the message is intended for. If a receiver determines that a message is not intended for it, it invokes the receiver wake-up function so that no additional program overhead is required for the rest of the message.

INTERRUPT FLAGS

The serial communications interface (SCI) generates a hardware interrupt (SCI interrupt) whenever any one of several flags is set and its corresponding interrupt enable bit is also set. These flags which are discussed below include:

1. Transmit Data register empty
2. Transmission complete
3. Idle line detected
4. Receive data register full or overrun error detected.

The transmit data register empty (TDRE) bit is set to indicate that the transmit parallel data register contents have been transferred to the transmit serial shift register. If the corresponding interrupt enable bit (transmit interrupt enable) is set then an SCI interrupt is generated.

The transmission complete (TC) bit is set when the transmitter no longer has any meaningful information to transmit; i.e., no data in the serial shifter, no queued preamble, and no queued break. If the transmitter is enabled when TC is set, the serial line will go idle (continuous mark).

The idle line detected (IDLE) bit is set whenever a receiver detects a receiver idle line. This could indicate the end of a message, the preamble of a new message, or resynchronization with the transmitter. If the corresponding interrupt enable bit (idle line interrupt enable) is set then an SCI interrupt is generated.

The receiver data register full (RDRF) bit is set whenever the receiver serial shift register contents are transferred to the serial communications data register. If the corresponding interrupt enable bit (receive interrupt enable) is set then an SCI interrupt is generated.

The overrun error bit is set to indicate that the next byte is ready for transfer from the receive shift register to the receive data register but that register is already full (RDRF bit set). Data transfer is then inhibited until the OR (overrun) bit is cleared. As with the RDRF bit, an SCI interrupt is generated if the corresponding interrupt enable bit is set.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) allows several MC68HC11A4 MCUs, or MC68HC11A4 MCUs plus peripheral devices, to be interconnected within a single "black box", on the same printed circuit board. In a serial peripheral interface, the MC68HC11A4 provides such features as:

- Full Duplex, Two, Three, or Four Wire Synchronous Transfers
- Master or Slave Operation
- Interface With Low Cost "Dumb" Peripherals
- Interface With Intelligent Peripherals on Master/Slave Basis
- Four Programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End of Transmission Interrupt Flag
- Write Collision Error Detection
- Master-Master Mode Fault Error Detection

Four basic signal lines are associated with the SPI system. These include a master-out-slave-in (MOSI) line; a master-in-slave-out (MISO) line; a serial clock (SCK) line; and a slave select ($\overline{SS}$) line. Two master-slave system configurations are shown in Figure 5 and the basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described below.

MASTER OUT SLAVE IN (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device. In this manner data is transferred serially from a master to a slave on this line; most significant bit first, least significant last.

MASTER IN SLAVE OUT (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant last.

SLAVE SELECT ($\overline{SS}$)

The slave select ($\overline{SS}$) is a fixed input which receives an active low signal that is generated by a master device to enable slave devices to accept data.

SERIAL CLOCK (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI or MISO pins. The master and slave devices can exchange a byte of information during a sequence of eight clock pulses. The SCK is generated by the controlling master device and becomes an input on all slave devices to synchronize slave data transfer.

ANALOG-TO-DIGITAL (A/D) CONVERTER

The MC68HC11A4 contains an 8-channel, multiplexed input, successive approximation analog-to-digital converter with sample and hold. Two dedicated pins (V_{REFL}, V_{REFH}) are provided for the reference supply voltage input. These dedicated pins are used instead of the device power pins to increase accuracy of the A/D conversion.

The 8-bit A/D conversions of the MC68HC11A4 are accurate to within ± one LSB (± ½ LSB quantizing error and ± ½ LSB non-linearity error). Each conversion is accomplished in 50 MCU E clock cycles or less. An internal control bit allows selection of an internal conversion clock oscillator which allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 25 to 50 microseconds to complete.

NOTE

In the 48-pin dual in-line package, four conversion channels are not implemented. These include channels four through seven.

ADDRESSING MODES

Six addressing modes can be used to reference memory; they include: immediate, direct, extended, indexed (with either of two 16-bit index registers and an 8-bit offset), inherent, and relative. Some instructions require an additional byte before the opcode to accommodate a multi-page opcode map; this byte is called a prebyte.

The following paragraphs provide a description of each addressing mode plus a discussion of the prebyte. In these descriptions the term effective address is used to indicate the address in memory from which the argument is fetched or stored, or from which execution is to proceed.

FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 1 of 2)

a. Single Master, Four Slaves

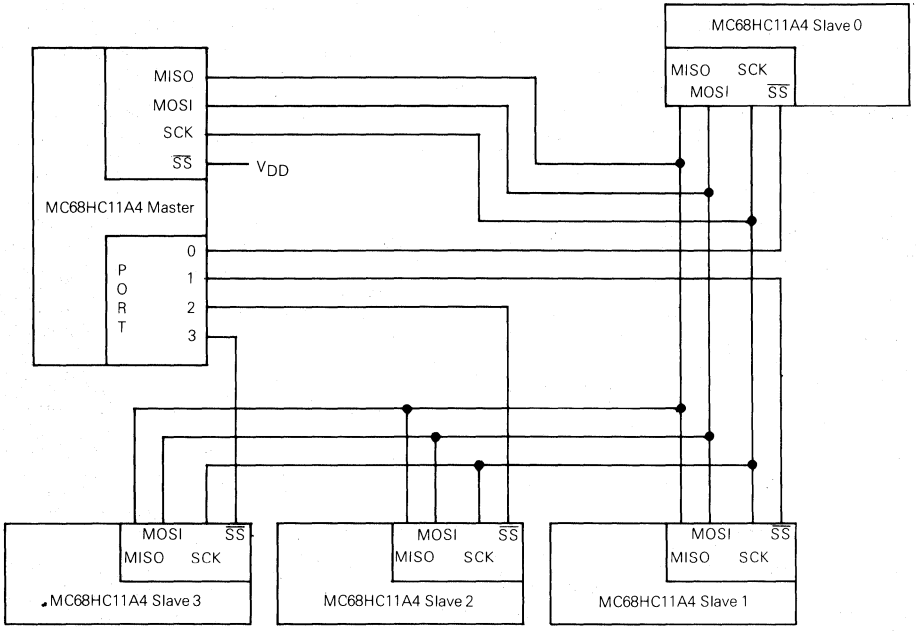
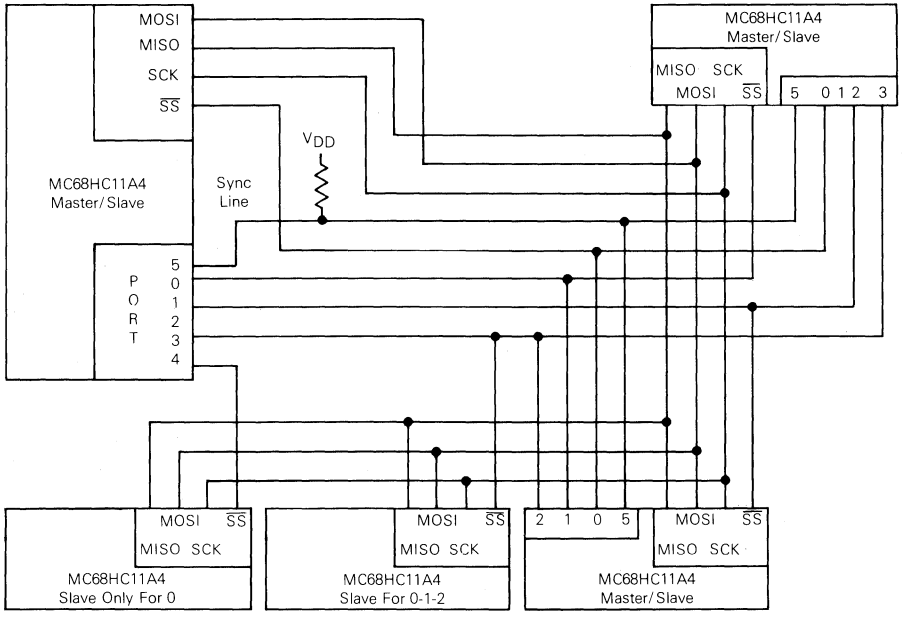


FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 2 of 2)

b. Three Master/Slave, Two Slaves



IMMEDIATE ADDRESSING

In the immediate addressing mode, the actual argument is contained in the byte(s) immediately following the instruction where the number of bytes matches the size of the register. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT ADDRESSING

In the direct addressing mode, the least significant byte of the operand address is contained in a single byte following the opcode and the most significant byte is assumed to be $00. Direct addressing allows the user to access addresses $0000 through $00FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. These are usually two or three (if prebyte is required) byte instructions.

EXTENDED ADDRESSING

In the extended addressing mode, the second and third bytes following the opcode contain the absolute address of the operand. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED ADDRESSING

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case the effective address is variable and depends on two factors: (1) the current contents of the index register (X or Y) being used, and (2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

INHERENT ADDRESSING

In the inherent addressing mode, all of the information to execute the instruction is contained in the opcode. The operands (if any) are registers and no memory reference is required. These are usually one or two byte instructions.

RELATIVE ADDRESSING

The relative addressing mode is used for branch instructions. If the branch condition is true and contents of the 8-bit signed byte following the opcode (the offset) is added to the contents of the program counter to form the effective branch address; otherwise, control proceeds to the next instruction. These are usually two byte instructions.

PREBYTE

In order to expand the number of instructions used in the MC68HC11A4, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. The opcode instructions which do not require a prebyte could be considred as page 1 of the overall opcode map. The remaining opcodes could be considered as pages 2, 3, and 4 of the opcode map and would require a prebyte; $18 for page 2, $1A for page 3, and $CD for page 4. Refer to **INSTRUCTION SUMMARY** for more detail.

INSTRUCTION SET

The central processing unit (CPU) in the MC68HC11A4 is basically a proper extension of the MC6801 CPU. In addition to its ability to execute all M6800 and M6801 instructions, the MC68HC11A4 CPU has a paged operation code (opcode) map with a total of 91 new opcodes. Major functional additions include a second 16-bit index register (Y register), two types of 16-by-16 divide instructions, a STOP instruction, and bit manipulation instructions.

Table 3 shows all MC68HC11A4 instructions in all possible addressing modes. For each instruction the operand construction is shown as well as the total number of machine code bytes and execution time in CPU E-clock cycles. Notes are provided at the end of Table 3 which explain the letters in the Operand and Execution Time columns of some instructions.

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
ABA	INH	1B		1	2
ABX	INH	3A		1	3
ABY	INH	18 3A		2	4
ADCA (opr)	A IMM	89	ii	2	2
	A DIR	99	dd	2	3
	A EXT	B9	hh ll	3	4
	A IND, X	A9	ff	2	4
	A IND, Y	18 A9	ff	3	5
ADCB (opr)	B IMM	C9	ii	2	2
	B DIR	D9	dd	2	3
	B EXT	F9	hh ll	3	4
	B IND, X	E9	ff	2	4
	B IND, Y	18 E9	ff	3	5
ADDA (opr)	A IMM	8B	ii	2	2
	A DIR	9B	dd	2	3
	A EXT	BB	hh ll	3	4
	A IND, X	AB	ff	2	4
	A IND, Y	18 AB	ff	3	5
ADDB (opr)	B IMM	CB	ii	2	2
	B DIR	DB	dd	2	3
	B EXT	FB	hh ll	3	4
	B IND, X	EB	ff	2	4
	B IND, Y	18 EB	ff	3	5
ADDD (opr)	IMM	C3	jj kk	3	4
	DIR	D3	dd	2	5
	EXT	F3	hh ll	3	6
	IND, X	E3	ff	2	6
	IND, Y	18 E3	ff	3	7
ANDA (opr)	A IMM	84	ii	2	2
	A DIR	94	dd	2	3
	A EXT	B4	hh ll	3	4
	A IND, X	A4	ff	2	4
	A IND, Y	18 A4	ff	3	5
ANDB (opr)	B IMM	C4	ii	2	2
	B DIR	D4	dd	2	3
	B EXT	F4	hh ll	3	4
	B IND, X	E4	ff	2	4
	B IND, Y	18 E4	ff	3	5
ASL (opr)	EXT	78	hh ll	3	6
	IND, X	68	ff	2	6
	IND, Y	18 68	ff	3	7
ASLA	A INH	48		1	2
ASLB	B INH	58		1	2
ASLD	INH	05		1	3
ASR (opr)	EXT	77	hh ll	3	6
	IND, X	67	ff	2	6
	IND, Y	18 67	ff	3	7
ASRA	A INH	47		1	2
ASRB	B INH	57		1	2
BCC (rel)	REL	24	rr	2	3
BCLR (opr) (msk)	DIR	15	dd mm	3	6
	IND, X	1D	ff mm	3	7
	IND, Y	18 1D	ff mm	4	8
BCS (rel)	REL	25	rr	2	3
BEQ (rel)	REL	27	rr	2	3
BGE (rel)	REL	2C	rr	2	3
BGT (rel)	REL	2E	rr	2	3

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
BHI (rel)	REL	22	rr	2	3
BHS (rel)	REL	24	rr	2	3
BITA (opr)	A IMM	85	ii	2	2
	A DIR	95	dd	2	3
	A EXT	B5	hh ll	3	4
	A IND, X	A5	ff	2	4
	A IND, Y	18 A5	ff	3	5
BITB (opr)	B IMM	C5	ii	2	2
	B DIR	D5	dd	2	3
	B EXT	F5	hh ll	3	4
	B IND, X	E5	ff	2	4
	B IND, Y	18 E5	ff	3	5
BLE (rel)	REL	2F	rr	2	3
BLO (rel)	REL	25	rr	2	3
BLS (rel)	REL	23	rr	2	3
BLT (rel)	REL	2D	rr	2	3
BMI (rel)	REL	2B	rr	2	3
BNE (rel)	REL	26	rr	2	3
BPL (rel)	REL	2A	rr	2	3
BRA (rel)	REL	20	rr	2	3
BRN (rel)	REL	21	rr	2	3
BRCLR (opr) (msk) (rel)	DIR	13	dd mm rr	4	6
	IND, X	1F	ff mm rr	4	7
	IND, Y	18 1F	ff mm rr	5	8
BRSET (opr) (msk) (rel)	DIR	12	dd mm rr	4	6
	IND, X	1E	ff mm rr	4	7
	IND, Y	18 1E	ff mm rr	5	8
BSET (opr) (msk)	DIR	14	dd mm	3	6
	IND, X	1C	ff mm	3	7
	IND, Y	18 1C	ff mm	4	8
BSR (rel)	REL	8D	rr	2	6
BVC (rel)	REL	28	rr	2	3
BVS (rel)	REL	29	rr	2	3
CBA	INH	11		1	2
CLC	INH	0C		1	2
CLI	INH	0E		1	2
CLR (opr)	EXT	7F	hh ll	3	6
	IND, X	6F	ff	2	6
	IND, Y	18 6F	ff	3	7
CLRA	A INH	4F		1	2
CLRB	B INH	5F		1	2
CLV	INH	0A		1	2
CMPA (opr)	A IMM	81	ii	2	2
	A DIR	91	dd	2	3
	A EXT	B1	hh ll	3	4
	A IND, X	A1	ff	2	4
	A IND, Y	18 A1	ff	3	5
CMPB (opr)	B IMM	C1	ii	2	2
	B DIR	D1	dd	2	3
	B EXT	F1	hh ll	3	4
	B IND, X	E1	ff	2	4
	B IND, Y	18 E1	ff	3	5

TABLE 3 – MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
COM (opr)	EXT	73	hh ll	3	6
	IND, X	63	ff	2	6
	IND, Y	18 63	ff	3	7
COMA	A INH	43		1	2
COMB	B INH	53		1	2
CPD (opr)	IMM	1A 83	jj kk	4	5
	DIR	1A 93	dd	3	6
	EXT	1A B3	hh ll	4	7
	IND, X	1A A3	ff	3	7
	IND, Y	CD A3	ff	3	7
CPX (opr)	IMM	8C	jj kk	3	4
	DIR	9C	dd	2	5
	EXT	BC	hh ll	3	6
	IND, X	AC	ff	2	6
	IND, Y	CD AC	ff	3	7
CPY (opr)	IMM	18 8C	jj kk	4	5
	DIR	18 9C	dd	3	6
	EXT	18 BC	hh ll	4	7
	IND, X	1A AC	ff	3	7
	IND, Y	18 AC	ff	3	7
DAA	INH	19		1	2
DEC (opr)	EXT	7A	hh ll	3	6
	IND, X	6A	ff	2	6
	IND, Y	18 6A	ff	3	7
DECA	A INH	4A		1	2
DECB	B INH	5A		1	2
DES	INH	34		1	3
DEX	INH	09		1	3
DEY	INH	18 09		2	4
EORA (opr)	A IMM	88	ii	2	2
	A DIR	98	dd	2	3
	A EXT	B8	hh ll	3	4
	A IND, X	A8	ff	2	4
	A IND, Y	18 A8	ff	3	5
EORB (opr)	B IMM	C8	ii	2	2
	B DIR	D8	dd	2	3
	B EXT	F8	hh ll	3	4
	B IND, X	E8	ff	2	4
	B IND, Y	18 E8	ff	3	5
FDIV	INH	03		1	41
IDIV	INH	02		1	41
INC (opr)	EXT	7C	hh ll	3	6
	IND, X	6C	ff	2	6
	IND, Y	18 6C	ff	3	7
INCA	A INH	4C		1	2
INCB	B INH	5C		1	2
INS	INH	31		1	3
INX	INH	08		1	3
INY	INH	18 08		2	4
JMP (opr)	EXT	7E	hh ll	3	3
	IND, X	6E	ff	2	3
	IND, Y	18 6E	ff	3	4
JSR (opr)	DIR	9D	dd	2	5
	EXT	BD	hh ll	3	6
	IND, X	AD	ff	2	6
	IND, Y	18 AD	ff	3	7

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
LDAA (opr)	A IMM	86	ii	2	2
	A DIR	96	dd	2	3
	A EXT	B6	hh ll	3	4
	A IND, X	A6	ff	2	4
	A IND, Y	18 A6	ff	3	5
LDAB (opr)	B IMM	C6	ii	2	2
	B DIR	D6	dd	2	3
	B EXT	F6	hh ll	3	4
	B IND, X	E6	ff	2	4
	B IND, Y	18 E6	ff	3	5
LDD (opr)	IMM	CC	jj kk	3	3
	DIR	DC	dd	2	4
	EXT	FC	hh ll	3	5
	IND, X	EC	ff	2	5
	IND, Y	18 EC	ff	3	6
LDS (opr)	IMM	8E	jj kk	3	3
	DIR	9E	dd	2	4
	EXT	BE	hh ll	3	5
	IND, X	AE	ff	2	5
	IND, Y	18 AE	ff	3	6
LDX (opr)	IMM	CE	jj kk	3	3
	DIR	DE	dd	2	4
	EXT	FE	hh ll	3	5
	IND, X	EE	ff	2	5
	IND, Y	CD EE	ff	3	6
LDY (opr)	IMM	18 CE	jj kk	4	4
	DIR	18 DE	dd	3	5
	EXT	18 FE	hh ll	4	6
	IND, X	1A EE	ff	3	6
	IND, Y	18 EE	ff	3	6
LSL (opr)	EXT	78	hh ll	3	6
	IND, X	68	ff	2	6
	IND, Y	18 68	ff	3	7
LSLA	A INH	48		1	2
LSLB	B INH	58		1	2
LSLD	INH	05		1	3
LSR (opr)	EXT	74	hh ll	3	6
	IND, X	64	ff	2	6
	IND, Y	18 64	ff	3	7
LSRA	A INH	44		1	2
LSRB	B INH	54		1	2
LSRD	INH	04		1	3
MUL	INH	3D		1	10
NEG (opr)	EXT	70	hh ll	3	6
	IND, X	60	ff	2	6
	IND, Y	18 60	ff	3	7
NEGA	A INH	40		1	2
NEGB	B INH	50		1	2
NOP	INH	01		1	2
ORAA (opr)	A IMM	8A	ii	2	2
	A DIR	9A	dd	2	3
	A EXT	BA	hh ll	3	4
	A IND, X	AA	ff	2	4
	A IND, Y	18 AA	ff	3	5
ORAB (opr)	B IMM	CA	ii	2	2
	B DIR	DA	dd	2	3
	B EXT	FA	hh ll	3	4
	B IND, X	EA	ff	2	4
	B IND, Y	18 EA	ff	3	5

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
PSHA	A INH	36		1	3
PSHB	B INH	37		1	3
PSHX	INH	3C		1	4
PSHY	INH	18 3C		2	5
PULA	A INH	32		1	4
PULB	B INH	33		1	4
PULX	INH	38		1	5
PULY	INH	18 38		2	6
ROL (opr)	EXT	79	hh ll	3	6
	IND, X	69	ff	2	6
	IND, Y	18 69	ff	3	7
ROLA	A INH	49		1	2
ROLB	B INH	59		1	2
ROR (opr)	EXT	76	hh ll	3	6
	IND, X	66	ff	2	6
	IND, Y	18 66	ff	3	7
RORA	A INH	46		1	2
RORB	B INH	56		1	2
RTI	INH	3B		1	12
RTS	INH	39		1	5
SBA	INH	10		1	2
SBCA (opr)	A IMM	82	ii	2	2
	A DIR	92	dd	2	3
	A EXT	B2	hh ll	3	4
	A IND, X	A2	ff	2	4
	A IND, Y	18 A2	ff	3	5
SBCB (opr)	B IMM	C2	ii	2	2
	B DIR	D2	dd	2	3
	B EXT	F2	hh ll	3	4
	B IND, X	E2	ff	2	4
	B IND, Y	18 E2	ff	3	5
SEC	INH	0D		1	2
SEI	INH	0F		1	2
SEV	INH	0B		1	2
STAA (opr)	A DIR	97	dd	2	3
	A EXT	B7	hh ll	3	4
	A IND, X	A7	ff	2	4
	A IND, Y	18 A7	ff	3	5
STAB (opr)	B DIR	D7	dd	2	3
	B EXT	F7	hh ll	3	4
	B IND, X	E7	ff	2	4
	B IND, Y	18 E7	ff	3	5
STD (opr)	DIR	DD	dd	2	4
	EXT	FD	hh ll	3	5
	IND, X	ED	ff	2	5
	IND, Y	18 ED	ff	3	6

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Machine Code Bytes (Total)	Execution Time (Cycles)
STOP	INH	CF		1	2
STS (opr)	DIR	9F	dd	2	4
	EXT	BF	hh ll	3	5
	IND, X	AF	ff	2	5
	IND, Y	18 AF	ff	3	6
STX (opr)	DIR	DF	dd	2	4
	EXT	FF	hh ll	3	5
	IND, X	EF	ff	2	5
	IND, Y	CD EF	ff	3	6
STY (opr)	DIR	18 DF	dd	3	5
	EXT	18 FF	hh ll	4	6
	IND, X	1A EF	ff	3	6
	IND, Y	18 EF	ff	3	6
SUBA (opr)	A IMM	80	ii	2	2
	A DIR	90	dd	2	3
	A EXT	B0	hh ll	3	4
	A IND, X	A0	ff	2	4
	A IND, Y	18 A0	ff	3	5
SUBB (opr)	B IMM	C0	ii	2	2
	B DIR	D0	dd	2	3
	B EXT	F0	hh ll	3	4
	B IND, X	E0	ff	2	4
	B IND, Y	18 E0	ff	3	5
SUBD (opr)	IMM	83	jj kk	3	4
	DIR	93	dd	2	5
	EXT	B3	hh ll	3	6
	IND, X	A3	ff	2	6
	IND, Y	18 A3	ff	3	7
SWI	INH	3F		1	14
TAB	INH	16		1	2
TAP	INH	06		1	2
TBA	INH	17		1	2
TEST	INH	00		1	*
TPA	INH	07		1	2
TST (opr)	EXT	7D	hh ll	3	6
	IND, X	6D	ff	2	6
	IND, Y	18 6D	ff	3	7
TSTA	A INH	4D		1	2
TSTB	B INH	5D		1	2
TSX	INH	30		1	3
TSY	INH	18 30		2	4
TXS	INH	35		1	3
TYS	INH	18 35		2	4
WAI	INH	3E		2	14 + n**
XGDX	INH	8F		1	3
XGDY	INH	18 8F		2	4

*-infinity or until reset occurs.

**-12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally two additional cycles are used to fetch the appropriate interrupt vector.

dd = 8-bit direct address ($0000-$00FF) (high byte assumed to be $00)
ff = 8-bit positive offset $00 (0) to $FF (255) (is added to index)
hh = high order byte of 16-bit extended address
ii = one byte of immediate data
jj = high order byte of 16-bit immediate data
kk = low order byte of 16-bit immediate data
ll = low order byte of 16-bit extended address
mm = 1-byte bit mask (set bits to be affected)
rr = signed relative offset $80 (−128) to $7F (+127) (offset relative to the address following the machine code offset byte)

MOTOROLA

MC68HC68A1

Product Preview

HCMOS
(HIGH-DENSITY CMOS SILICON-GATE)

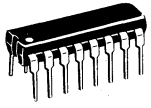
SERIAL 10-BIT ANALOG-TO-DIGITAL CONVERTER

SERIAL 10-BIT ANALOG-TO-DIGITAL CONVERTER

The MC68HC68A1 is an HCMOS serial 10-bit analog-to-digital (A/D) converter. Interface to the A/D converter is through a standard serial peripheral interface (SPI) unit. This device can interface directly with the MC68HC05D2 microcomputer without additional components.

The following is a summary of the features offered by the MC68HC68A1.

- 10 Bits of Resolution
- Eight Bits of Accuracy
- Serial Peripheral Interface Capability
- Conversion Time 100 μs Maximum
- Eight Analog Input Channels
- Common Mode V_{DD}-V_{SS}
- Continuous or Single-Channel Scan
- Sample and Hold Capability
- Schmitt Oscillator Clock Input



P SUFFIX
PLASTIC PACKAGE
CASE 648

GENERAL DESCRIPTION

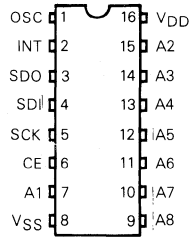
The MC68HC68A1 is an HCMOS 10-bit analog-to-digital converter. Interface to the MC68HC68A1 is through a standard serial peripheral interface (SPI). Data is shifted in on the shift-data-in (SDI) pin and out on the shift-data-out (SDO) pin synchronous with the second edge of the shift-clock (SCK) pin following chip enable (CE) being activated. This device is compatible with the MC68HC05D2 microcomputer and will interface directly without additional components.

The MC68HC68A1 performs a 10-bit analog-to-digital conversion in a maximum conversion time of 100 microseconds. Data out from the device is transferred in two 8-bit bytes using the serial peripheral interface burst mode operation. The most significant 8-bit byte contains data in valid bits which the controlling microcomputer can monitor to insure correct data.

One of eight analog channels can be accessed through three of the eight address bits. Additional address information can be used to establish the systems operation to allow for system polling or interrupt driven communications from the controlling microcomputer. The address register will allow direct access to any of the sixteen 8-bit on-chip registers containing the eight channels of A/D information.

The device is available in a 16-pin package and contains an on-chip Schmitt oscillator clock input which can be directly driven by a system clock or connected to an external capacitor to develop an independent clock for the A/D device.

PIN ASSIGNMENT



MOTOROLA

MC68HC68R1
MC68HC68R2

Advance Information

HCMOS
(HIGH-DENSITY CMOS SILICON-GATE)

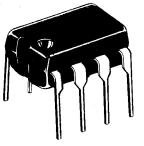
8-BIT SERIAL STATIC RAMs

3

8-BIT SERIAL STATIC RAMs

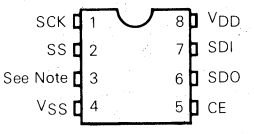
The MC68HC68R1 and MC68HC68R2 are serially organized 128-word (MC68HC68R1) or 256-word (MC68HC68R2) by 8-bit static random access memories (RAMs). These RAMs are intended for use in systems where minimum package and interconnect size, low power, and simplicity of use are desirable; for example, in systems utilizing synchronous serial 3-wire (clock, data in, data out) interfaces. Interface can be made with the MC68HC05D2 without additional components, provided the MC68HC05D2 SPI control register bits CPHA and CPOL are set.

- Fully Static Operation
- Operating Voltage Range: 3 V to 5.5 V
- Maximum Standby Current = 2 μA
- Directly Compatible with SPI Interface
- Separate Data Input and Data Output Pins
- Input Data and Clock Buffers Gated Off with Chip Enable
- Protocol for Fast Sequential Multiple Byte Accesses
- Minimum Data Retention Voltage: 2 V
- Small 8-Lead Plastic Package



P SUFFIX
PLASTIC PACKAGE
CASE 626

PIN ASSIGNMENT



NOTE:
Pin 3 = N/C for MC68HC68R1
Pin 3 = A7 for MC68HC68R2

SIGNAL DESCRIPTION

CHIP ENABLE AND SLAVE SELECT (CE AND SS)

A high level on the CE pin, coincident with a low level on the SS pin, is required for the RAM serial interface logic to become enabled. The device is held in the reset state if either CE is low or SS is high.

SERIAL CLOCK (SCK)

This clock input is used to synchronously latch data in and shift data out of the RAM chip.

SERIAL DATA IN (SDI)

Serial data, present at this port, is latched into the RAM chip by SCK if the chip is enabled and in a write cycle.

SERIAL DATA OUT (SDO)

Serial data is shifted out of this port by SCK if the RAM chip is enabled and in a read cycle.

V_{DD} AND V_{SS}

The V_{DD} pin is the +5 volt power supply and V_{SS} is the ground reference pin.

ADDRESS LINE (A7) — MC68HC68R2 ONLY

This address input is used in the 256-word RAM version to select either of two 128-word memory areas. (Address bits A0-A6, used to provide the address within the 128-word memory area in both the MC68HC68R1 and MC68HC68R2 versions, are the seven least significant bits of the first serial 8-bit byte received at the SDI port at the start of a read or write cycle. The most significant bit of this first byte is the read/write mode bit.)

DATA FORMAT, TRANSFER, AND TIMING

FORMAT

Two type of 8-bit bytes are used when storing or retrieving data in the RAM chip, as shown in Figure 1.

3

FIGURE 1 — SERIAL DATA FORMAT

Address/Control Byte

Bit:	7	6	5	4	3	2	1	0
	R/W	A6	A5	A4	A3	A2	A1	A0

A0-A6: The seven least significant RAM address bits, sufficient to address 128 bytes.

R/W: Read or write data transfer control bit. R/W = 0 initiates one or more memory read cycles; R/W = 1 initiates one or more memory write cycles.

Data Byte

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0

D0-D7: 8 bits of data

TRANSFER

Data transfers, occurring only while CE is high and SS is low, are either single data byte or multiple data transfers. Only only address byte is required for each type of transfer. For multiple transfers, the RAM automatically increments the address as long as it remains enabled. However, anytime enabling signals CE and SS are removed, RAM is reset, and when re-enabled, interprets the first word received as an address word. Therefore, RAM must remain enabled throughout the entire transfer, whether single or multiple, as shown in Figure 2.

TIMING

Address, control, or data bits are latched into RAM by the rising edge of SCK during a write cycle. During a read cycle, the rising edge of SCK shifts out the data bits. Bit switching occurs during the trailing edge of SCK and ensures that the bit value is valid when the SCK rising edge occurs, as shown in Figure 3.

FIGURE 2 — SERIAL TRANSMISSION BYTE SEQUENCES

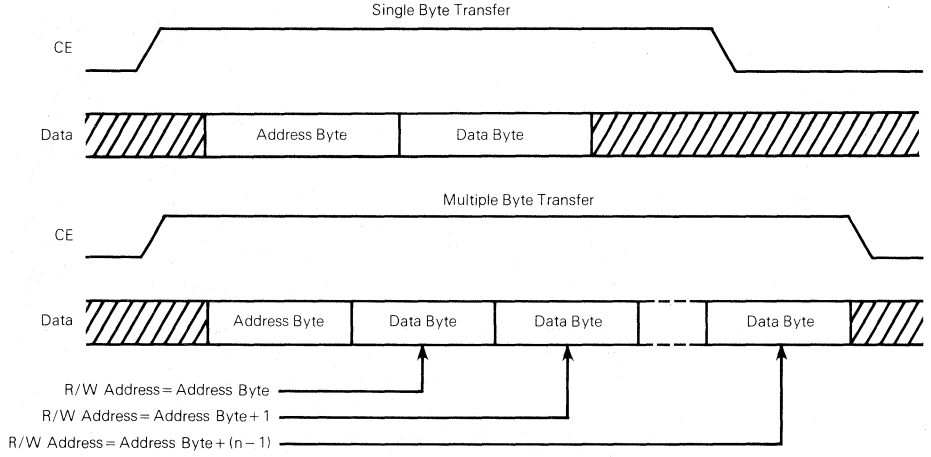
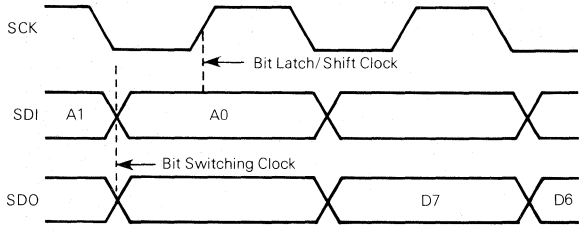


FIGURE 3 — RAM TIMING DIAGRAM



MOTOROLA

MC68HC68T1

Product Preview

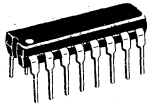
HCMOS
(HIGH-DENSITY CMOS
SILICON-GATE)

REAL-TIME CLOCK
PLUS RAM AND
POWER SENSE/CONTROL

REAL-TIME CLOCK PLUS RAM AND POWER SENSE/CONTROL

The MC68HC68T1 HCMOS clock/RAM peripheral contains a real-time clock/calendar, a 32x8 static RAM, and a synchronous, serial, three-wire interface for communication with a microcomputer. The following summarizes the features of the MC68HC68T1.

- Full Clock Features - Seconds, Minutes, Hours (AM/PM), Day-of-Week, Date, Month, Year (0-99), Auto Leap Year
- 32 Word by 8-Bit RAM
- Minimum Operating Voltage 2.2 Volts
- Burst Mode for Reading/Writing Successive Addresses in Clock or RAM
- Selectable Crystal or 50/60 Hz Line Input
- BCD Data Contained in Registers
- Buffered Clock Output for Driving CPU Clock, Timer, Colon, or LCD Backplane
- Power-On-Reset with First-Time-Up Flag
- Freeze Circuit Eliminates Software Overhead During a Clock Read
- Three Independent Interrupt Modes — Alarm, Periodic, or Power-Down Sense
- CPU Reset Output — Provides Orderly Power Up/Down
- Watch-Dog Circuit
- Auto Switchable Clock



P SUFFIX
PLASTIC PACKAGE
CASE 648

DESCRIPTION

The MC68HC68T1 HCMOS clock/RAM peripheral contains a real-time clock/calendar, a 32 × 8 static RAM, and a synchronous, serial, three-wire interface for communication with a microcomputer. Operating in a burst mode, successive clock or RAM locations can be read or written using only a single starting address. An on-chip oscillator allows acceptance of a selectable crystal frequency or can be programmed to accept a 50/60 hertz line input frequency.

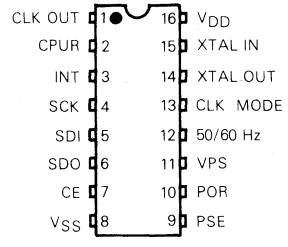
Three pins give the MC68HC68T1 the capability for sensing powerup/powerdown conditions, a capability useful for battery-backup systems. The 16-pin dual-in-line package has an interrupt output capable of signalling the microcomputer of the occurrence of three separately selectable conditions. An alarm can be set for comparison with the seconds, minutes, and hours registers. This alarm can be used in conjunction with the power supply enable output to initiate a system power-up sequence.

A software power-down sequence can be initiated by setting a bit in the interrupt control register. This applies a reset to the CPU, using the CPU RESET output, sets the clock (CLK) and power supply enable (PSE) output pins low, and disables the serial interface. This condition is held until an edge is sensed on the varying power sense (VPS) input, signalling system power coming on, or by activation of a previously enabled interrupt.

A watch-dog circuit can be enabled that requires the microcomputer to toggle the chip enable (CE) pin of the MC68HC68T1 approximately every 8 microseconds, without performing a serial transfer. If this condition is not sensed, the CPU RESET line resets the CPU.

A block diagram of the MC68HC68T1 is shown in Figure 1.

PIN ASSIGNMENT



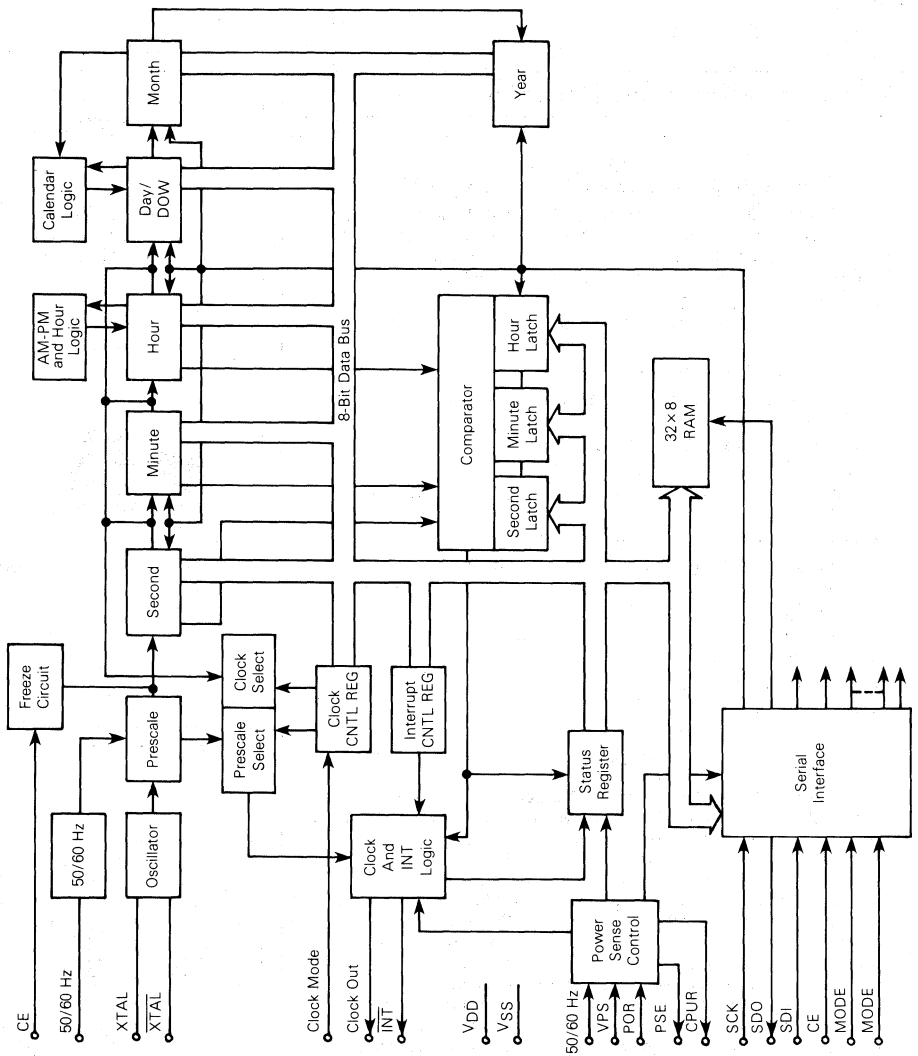


FIGURE 1 - FUNCTIONAL BLOCK DIAGRAM

SIGNAL DESCRIPTION

CLOCK OUTPUT (CKL OUT)

This signal is the buffered clock output which can provide one of fifteen selectable frequencies.

CPU RESET (CPUR)

This output can be used to drive the CPU reset pin to permit orderly powerup/powerdown.

INTERRUPT (INT)

The interrupt output is driven by a single NFET pulldown transistor and can be activated by three selectable conditions.

SERIAL CLOCK (SCK)

The serial clock input is used to latch data into, and shift data out of, the interface logic.

SERIAL DATA IN (SDI)

The serial data input, present at this port, is latched into the interface logic, by SCK, if the logic is enabled.

SERIAL DATA OUT (SDO)

The serial data ouput, present at this port, is shifted out of the interface logic, by SCK, if the logic is enabled.

CHIP ENABLE (CE)

When high, the chip-enable input enables the interface logic. Otherwise the logic is in a reset state. The watch-dog circuit can be toggled at this pin.

POWER SUPPLY ENABLE (PSE)

The power supply enable output is used to control system power and is enabled high on a varying power sense edge, an interrupt, or a power-on reset. PSE is set low by writing a high into the power-down bit of the interrupt control register.

POWER—ON RESET (POR)

This Schmitt trigger generates a power-on reset signal using an external RC network.

VARYING POWER SENSE (VPS)

The varying power sense input is connected to system power and detects a positive edge that indicates a switch from battery-backup power to an external power source. This action disables the CPUR output, enables the PSE output and switches to line from crystal operation, if the auto switchable clock option in being used. If this pin is not used, it should be connected to V_{DD}.

LINE SENSE (50/60 Hz)

The line sense input can be used to drive two functions. If the clock is selected for line operation, a Schmitt trigger input senses the 50/60 Hz input. If the power sense interrupt is enabled, this input is used to sense when external power to the system is turning off. If this pin is not used, it should be connected to V_{DD}.

CLOCK MODE SELECT (CLK MODE)

When tied to V_{DD}, the clock mode select intput selects the clock output for XTAL following a power-on reset. When tied to V_{SS}, the clock output is disabled following a power-reset.

CRYSTAL INPUT/OUTPUT (XTAL IN AND XTAL OUT)

For crystal operation, these two pins are connected to a 32768 hertz, 1.048576 megahertz, 2.097152 megahertz, or 4.194304 megahertz crystal. If crystal operation is not required, connect XTAL IN to V_{DD} or V_{SS} and leave XTAL OUT open. If an external clock is used, connect the external clock to XTAL IN, and leave XTAL OUT open.

V_{DD} AND V_{SS}

Power is supplied to the MC68HC68T1 using these two pins. V_{DD} is the +5 volts power input and V_{SS} is the power supply ground reference pin.

CLOCK/RAM TRANSFER AND WORD FORMAT

The following paragraphs describe the clock/RAM transfer and word format.

TRANSFER FORMAT

Data transfers, occurring only while CE is high are either single data byte or multiple data byte transfers. Only one address byte is required for each type of transfer.

For mulitple transfers, the clock/RAM automatically increments the address as long as it remains enabled. The clock/RAM must remain enabled between transmission of address and data bytes or between successive data bytes in the case of mulitple byte transfers. The serial control logic in the clock/RAM is held reset when the clock/RAM is not enabled or the software powerdown is enabled (refer to Figure 2).

SERIAL DATA FORMAT

The address, control, and status 8-bit byte formats, used in the clock/RAM registers, are shown in Figures 3 through 9.

FIGURE 2 — SERIAL TRANSMISSION BYTE SEQUENCES

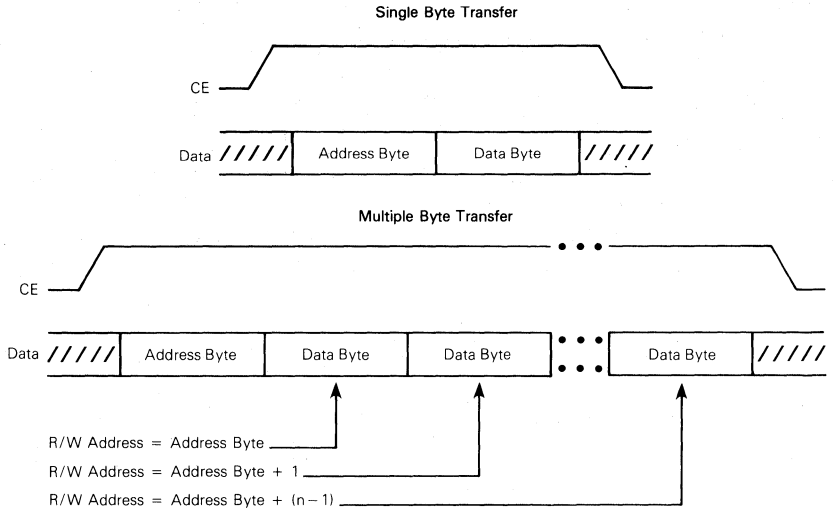


FIGURE 3 — ADDRESS/CONTROL BYTE

7	6	5	4	3	2	1	0
Read Write	0	Clock RAM	A4	A3	A2	A1	A0

Read/Write: Read or write data transfer control bit. R/W = 0 initiates one or more write cycles. R/W = 1 initiates one or more read cycles.

Clock/RAM: Clock or RAM select bit. If bit is set high, the clock is selected. If bit is set low, the RAM is selected.

A0-A4: Selects desired address of RAM or specifies clock register. If invalid address specified, it is ignored.

Test Mode: If the address is set to 01010101, the test mode is entered.

FIGURE 4 — RAM DATA FORMAT

	7	6	5	4	3	2	1	0
$00-1F	D7	D6	D5	D4	D3	D2	D1	0

FIGURE 5 — CLOCK DATA REGISTERS

	7	6	5	4	3	2	1	0
$00	X	Tens of Seconds			Seconds Units			
$01	X	Tens of Minutes			Minutes Units			
$02	12/24	X	AM/PM	10 HR	Hours Units			
$03	X	X	X	X	X	Day of Week		
$04	X	X	10 Date		Date Units			
$05	X	X	X	Tens	Month Units			
$06	Tens of Years				Year Units			

FIGURE 6 — ALARM DATA REGISTERS (WRITE ONLY)

	7	6	5	4	3	2	1	0
$08	X	Alarm Seconds						
$09	X	Alarm Minutes						
$0A	X	X	Alarm Hours					

FIGURE 7 — CLOCK CONTROL FORMAT

	7	6	5	4	3	2	1	0
$10	Start Stop	Line XTAL	XTAL Sel 1	XTAL Sel 0	50 Hz 60 Hz	Clk Out 2	Clk Out 1	Clk Out 0

Start/Stop: This bit high enables the clock counters. A low resets all counter bits divider stage and inhibits clock operation.

Line/XTAL: This bit high, selects clock operation on the 50/60Hz input. If low, it enables crystal operation. If line operation is selected and the power sense interrupt is enabled, operation is automatically switched to crystal operation at power off. If a rising edge is sensed on the VPS pin, line operation is automatically selected. If necessary, the crystal can then be tuned for battery-backup operation.

XTAL Select: These pins select one of four possible crystal frequencies as specified below.

0 - 4.194304 MHz
1 - 2.097152 MHz
2 - 1.048576 MHz
3 - 32768 Hz

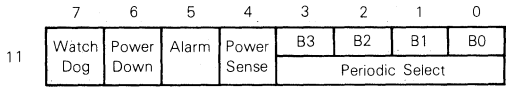
50/60 Hz: This bit high, selects 50 Hz input; a low selects 60 Hz.

Clock Out: These three bits specify one of seven output frequencies:

0 - Disable
1 - 1 Hz
2 - 2 Hz
3 - 50 Hz, 60 Hz, or 64 Hz
4 - XTAL
5 - XTAL %2
6 - XTAL %4
7 - XTAL %8

Read/Reset: All register bits can be read, and are automatically reset by POR except the clock out 2 bit. If clock mode input pin is low, clock out 2 is reset on POR. If clock mode is high, clock out 2 is set on POR, permitting the clock output pin to drive an MPU clock input.

FIGURE 8 — INTERRUPT CONTROL FORMAT



Watch Dog: This bit high enables watch-dog function. This requires CPU to toggle chip enable input without a serial transfer. Otherwise, the CPU RESET signal resets CPU. Maximum time between toggling depends on input clock selected, as listed below:

Selected Clock	50 Hz	60 Hz	XTAL
Maximum Time	10 ms	8.3 ms	7.8 ms

Power Down: This bit high, initiates a power down. This applies a CPU reset, sets the clock and PSE output pins low, and disables the serial interface. Power down is released if a previously enabled interrupt becomes active or the VPS pin senses a rising edge, signalling CPU power up.

Alarm: This bit high enables comparator output to trigger the interrupt circuit.

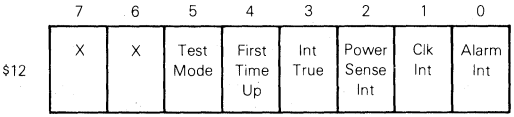
Power Sense: This bit high enables sense circuits to detect main power down via the 50/60 Hz input pin. This activates a threshold detector centered at V_{DD}. Maximum time required to sense power down is 2.5 ms plus associated RC time constant of input circuit. For this function, the crystal need not be enabled, but proper selection/operation is required. The circuit is automatically disabled after the interrupt.

Periodic Select: These four bits select periodic interrupt frequency. Selectable options are listed below:

0 - Disable
1 - 2048 Hz XTAL Only
2 - 1024 Hz XTAL Only
3 - 512 Hz XTAL Only
4 - 256 Hz XTAL Only
5 - 128 Hz XTAL Only
6 - 50 Hz, 60 Hz, or 64 Hz Line or XTAL
7 - 32 Hz XTAL Only
8 - 32 Hz XTAL Only
9 - 8 Hz XTAL Only
10 - 4 Hz XTAL Only
11 - 2 Hz
12 - 1 Hz
13 - Minute
14 - Hour
15 - Day

Read/Reset: All bits in the register can be read to as well as being written to. All bits are also rest by the power-on-reset.

FIGURE 9 — STATUS REGISTER (READ ONLY)



Test Mode: If clock/RAM enters test mode, this bit goes high. Test mode and this bit are reset by a POR or a status register read.

First Time: This bit high signifies that a POR has occurred. This occurs if data in clock/RAM is not correct and should be initialized. This bit is reset by a status register read.

Interrupt True: This bit high signifies a valid power, clock or alarm interrupt. This bit is reset on a register read.

Power Sense: This bit high indicates a power sense interrupt. The MCU can now initiate a power down sequence. This bit is reset on a read.

Clock Interrupt: This bit high indicates a clock-selected, interrupt. This bit is reset on a read.

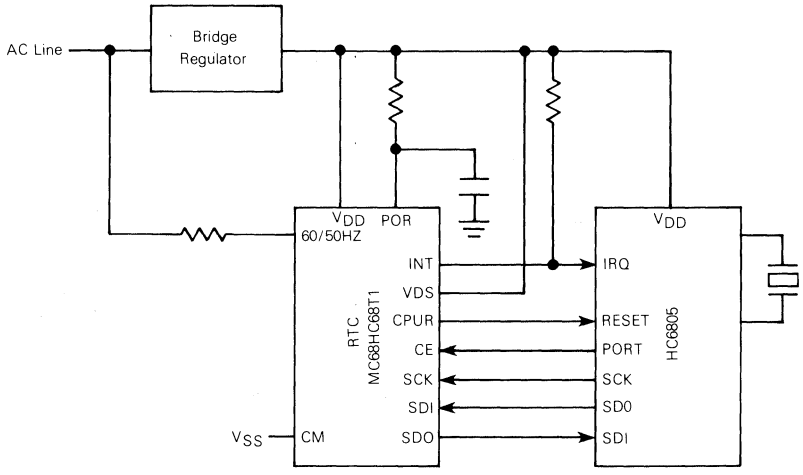
Alarm Interrupt: This bit set high indicates an alarm function interrupt. This bit is reset on a read.

Reset: All register bits are reset by POR except the first time up bit, which is set. The interrupt output is reset on a register read.

SYSTEMS CONFIGURATIONS

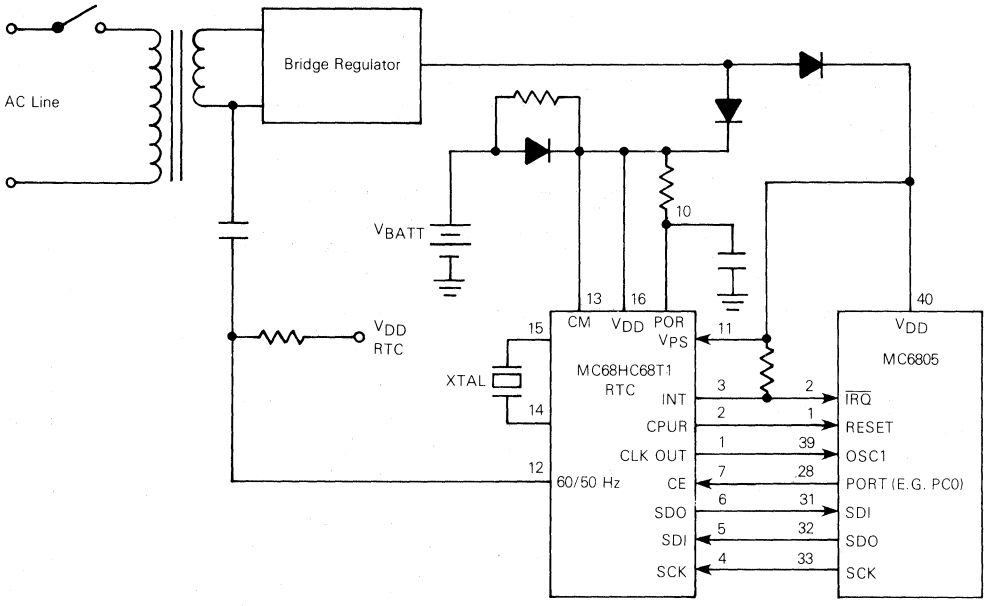
Examples of four system configurations (Figure 10 through 13) are shown in schematic diagrams.

FIGURE 10 — POWER ALWAYS ON SYSTEM



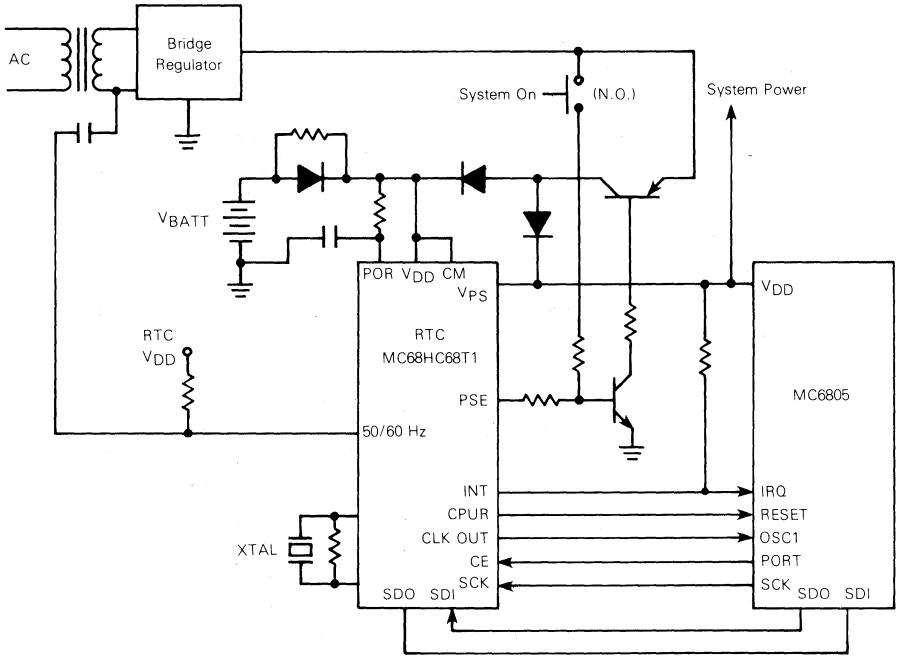
NOTE: Clock circuit driven by line input frequency. Power-on-reset circuit included to detect power failure.

FIGURE 11 — EXTERNALLY CONTROLLED POWER SYSTEM



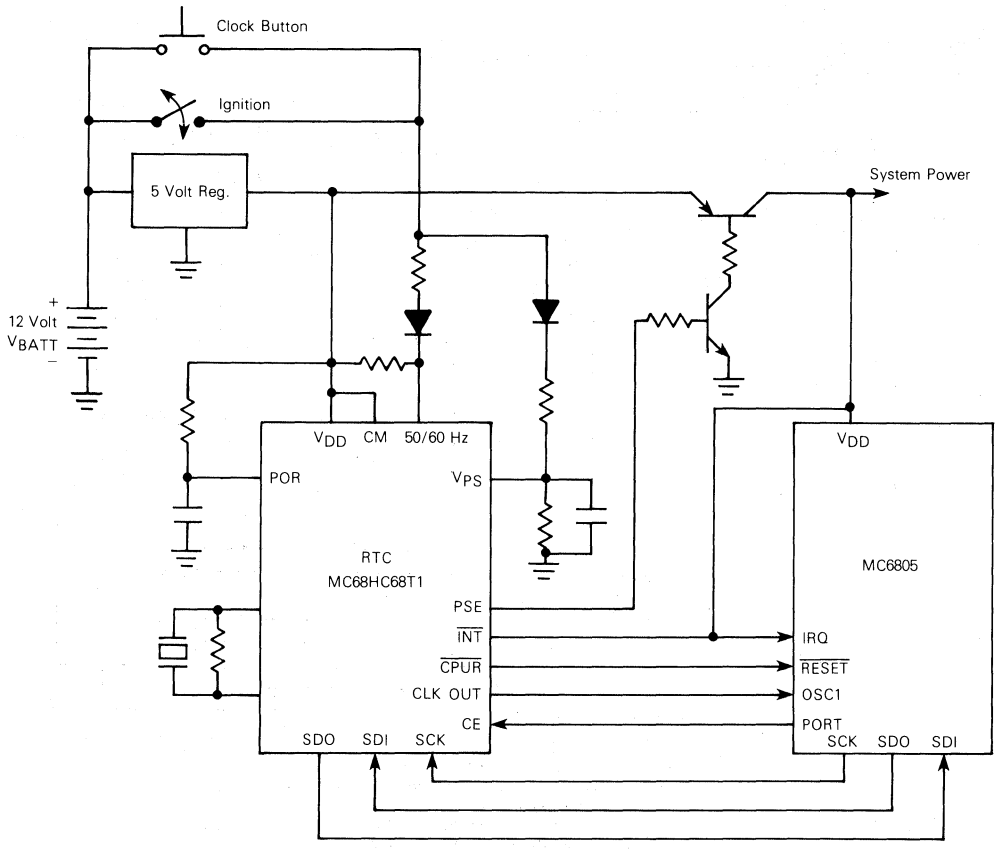
NOTE: The 50/60 Hz input can drive the clock and sense when the switch opens. The switchable clock option can be used. Also one crystal can be used by employing the clock output pin with the clock mode input tied to V_{DD}.

FIGURE 12 — CPU CONTROLLED POWER SYSTEM



NOTE: To power down the system, the CPU can give the clock a power down instruction. Before powering down, the interrupt can be programmed using the alarm option, to permit the interrupt to power the system back up with the PSE output. An external switch can be included to power the system up independent of a programmed power up.

FIGURE 13 — AUTOMOTIVE SYSTEM



NOTE: The VPS and 50/60 Hz inputs can be used to sense the ignition turning on and off. The RC time constant on the VPS pin must be made long enough to be able to perform a power down sequence. An external swtich is included to activate the system without turning on the ignition.

MOTOROLA

MC6875
MC6875A

Specifications and Applications Information

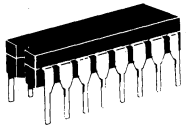
M6800 CLOCK GENERATOR

Intended to supply the non-overlapping $\phi 1$ and $\phi 2$ clock signals required by the microprocessor, this clock generator is compatible with 1.0, 1.5, and 2.0 MHz versions of the MC6800. Both the oscillator and high capacitance driver elements are included along with numerous other logic accessory functions for easy system expansion.

Schottky technology is employed for high speed and PNP-buffered inputs are employed for NMOS compatibility. A single +5 V power supply, and a crystal or RC network for frequency determination are required.

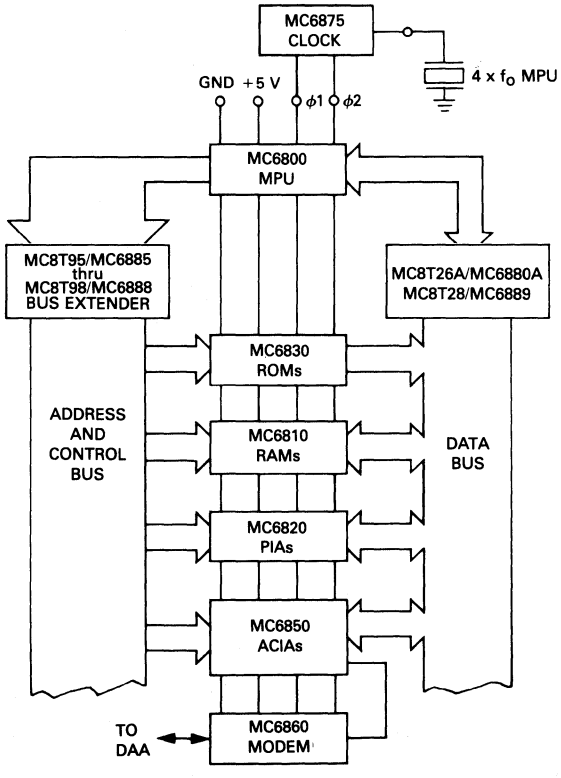
M6800 TWO-PHASE CLOCK GENERATOR/DRIVER

SCHOTTKY MONOLITHIC INTEGRATED CIRCUIT

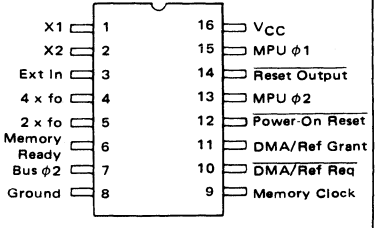


L SUFFIX
CERAMIC PACKAGE
CASE 620-02

Typical MPU System with Bus Extenders



PIN CONNECTIONS



ORDERING INFORMATION		
Device	**Temperature Range**	**Package**
MC6875L	0 to +70°C	Ceramic Dip
MC6875AL	-55 to +125°C	Ceramic Dip

ABSOLUTE MAXIMUM RATINGS (Unless otherwise noted $T_A = 25^oC$.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	+7.0	Vdc
Input Voltage	V_I	+5.5	Vdc
Operating Ambient Temperature Range MC6875L MC6875AL	T_A	 0 to +70 -55 to +125	^oC
Storage Temperature Range	T_{stg}	-65 to +150	^oC
Operating Junction Temperature	T_J	175	^oC

NOTE:
Operation of the MC6875AL over the full military temperature range (to maximum T_A) will result in excessive operating junction temperature.

The use of a clip on 16 pin heat sink similar to AAVID Engineering, Inc., Model 5007 ($R_{\theta CA}$ = 18°C/W) is recommended above $T_A \approx 95°C$.

Contact AAVID Engineering, Inc.
30 Cook Court
Laconia, New Hampshire 03246
Tel. (603) 524-4443

RECOMMENDED OPERATING CONDITIONS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	+4.75 to +5.25	Vdc
Operating Ambient Temperature Range	T_A	0 to +70	^oC

3

OPERATING DYNAMIC POWER SUPPLY CURRENT

Characteristic	Symbol	Min	Typ	Max	Unit
Power Supply Currents (V_{CC} = 5.25 V, f_{osc} = 8.0 MHz, V_{IL} = 0 V, V_{IH} = 3.0 V) Normal Operation (Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs at High Logic State)	I_{CCN}	–	–	150	mA
Memory Ready Stretch Operation (Memory Ready Input at Low Logic State; $\overline{\text{DMA/Refresh Request}}$ Input at High Logic State)	I_{CCMR}	–	–	135	mA
$\overline{\text{DMA/Refresh Request}}$ Stretch Operation (Memory Ready Input at High Logic State; $\overline{\text{DMA/Refresh Request}}$ Input at Low Logic State)	I_{CCDR}	–	–	135	mA

ELECTRICAL CHARACTERISTICS

(Unless otherwise noted specifications apply over recommended power supply and temperature ranges.
Typical values measured at V_{CC} = 5.0 V and T_A = 25°C.)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage — High Logic State					
MPU ϕ1 and ϕ2 Outputs					V
(V_{CC} = 4.75 V, I_{OHM} = -200 μA)	V_{OHM}	V_{CC} - 0.6	–	–	
(V_{CC} = 5.25 V, I_{OHMK} = +5.0 mA)	V_{OHMK}	–	–	V_{CC} + 1.0	
Bus ϕ2 Output					V
(V_{CC} = 4.75 V, I_{OHB} = -10 mA)	V_{OHB}	2.4	–	–	
(V_{CC} = 5.25 V, I_{OHBK} = +5.0 mA)	V_{OHBK}	–	–	V_{CC} + 1.0	
4 x fo Output					V
(V_{CC} = 4.75 V, V_{IH} = 2.0 V, I_{OH4X} = -500 μA)	V_{OH4X}	2.4	–	–	
2 x fo, DMA/Refresh Grant and Memory Clock Outputs	V_{OH}	2.4	–	–	V
(V_{CC} = 4.75 V, I_{OH} = -500 μA)					
$\overline{\text{Reset}}$ Output	$V_{OH\overline{R}}$	2.4	–	–	V
(V_{CC} = 4.75 V, V_{IH} = 3.3 V, $I_{OH\overline{R}}$ = -100 μA)					
Output Voltage — Low Logic State					
MPU ϕ1 and ϕ2 Outputs					V
(V_{CC} = 4.75 V, I_{OLM} = +200 μA)	V_{OLM}	–	–	0.4	
(V_{CC} = 4.75 V, I_{OLMK} = -5.0 mA)	V_{OLMK}	–	–	-1.0	
Bus ϕ2 Output					V
(V_{CC} = 4.75 V, I_{OLB} = +48 mA)	V_{OLB}	–	–	0.5	
(V_{CC} = 4.75 V, I_{OLBK} = -5.0 mA)	V_{OLBK}	–	–	-1.0	
4 x fo Output					V
(V_{CC} = 4.75 V, V_{IL} = 0.8 V, I_{OL4X} = 16 mA)	V_{OL4X}	–	–	0.5	
2 x fo, DMA/Refresh Grant and Memory Clock Outputs	V_{OL}	–	–	0.5	V
(V_{CC} = 4.75 V, I_{OL} = 16 mA)					
$\overline{\text{Reset}}$ Output	$V_{OL\overline{R}}$	–	–	0.5	V
(V_{CC} = 4.75 V, V_{IL} = 0.8 V, $I_{OL\overline{R}}$ = 3.2 mA)					
Input Voltage — High Logic State					V
Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	V_{IH}	2.0	–	–	
Input Voltage — Low Logic State					V
Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	V_{IL}	–	–	0.8	
Input Thresholds — $\overline{\text{Power-On Reset}}$ Input (See Figure 2)					V
Output Low to High	V_{ILH}	–	2.8	3.6	
Output High to Low	V_{IHL}	0.8	1.4	–	
Input Clamp Voltage MC6875L	V_{IK}	–	–	-1.0	V
(V_{CC} = 4.75 V, I_{IC} = -5.0 mA) MC6875AL		–	–	-1.5	
Input Current — High Logic State					
Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	I_{IH}	–	–	25	μA
(V_{CC} = 4.75 V, V_{IH} = 5.0 V)					
$\overline{\text{Power-On Reset}}$	$I_{IH\overline{R}}$	–	–	50	μA
(V_{CC} = 5.0 V, $V_{IH\overline{R}}$ = 5.0 V)					
Input Current — Low Logic State					
Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	I_{IL}	–	–	-250	μA
(V_{CC} = 5.25 V, V_{IL} = 0.5 V)					
$\overline{\text{Power-On Reset}}$ Input[1]	$I_{IL\overline{R}}$	–	–	-250	μA
(V_{CC} = 5.25 V, V_{IL} = 0.5 V)					

SWITCHING CHARACTERISTICS

(These specifications apply whether the Internal Oscillator (see Figure 9) or an External Oscillator is used (see Figure 10). Typical values measured at V_{CC} = 5.0 V, T_A = 25°C, fo = 1.0 MHz (see Figure 8).

Characteristic	Symbol	Min	Typ	Max	Unit
MPU φ1 AND φ2 CHARACTERISTICS					
Output Period (Figure 3)	t_o	500	–	–	ns
Pulse Width (Figure 3)	t_{PWM}				ns
(fo = 1.0 MHz)		400	–	–	
(fo = 1.5 MHz)		230	–	–	
(fo = 2.0 MHz)		180	–	–	
Total Up Time (Figure 3)	t_{UPM}				ns
(fo = 1.0 MHz)		900	–	–	
(fo = 1.5 MHz)		600	–	–	
(fo = 2.0 MHz)		440	–	–	
Delay Time Referenced to Output Complement (Figure 3)					
Output High to Low State (Clock Overlap at 1.0 V)	t_{PLHM}	0	–	–	ns
Delay Times Referenced to 2 x fo (Figure 4 MPU φ2 only)					
Output Low to High Logic State	t_{PLHM2X}	–	–	85	ns
Output High to Low Logic State	t_{PHLM2X}	–	–	70	ns
Transition Times (Figure 3)					
Output Low to High Logic State	t_{TLHM}	–	–	25	ns
Output High to Low Logic State	t_{THLM}	–	–	25	ns
BUS φ2 CHARACTERISTICS					
Pulse Width – Low Logic State (Figure 4)	t_{PWLB}				ns
(fo = 1.0 MHz)		430	–	–	
(fo = 1.5 MHz)		280	–	–	
(fo = 2.0 MHz)		210	–	–	
Pulse Width – High Logic State	t_{PWHB}				ns
(fo = 1.0 MHz)		450	–	–	
(fo = 1.5 MHz)		295	–	–	
(fo = 2.0 MHz)		235	–	–	
Delay Times – (Referenced to MPU φ1) (Figure 4)					
Output Low to High Logic State	t_{PLHBM1}				ns
(fo = 1.0 MHz)		480	–	–	
(fo = 1.5 MHz)		320	–	–	
(fo = 2.0 MHz)		240	–	–	
Output High to Low Logic State	t_{PHLBM1}				
(C_L = 300 pF)		–	–	25	
(C_L = 100 pF)		–	–	20	
Delay Times (Referenced to MPU φ2) (Figure 4)					
Output Low to High Logic State	t_{PLHBM2}	-30	–	+25	ns
Output High to Low Logic State	t_{PHLBM2}	0	–	+40	ns
Transition Times (Figure 4)					
Output Low to High Logic State	t_{TLHB}	–	–	20	ns
Output High to Low Logic State	t_{THLB}	–	–	20	ns

SWITCHING CHARACTERISTICS (continued)

Characteristic	Symbol	Min	Typ	Max	Unit
MEMORY CLOCK CHARACTERISTICS					
Delay Times (Referenced to MPU φ2) (Figure 4)					
Output Low to High Logic State	t_{PLHCM}	-50	–	+25	ns
Output High to Low Logic State	t_{PHLCM}	0	–	+40	ns
Delay Times (Referenced to 2 x fo) (Figure 4)					
Output Low to High Logic State	t_{PLHC2X}	–	–	65	ns
Output High to Low Logic State	t_{PHLC2X}	–	–	85	ns
Transition Times (Figure 4)					
Output Low to High State	t_{TLHC}	–	–	25	ns
Output High to Low State	t_{THLC}	–	–	25	ns

2 x fo CHARACTERISTICS

Delay Times (Referenced to 4 x fo) (Figure 4)					
Output Low to High Logic State	t_{PLH2X}	–	–	50	ns
Output High to Low Logic State	t_{PHL2X}	–	–	65	ns
Delay Time (Referenced to MPU ϕ1) (Figure 4)					
Output High to Low Logic State	$t_{PHL2XM1}$				ns
(fo = 1.0 MHz)		365	–	–	
(fo = 1.5 MHz)		220	–	–	
Transition Times (Figure 4)					
Output Low to High Logic State	t_{TLH2X}	–	–	25	ns
Output High to Low Logic State	t_{THL2X}	–	–	25	ns

4 x fo CHARACTERISTICS

Delay Times (Referenced to Ext. In) (Figure 4)					
Output Low to High Logic State	t_{PLH4X}	–	–	50	ns
Output High to Low Logic State	t_{PHL4X}	–	–	30	ns
Transition Time (Figure 4)					
Output Low to High Logic State	t_{TLH4X}	–	–	25	ns
Output High to Low Logic State	t_{THL4X}	–	–	25	ns

MEMORY READY CHARACTERISTICS

Set-Up Times (Figure 5)					
Low Input Logic State	t_{SMRL}	55	–	–	ns
High Input Logic State	t_{SMRH}	75	–	–	ns
Hold Time (Figure 5)					
Low Input Logic State	t_{HMRL}	10	–	–	ns

$\overline{\text{DMA/REFRESH REQUEST}}$ CHARACTERISTICS

Set-Up Times (Figure 6)					
Low Input Logic State	t_{SDRL}	65	–	–	ns
High Input Logic State	t_{SDRH}	75	–	–	ns
Hold Time (Figure 6)					
Low Input Logic State	t_{HDRL}	10	–	–	ns

DMA/REFRESH GRANT CHARACTERISTICS

Delay Time Referenced to Memory Clock (Figure 6)					
Output Low to High Logic State	t_{PLHG}	-15	–	+25	ns
Output High to Low Logic State	t_{PHLG}	-25	–	+15	ns
Transition Times (Figure 6)					
Output Low to High Logic State	t_{TLHG}	–	–	25	ns
Output High to Low Logic State	t_{THLG}	–	–	25	ns

$\overline{\text{RESET}}$ CHARACTERISTICS

Delay Time Referenced to Power-On Reset (Figure 7)					
Output Low to High Logic State	$t_{PLH\overline{R}}$	–	–	1000	ns
Output High to Low Logic State	$t_{PHL\overline{R}}$	–	–	250	ns
Transition Times (Figure 7)					
Output Low to High Logic State	$t_{TLH\overline{R}}$	–	–	100	ns
Output High to Low Logic State	$t_{THL\overline{R}}$	–	–	50	ns

DESCRIPTION OF PIN FUNCTIONS

- 4 x fo – A free running oscillator at four times the MPU clock rate useful for a system sync signal.
- 2 x fo – A free running oscillator at two times the MPU clock rate.
- $\overline{\text{DMA/REF REQ}}$ – An asynchronous input used to freeze the MPU clocks in the ϕ1 high, ϕ2 low state for dynamic memory refresh or cycle steal DMA (Direct Memory Access).
- REF GRANT – A synchronous output used to synchronize the refresh or DMA operation to the MPU.
- MEMORY READY – An asynchronous input used to freeze the MPU clocks in the ϕ1 low, ϕ2 high state for slow memory interface.
- MPU ϕ1
MPU ϕ2 – Capable of driving the ϕ1 and ϕ2 inputs on two MC6800s.
- BUS ϕ2 – An output nominally in phase with MPU ϕ2 having MC8T26A type drive capability.
- MEMORY CLOCK – An output nominally in phase with MPU ϕ2 which free runs during a refresh request cycle.
- $\overline{\text{POWER-ON RESET}}$ – A Schmitt trigger input which controls Reset. A capacitor to ground is required to set the desired time constant. Internal 50 k resistor to V_{CC}. See General Design Suggestions for Manual Reset Operation.
- $\overline{\text{RESET}}$ – An output to the MPU and I/O devices.
- X1, X2 – Provision to attach a series resonant crystal or RC network.
- EXT IN – Allows driving by an external TTL signal to synchronize the MPU to an external system.

FIGURE 1 – BLOCK DIAGRAM

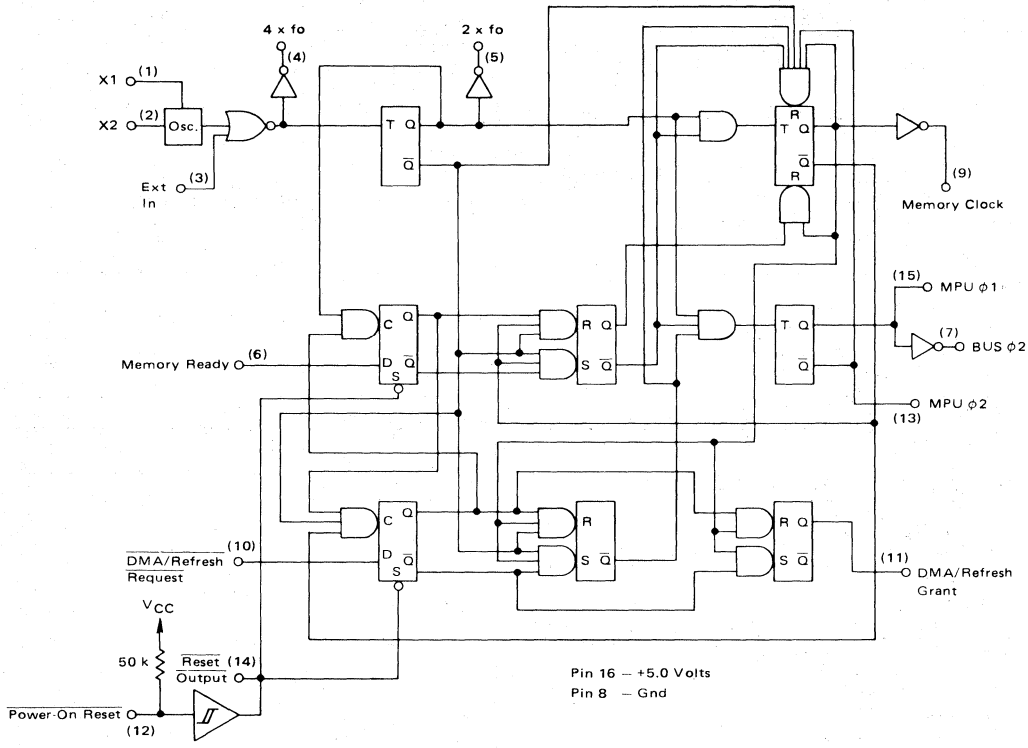


FIGURE 2 – TYPICAL HYSTERESIS CHARACTERISTIC OF $\overline{\text{RESET}}$ FUNCTION

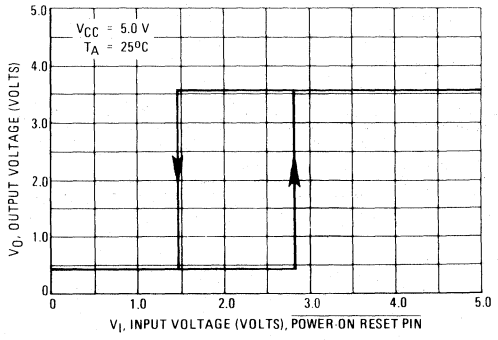
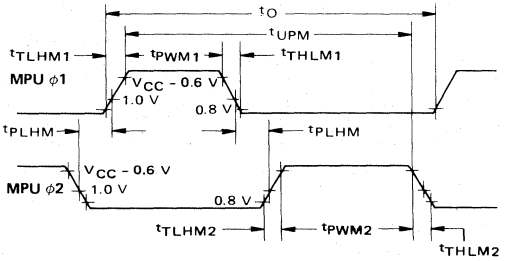


FIGURE 3 – TIMING DIAGRAM FOR MPU ϕ1 AND ϕ2



V_{OV} = 1.0 V = Clock Overlap measurement point

FIGURE 4 – TIMING DIAGRAM FOR NON-STRETCHED OPERATION
(Memory Ready and $\overline{\text{DMA/Refresh Request}}$ held high continuously)
Ext. In Input Voltage: 0 V to 3.0 V, f = 8.0 MHz, Duty Cycle = 50%, $t_{TLHEX} = t_{THLEX} = 5.0$ ns

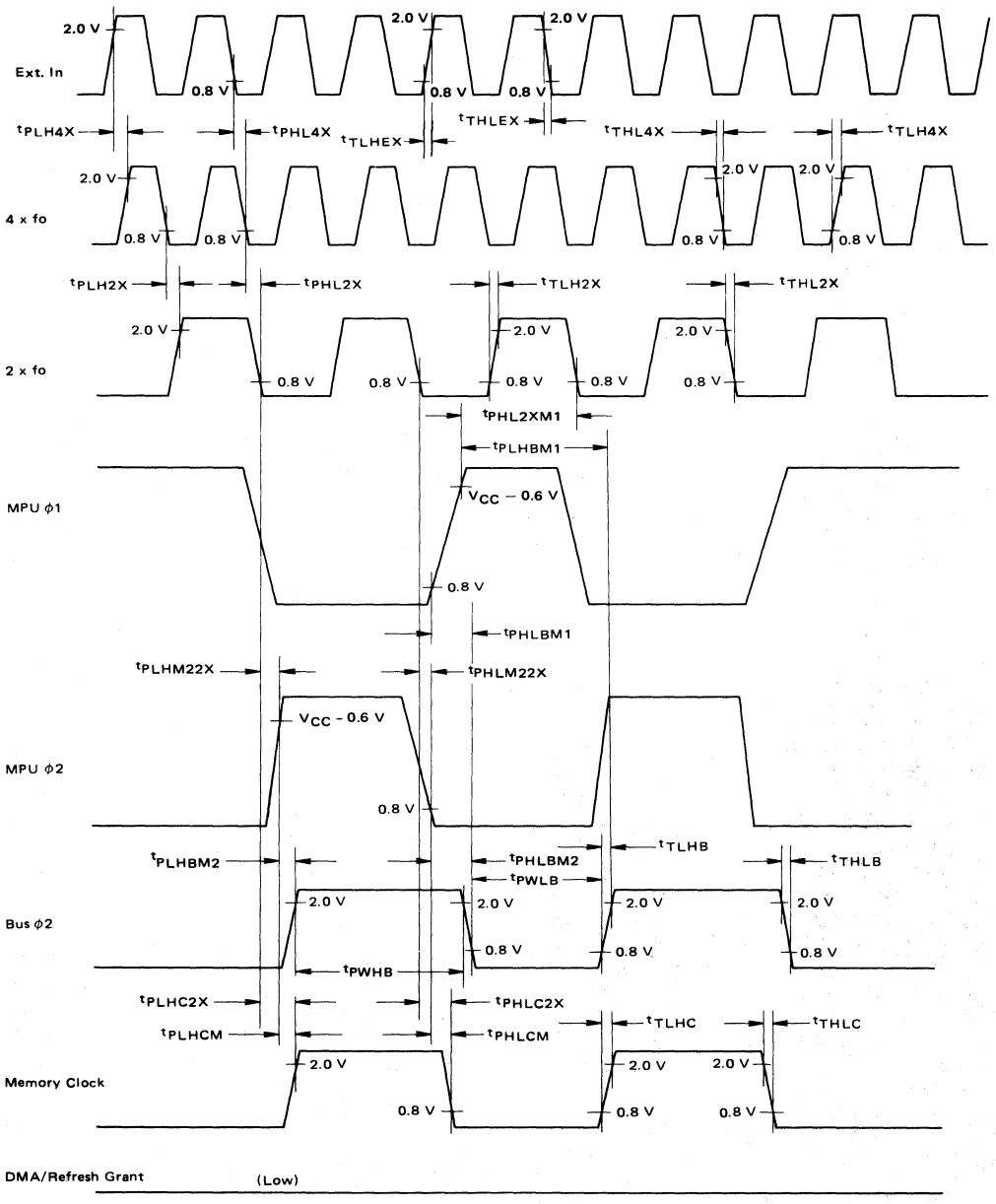


FIGURE 5 – TIMING DIAGRAM FOR MEMORY READY STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLMR} = t_{TLHMR} = 5.0$ ns

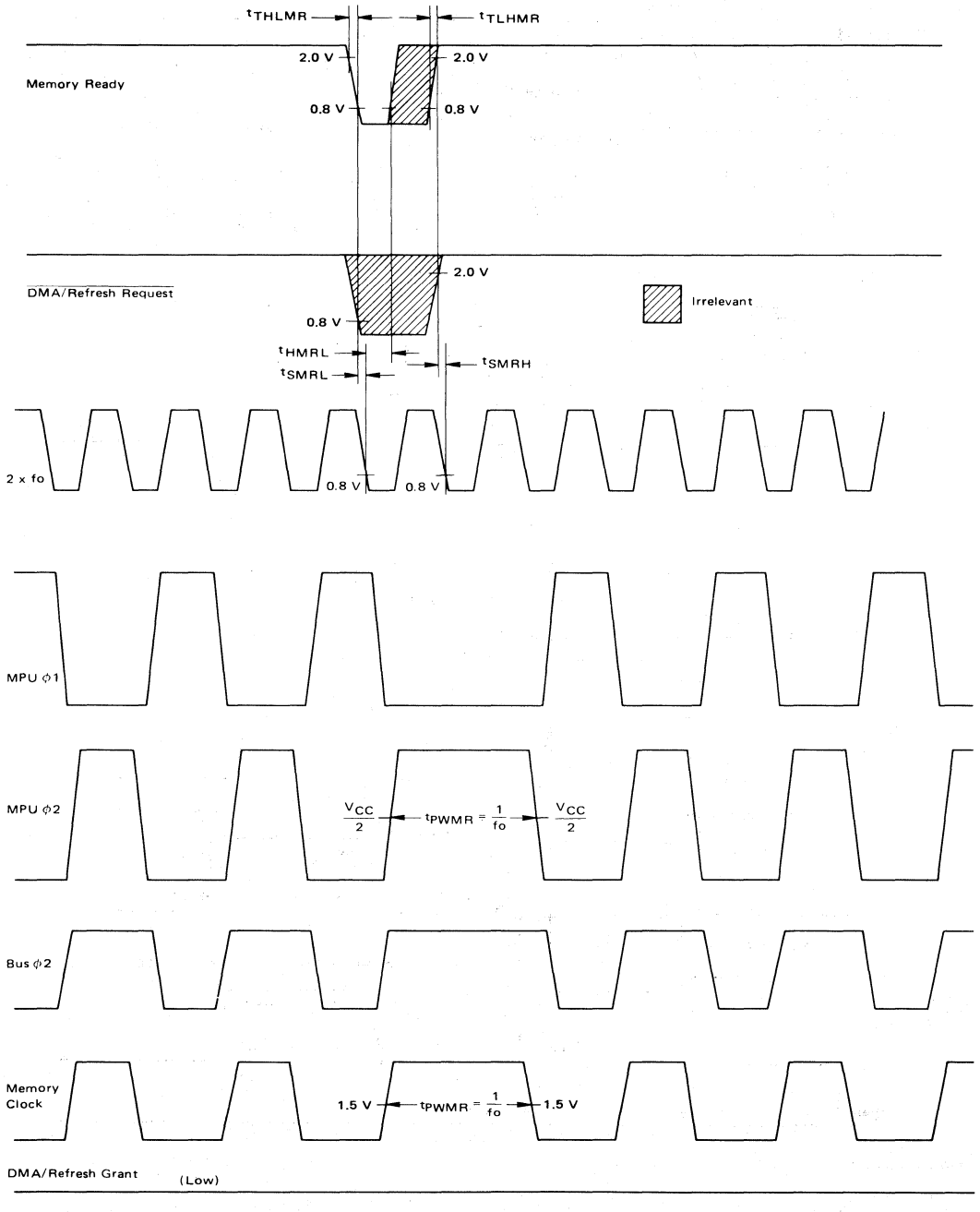


FIGURE 6 – TIMING DIAGRAM FOR $\overline{\text{DMA/REFRESH REQUEST}}$ STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLDR} = t_{TLHDR} = 5.0$ ns

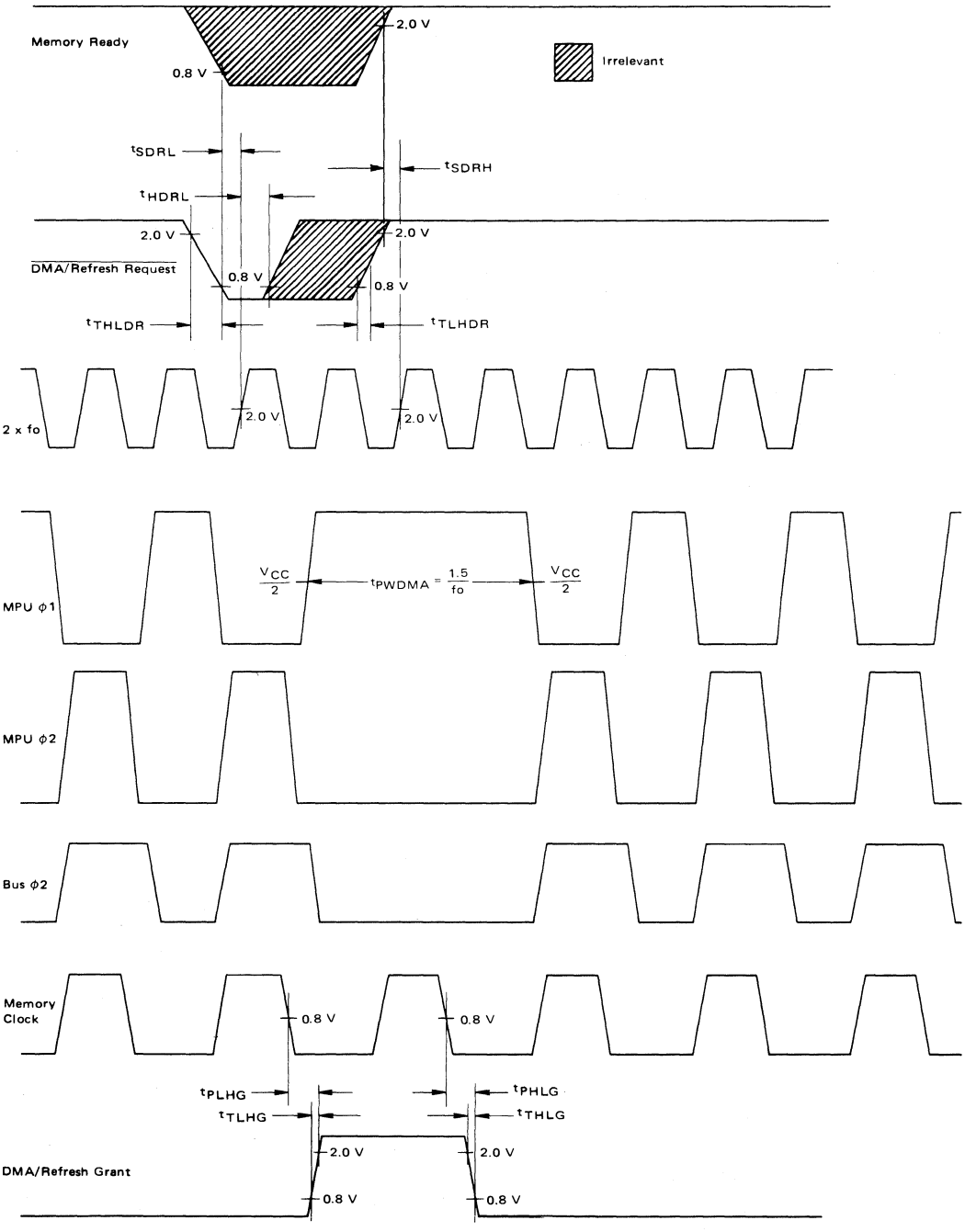


FIGURE 7 – $\overline{\text{POWER ON RESET}}$

Input Voltage: 0 to 5.0 V, f = 100 kHz – Pulse Width = 1.0 μs, $t_{TLH} = t_{THL}$ = 25 ns

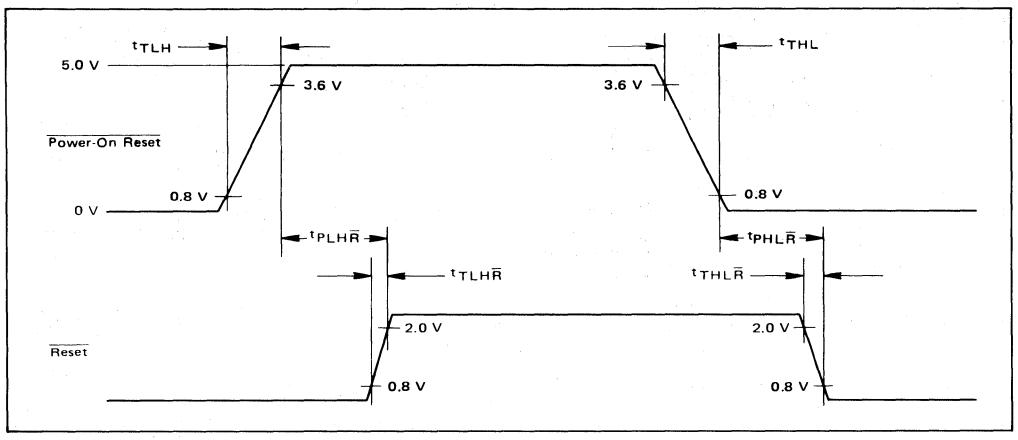
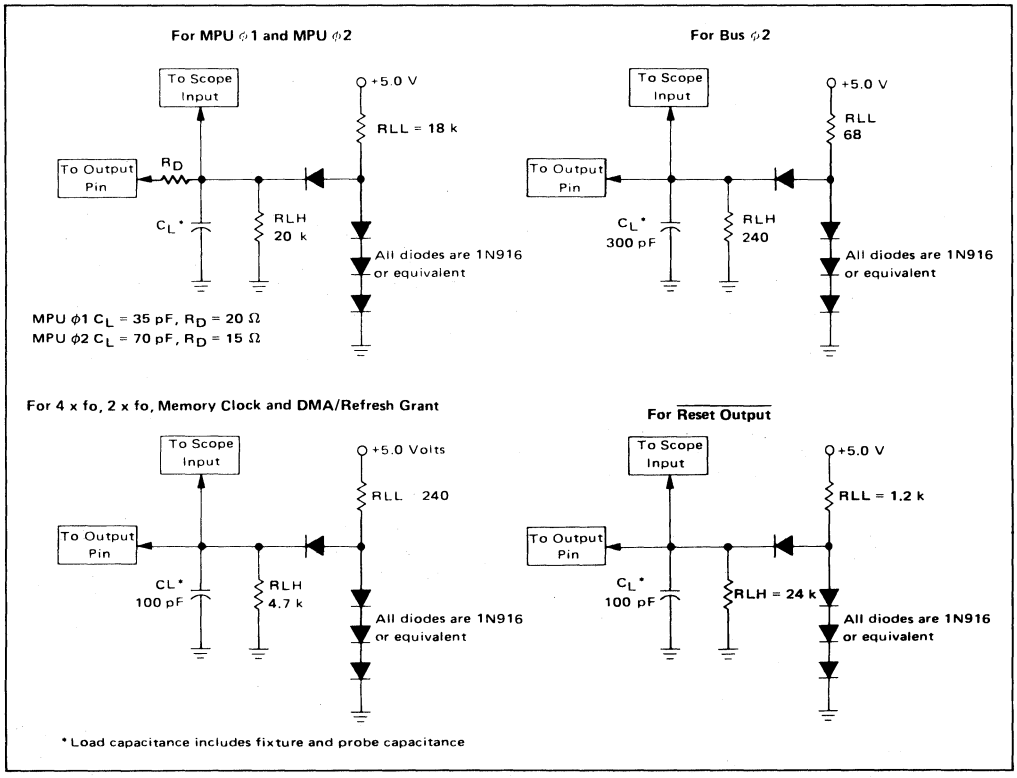


FIGURE 8 – LOAD CIRCUITS



NOTE:
Operation of the MC6875AL over the full military temperature range (to maximum T_A) will result in excessive operating junction temperature.

The use of a clip on 16 pin heat sink similar to AAVID Engineering, Inc., Model 5007 ($R_{\theta CA}$ = 18°C/W) is recommended above $T_A \approx 95°C$.

Contact AAVID Engineering, Inc.
30 Cook Court
Laconia, New Hampshire 03246
Tel. (603) 524-4443

APPLICATIONS INFORMATION

FIGURE 9 – TYPICAL RC FREQUENCY versus VOLTAGE

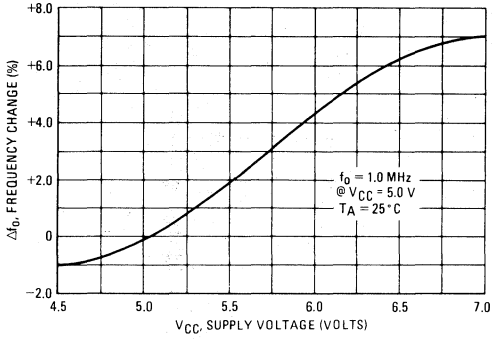


FIGURE 10 – TYPICAL RC FREQUENCY versus TEMPERATURE

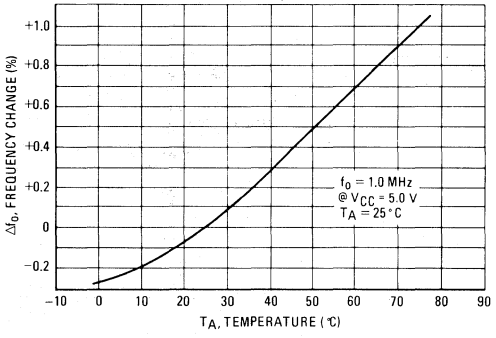
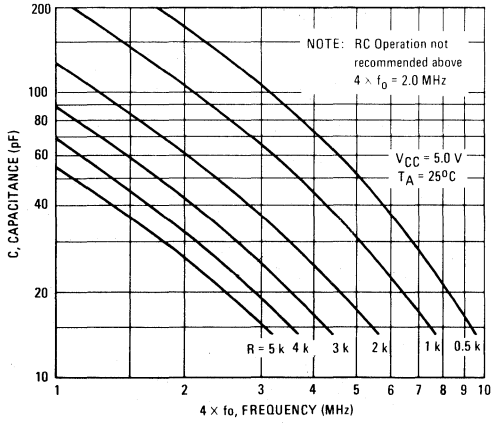


FIGURE 11 – TYPICAL FREQUENCY versus RESISTANCE FOR C VARIABLE



GENERAL

The MC6875 Clock Generator/Driver should be located on the same board and within two inches of the MC6800 MPU. Series damping resistors of 10-30 ohms may be utilized between the MC6875 and the MC6800 on the $\phi 1$ and $\phi 2$ clocks to suppress overshoot and reflections.

The V_{CC} pin (pin 16) of the MC6875 should be bypassed to the ground pin (pin 8) at the package with a 0.1 μF capacitor. Because of the high peak currents associated with driving highly capacitive loads, an adequately large ground strip to pin 8 should be used on the MC6875. Grounds should be carefully routed to minimize coupling of noise to the sensitive oscillator inputs. Unnecessary grounds or ground planes should be avoided near pin 2 or the frequency determining components. These components should be located as near as possible to the respective pins of the MC6875. Stray capacitance near pin 2 or the crystal, can affect the frequency. The can of the crystal should not be grounded. The ground side of the crystal or the C of the R-C oscillator should be connected as directly as possible to pin 8.

Unused inputs should be connected to V_{CC} or ground. Memory Ready, DMA/Refresh Request and Power-On Reset should be connected to V_{CC} when not used. The External Input should be connected to ground when not used.

OSCILLATOR

A tank circuit tuned to the desired crystal frequency connected between terminals X_1 and X_2 as shown in Figure 12, is recommended to prevent the oscillator from starting at other than the desired frequency. The 1kΩ resistor reduces the Q sufficiently to maintain stable crystal control. Crystal manufacturers may recommend a capacitance (C_L) to be used in series with the crystal for optimum performance at series resonance.

See Figures 9 and 10 for typical oscillator temperature and V_{CC} supply dependence for R-C operation.

FIGURE 12 – OSCILLATOR–CRYSTAL OPERATION

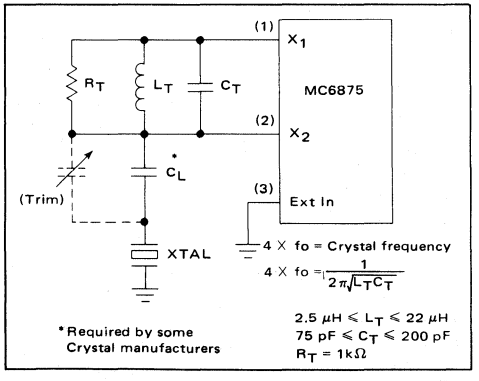
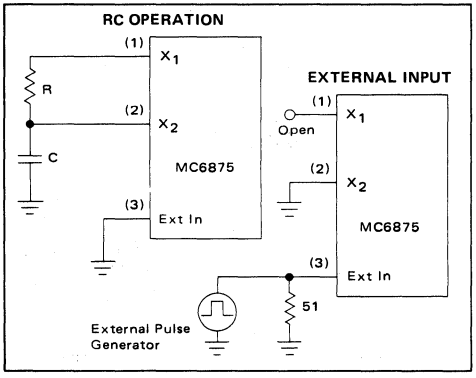


TABLE 1 – OSCILLATOR COMPONENTS

TANK CIRCUIT PARAMETERS		APPROXIMATE CRYSTAL PARAMETERS				CTS KNIGHTS 400 REIMANN AVE. SANDWICH, IL 60548 (815) 786-8411	McCOY ELECT. CO. WATTS & CHESTNUTS STS. MT. HOLLY SPRING, PA 17065 (717) 486-3411	TYCO CRYSTAL PRODUCTS 3940 W. MONTECITO PHOENIX, AZ 85019 (602) 272-7945
L_T μH	C_T pF	R_S Ohms	Co pF	C_1 mpF	fo MHz			
10	150	15-75	3-6	12	4.0	MP-04A * 390 pF	113-31	150-3260
4.7	82	8-45	4-7	23	8.0	MP-080 * 47 pF	113-32	150-3270

Inductors may be obtained from: Coilcraft, Cary, IL 60013 (312) 639-2361

FIGURE 13



To precisely time a crystal to desired frequency, a variable trimmer capacitor in the range of 7 to 40 pF would typically be used. Note it is not a recommended practice to tune the crystal with a parallel load capacitance.

The table above shows typical values for C_T and L_T, typical crystal characteristics, and manufacturers' part numbers for 4.0 and 8.0 megahertz operation.

The MC6875 will function as an R-C oscillator when connected as shown in Figure 13. The desired output frequency (Mϕ1) is approximately:

Formula

$$4 \times fo \approx \frac{320}{C\,(R + .27) + 23}$$

C in picofarads
R in K ohms
4 x fo in Megahertz

(See Figure 11)

It would be desirable to select a capacitor greater than 15 pF to minimize the effects of stray capacitance. It is also desirable to keep the resistor in the 1 to 5 k Ω range. There is a nominal 270 Ω resistor internally at X_1 which is in series with the external R. By keeping the external R as large as possible, the effects due to process variations of the internal resistor on the frequency will be reduced. There will, however, still be some variation in frequency in a production lot both from the resistance variations, external and internal, and process variations of the input switching thresholds. Therefore, in a production system, it is recommended a potentiometer be placed in series with a fixed R between X_1 and X_2.

POWER-ON RESET

As the power to the MC6875 comes up, the $\overline{\text{Reset Output}}$ will be in a high impedance state and will not give a solid V_{OL} output level until V_{CC} has reached 3.5 to 4.0 V. During this time transients may appear on the clock outputs as the oscillator begins to start. This happens at approximately V_{CC} = 3 V. At some V_{CC} level above that, where $\overline{\text{Reset Output}}$ goes low, all the clock outputs will begin functioning normally. This phenomenon of the start-up sequence should not cause any problems except possibly in systems with battery back-up memory. The transients on the clock lines during the time the $\overline{\text{Reset Output}}$ is high impedance could initiate the system in some unknown mode and possibly write into the backup memory system. Therefore in battery backup systems, more elaborate reset circuitry will be required.

Please note that the $\overline{\text{Power-On Reset}}$ input pin of the MC6875 is not suitable for use with a manual MPU reset switch if the $\overline{\text{DMA/Ref Req}}$ or Memory Ready inputs are going to be used. The power on reset circuitry is used to initialize the internal control logic and whenever the input is switched low, the MC6875 is irresponsive to the $\overline{\text{DMA/Ref Req}}$ or Memory Ready inputs. This may result in the loss of dynamic memory and/or possibly a byte of slow static memory. The circuit of Figure 14 is recommended for applications which do not utilize the $\overline{\text{DMA/Ref Req}}$ or Memory Ready inputs. The circuit of Figure 15 is recommended for those applications that do.

FIGURE 14 – MANUAL RESET FOR APPLICATIONS NOT USING $\overline{\text{DMA/REFRESH REQUEST}}$ OR MEMORY READY INPUTS

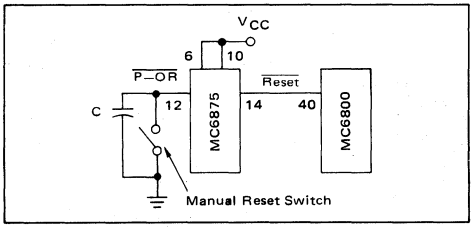
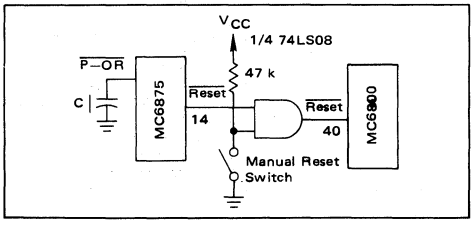


FIGURE 15 – MANUAL RESET FOR SYSTEMS USING DYNAMIC RAM OR SLOW STATIC RAM IN CONJUNCTION WITH MEMORY READY OR $\overline{\text{DMA/REFRESH REQUEST}}$ INPUTS



MOTOROLA

MC8T26A
MC6880A

QUAD THREE-STATE BUS TRANSCEIVER

This quad three-state bus transceiver features both excellent MOS or MPU compatibility, due to its high impedance PNP transistor input, and high-speed operation made possible by the use of Schottky diode clamping. Both the −48 mA driver and −20 mA receiver outputs are short-circuit protected and employ three-state enabling inputs.

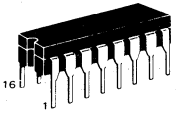
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier diode clamps to suppress excessive undershoot voltages.

The MC8T26A is identical to the NE8T26A and it operates from a single +5 V supply.

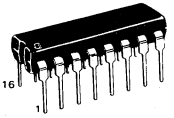
- High Impedance Inputs
- Single Power Supply
- High Speed Schottky Technology
- Three-State Drivers and Receivers
- Compatible with M6800 Family Microprocessor

QUAD THREE-STATE BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

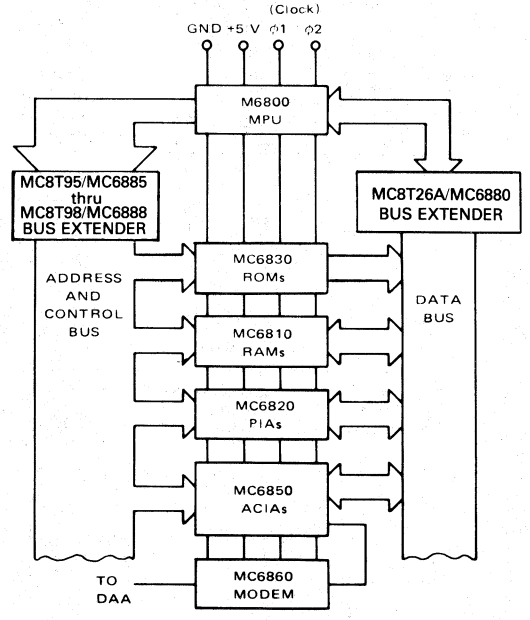


L SUFFIX
CERAMIC PACKAGE
CASE 620-02

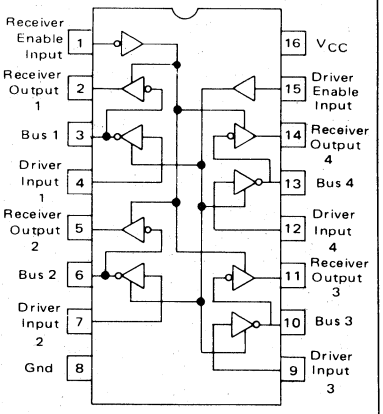


P SUFFIX
PLASTIC PACKAGE
CASE 648-05

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS – MC8T26A MC6880A



ORDERING INFORMATION

Device	Alternate	Temperature Range	Package
MC6880AL	MC8T26AL	0 to +75°C	Ceramic DIP
MC6880AP	MC8T26AP	0 to +75°C	Plastic DIP

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Junction Temperature Ceramic Package Plastic Package	T_J	 175 150	°C
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (4.75 V ≤ V_{CC} ≤ 5.25 V and 0°C ≤ T_A ≤ 75°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current – Low Logic State					
(Receiver Enable Input, $V_{IL(RE)}$ = 0.4 V)	$I_{IL(\overline{RE})}$	–	–	-200	μA
(Driver Enable Input, $V_{IL(DE)}$ = 0.4 V)	$I_{IL(DE)}$	–	–	-200	
(Driver Input, $V_{IL(D)}$ = 0.4 V)	$I_{IL(D)}$	–	–	-200	
(Bus (Receiver) Input, $V_{IL(B)}$ = 0.4 V)	$I_{IL(B)}$	–	–	-200	
Input Disabled Current – Low Logic State (Driver Input, $V_{IL(D)}$ = 0.4 V)	$I_{IL(D)\ DIS}$	–	–	-25	μA
Input Current-High Logic State					
(Receiver Enable Input, $V_{IH(RE)}$ = 5.25 V)	$I_{IH(\overline{RE})}$	–	–	25	μA
(Driver Enable Input, $V_{IH(DE)}$ = 5.25 V)	$I_{IH(DE)}$	–	–	25	
(Driver Input, $V_{IH(D)}$ = 5.25 V)	$I_{IH(D)}$	–	–	25	
(Receiver Input, $V_{IH(B)}$ = 5.25 V)	$I_{IH(B)}$	–	–	100	
Input Voltage – Low Logic State					
(Receiver Enable Input)	$V_{IL(\overline{RE})}$	–	–	0.85	V
(Driver Enable Input	$V_{IL(DE)}$	–	–	0.85	
(Driver Input)	$V_{IL(D)}$	–	–	0.85	
(Receiver Input)	$V_{IL(B)}$	–	–	0.85	
Input Voltage – High Logic State					
(Receiver Enable Input)	$V_{IH(\overline{RE})}$	2.0	–	–	V
(Driver Enable Input)	$V_{IH(DE)}$	2.0	–	–	
(Driver Input)	$V_{IH(D)}$	2.0	–	–	
(Receiver Input)	$V_{IH(B)}$	2.0	–	–	
Output Voltage – Low Logic State					
(Bus Driver) Output, $I_{OL(B)}$ = 48 mA)	$V_{OL(B)}$	–	–	0.5	V
(Receiver Output, $I_{OL(R)}$ = 20 mA)	$V_{OL(R)}$	–	–	0.5	
Output Voltage – High Logic State					
(Bus (Driver) Output, $I_{OH(B)}$ = −10 mA)	$V_{OH(B)}$	2.4	3.1	–	V
(Receiver Output, $I_{OH(R)}$ = −2.0 mA)	$V_{OH(R)}$	2.4	3.1	–	
(Receiver Output, $I_{OH(R)}$ = −100 μA, V_{CC} = 5.0 V)		3.5	–	–	
Output Disabled Leakage Current – High Logic State					
(Bus Driver) Output, $V_{OH(B)}$ = 2.4 V)	$I_{OHL(B)}$	–	–	100	μA
(Receiver Output, $V_{OH(R)}$ = 2.4 V)	$I_{OHL(R)}$	–	–	100	
Output Disabled Leakage Current – Low Logic State					
(Bus Output, $V_{OL(B)}$ = 0.5 V)	$I_{OLL(B)}$	–	–	−100	μA
(Receiver Output, $V_{OL(R)}$ = 0.5 V)	$I_{OLL(R)}$	–	–	−100	
Input Clamp Voltage					
(Driver Enable Input $I_{ID(DE)}$ = −12 mA)	$V_{IC(DE)}$	–	–	-1.0	V
(Receiver Enable Input $I_{IC(RE)}$ = +12 mA)	$V_{IC(RE)}$	–	–	-1.0	
(Driver Input $I_{IC(D)}$ = −12 mA)	$V_{IC(D)}$	–	–	-1.0	
Output Short-Circuit Current, V_{CC} = 5.25 V (1)					
(Bus (Driver) Output)	$I_{OS(B)}$	-50	–	-150	mA
(Receiver Output)	$I_{OS(R)}$	-30	–	-75	
Power Supply Current (V_{CC} = 5.25 V)	I_{CC}	–	–	87	mA

(1) Only one output may be short-circuited at a time.

SWITCHING CHARACTERISTICS (Unless otherwise noted, specifications apply at $T_A = 25^oC$ and $V_{CC} = 5.0$ V)

Characteristic	Symbol	Figure	Min	Max	Unit
Propagation Delay Time from Receiver (Bus) Input to High Logic State Receiver Output	$t_{PLH(R)}$	1	–	14	ns
Propagation Delay Time from Receiver (Bus) Input to Low Logic State Receiver Output	$t_{PHL(R)}$	1	–	14	ns
Propagation Delay Time from Driver Input to High Logic State Driver (Bus) Output	$t_{PLH(D)}$	2	–	14	ns
Propagation Delay Time from Driver Input to Low Logic State Driver (Bus) Output	$t_{PHL(D)}$	2	–	14	ns
Propagation Delay Time from $\overline{\text{Receiver Enable}}$ Input to High Impedance (Open) Logic State Receiver Output	$t_{PLZ(RE)}$	3	–	15	ns
Propagation Delay Time from $\overline{\text{Receiver Enable}}$ Input to Low Logic Level Receiver Output	$t_{PZL(RE)}$	3	–	20	ns
Propagation Delay Time from Driver Enable Input to High Impedance Logic State Driver (Bus) Output	$t_{PLZ(DE)}$	4	–	20	ns
Propagation Delay Time from Driver Enable Input to Low Logic State Driver (Bus) Output	$t_{PZL(DE)}$	4	–	25	ns

3

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

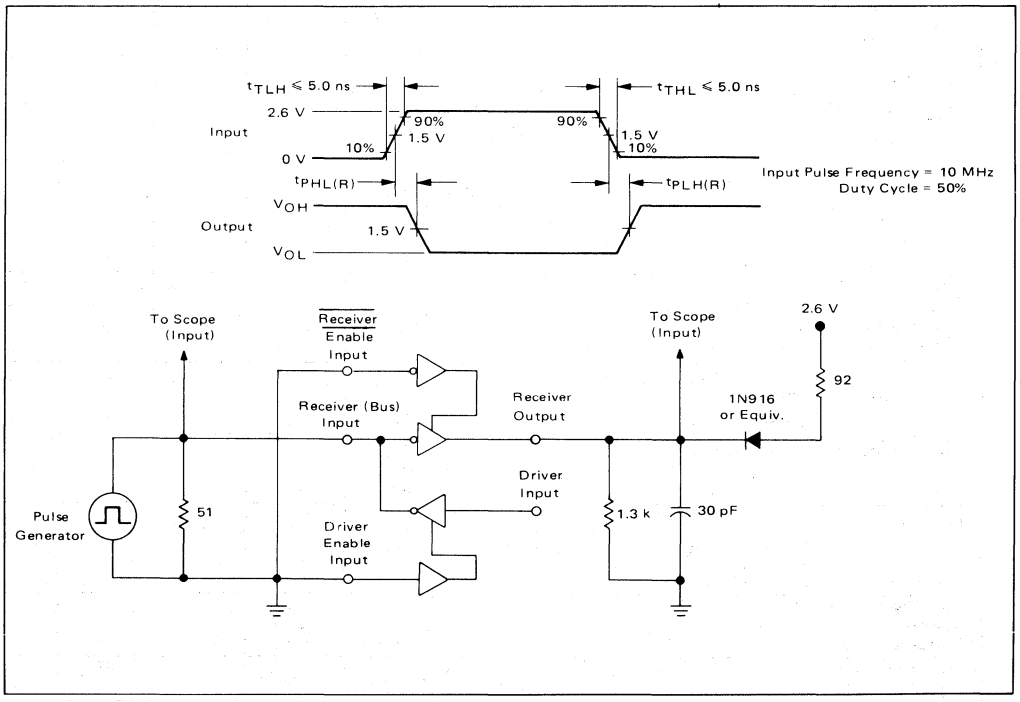


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

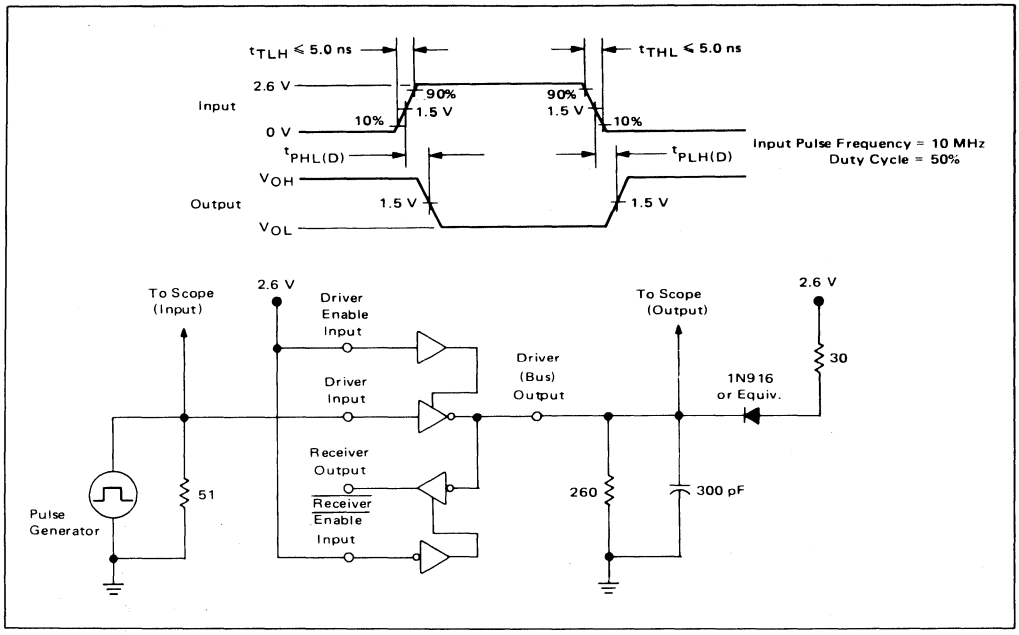


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

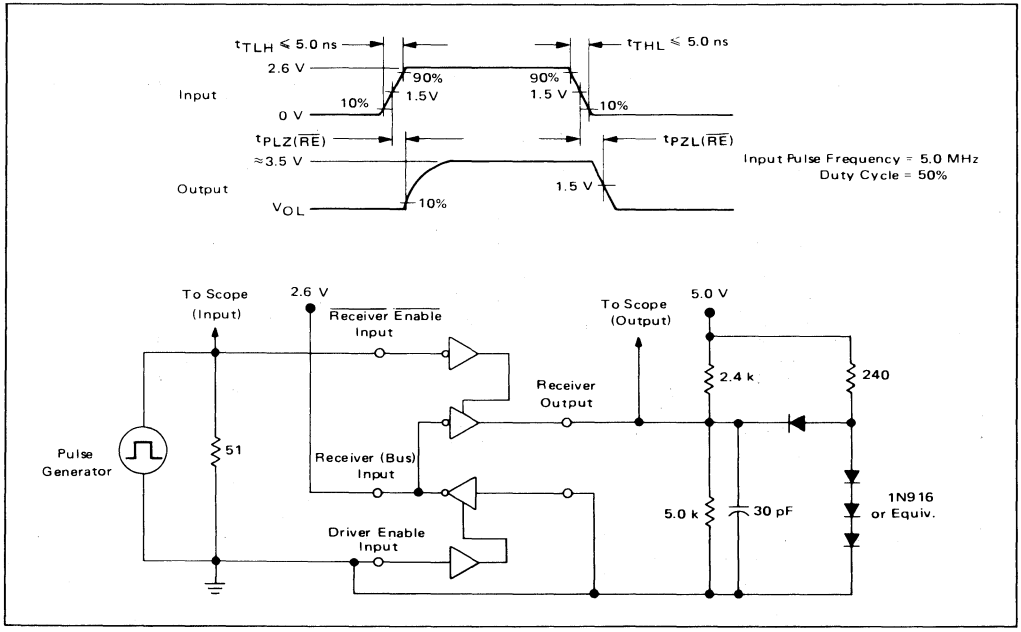


FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

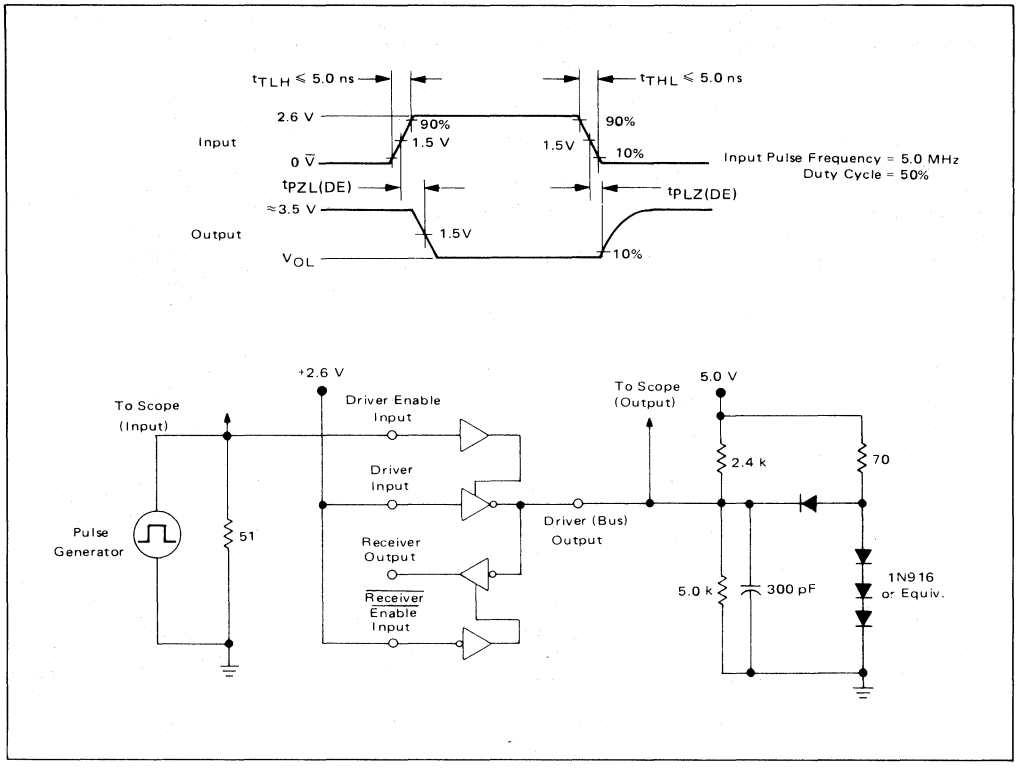
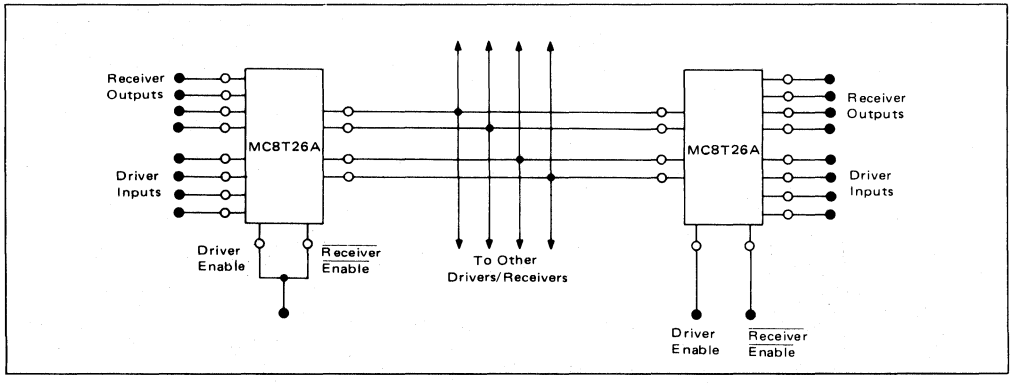


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS



MOTOROLA

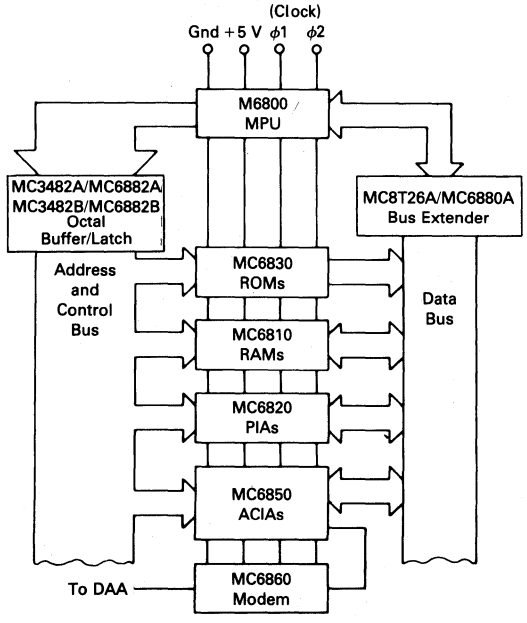
MC3482A/MC6882A
MC3482B/MC6882B

OCTAL THREE-STATE BUFFER/LATCH

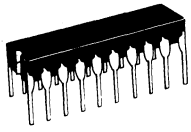
This series of devices combines four features usually found desirable in bus-oriented systems: 1) High impedance logic inputs insure that these devices do not seriously load the bus; 2) Three-state logic configuration allows buffers not being utilized to be effectively removed from the bus; 3) Schottky technology allows for high-speed operation; 4) 48 mA drive capability.

- Inverting and Non-Inverting Options of Data
- SN74S373 Function Pinouts
- Eight Transparent Latches/Buffers in a Single Package
- Full Parallel-Access for Loading and Reloading
- Buffered Control Inputs
- All Inputs Have Hysteresis to Improve Noise Rejection
- High Speed – 8.0 ns (Typ)
- Three-State Logic Configuration
- Single +5 V Power Supply Requirement
- Compatible with 74S Logic or M6800 Microprocessor Systems
- High Impedance PNP Inputs Assure Minimal Loading of the Bus

MICROPROCESSOR BUS EXTENDER APPLICATION

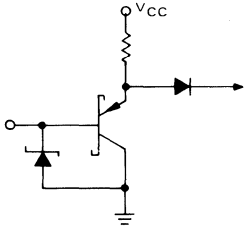


OCTAL THREE-STATE BUFFER/LATCH

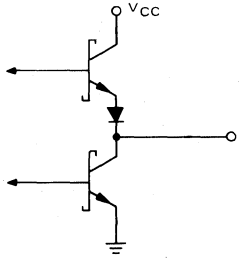


L SUFFIX
CASE 732-03

INPUT EQUIVALENT CIRCUIT



OUTPUT EQUIVALENT CIRCUIT



ORDERING INFORMATION

(Temperature Range for the following devices = 0 to +75°C.)

Device	Alternate	Package
MC3482AL	MC6882AL	Ceramic DIP
MC3482BL	MC6882BL	Ceramic DIP

3

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	–65 to +150	°C
Operating Junction Temperature Ceramic Package	T_J	 175	°C

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, 0°C ≤ T_A ≤ 75°C and 4.75 V ≤ V_{CC} ≤ 5.25 V)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Voltage – High Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IH}	2.0	–	–	V
Input Voltage – Low Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IL}	–	–	0.8	V
Input Current – High Logic State (V_{CC} = 5.25 V, V_{IH} = 2.4 V)	I_{IH}	–	–	40	μA
Input Current – Low Logic State (V_{CC} = 5.25 V, V_{IL} = 0.5 V, $V_{IL(\overline{OE})}$ = 0.5 V)	I_{IL}	–	–	–250	μA
Output Voltage – High Logic State (V_{CC} = 4.75 V, I_{OH} = –20 mA)	V_{OH}	2.4	–	–	V
Output Voltage – Low Logic State (I_{OL} = 48 mA)	V_{OL}	–	–	0.5	V
Output Current – High Impedance State (V_{CC} = 5.25 V, V_{OH} = 2.4 V) (V_{CC} = 5.25 V, V_{OL} = 0.5 V)	I_{OZ}	 – –	 – –	 100 –100	μA
Output Short-Circuit Current (V_{CC} = 5.25 V, V_O = 0) (only one output can be shorted at a time)	I_{OS}	–30	–80	–130	mA
Power Supply Current (V_{CC} = 5.25 V) MC3482A/MC6882A MC3482B/MC6882B	I_{CC}	–		130 150	mA
Input Clamp Voltage (V_{CC} = 4.75 V, I_{IK} = –12 mA)	V_{IK}	–	–	–1.2	V

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, 0°C ≤ T_A ≤ +75°C, unless otherwise noted, typical @ T_A = 25°C.)

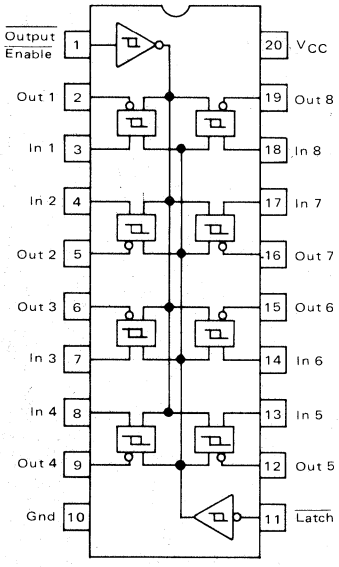
Characteristics	Symbol	MC3482A/ MC6882A			MC3482B/ MC6882B			Unit
		Min	Typ	Max	Min	Typ	Max	
Propagation Delay Times								ns
Data to Output								
Low to High	$t_{PLH(D)}$							
C_L = 50 pF		4.0	9.0	16	4.0	9.0	16	
C_L = 250 pF		–	12	20	–	12	20	
C_L = 375 pF		–	14	22	–	14	22	
C_L = 500 pF		10	16	24	10	16	24	
High to Low	$t_{PHL(D)}$							
C_L = 50 pF		4.0	8.0	16	4.0	8.0	16	
C_L = 250 pF		–	15	22	–	15	22	
C_L = 375 pF		–	18	25	–	17	24	
C_L = 500 pF		16	21	28	14	18	27	
Propagation Delay Times								ns
Latch Disable (Low to High) to Output								
Low to High	$t_{PLH(L)}$							
C_L = 50 pF		–	22	30	–	18	30	
High to Low	$t_{PHL(L)}$							
C_L = 50 pF		–	23	30	–	14	25	
Propagation Delay Times (C_L = 20 pF)								ns
High Output Level to High Impedance	$t_{PHZ(\overline{OE})}$	–	8.0	15	–	6.0	13	
Low Output to High Impedance	$t_{PLZ(\overline{OE})}$	–	20	27	–	15	23	
High Impedance to High Output	$t_{PZH(\overline{OE})}$	–	9.0	16	–	11	18	
High Impedance to Low Output	$t_{PZL(\overline{OE})}$	–	13	20	–	9.0	16	

AC SETUP CHARACTERISTICS (V_{CC} = 5.0 V, 0°C ≤ T_A ≤ +75°C, unless otherwise noted, typical @ T_A = 25°C.)

Characteristic	Symbol	MC3482A/ MC6882A			MC3482B/ MC6882B			Unit
		Min	Typ	Max	Min	Typ	Max	
Setup Time (Data to Negative Going Latch Enable)	$t_{su(D)}$	10	0	–	7.0	0	–	ns
Hold Time (Data to Negative Going Latch Enable)	$t_{h(D)}$	10	–	–	8.0	–	–	ns
Minimum Latch Enable Pulse Width (High or Low)	$t_{W(L)}$	–	15	–	–	15	–	ns

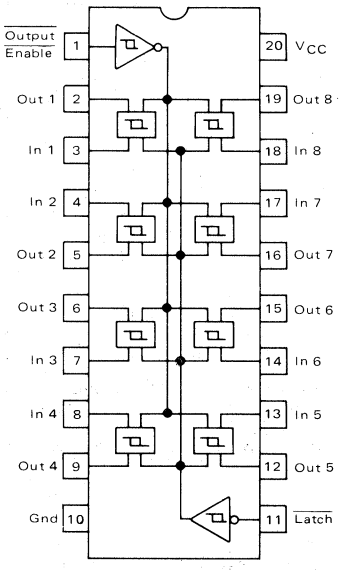
PIN CONNECTIONS AND TRUTH TABLES

MC3482A/MC6882A



Output Enable	Latch	Input	Output
0	1	0	1
0	1	1	0
0	0	X	Q_0
1	X	X	Z

MC3482B/MC6882B



Output Enable	Latch	Input	Output
0	1	0	0
0	1	1	1
0	0	X	Q_0
1	X	X	Z

FIGURE 1 – TEST CIRCUIT FOR SWITCHING CHARACTERISTICS

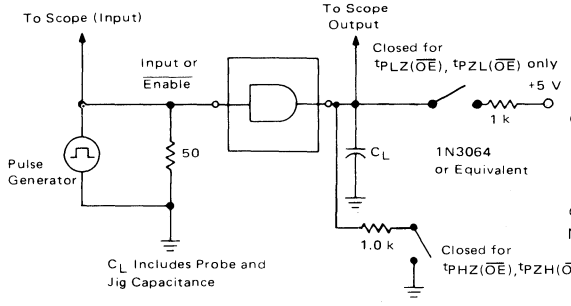


FIGURE 2 – WAVEFORMS FOR PROPAGATION DELAY TIMES DATA TO OUTPUT

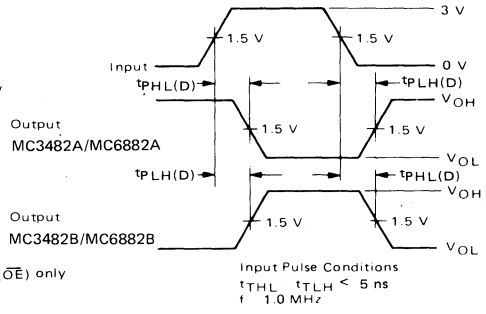


FIGURE 3 – WAVE FORMS FOR AC SETUP AND LATCH DISABLE TO OUTPUT DELAY

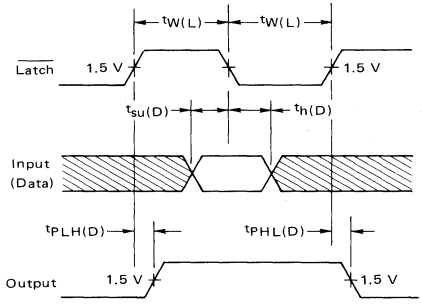
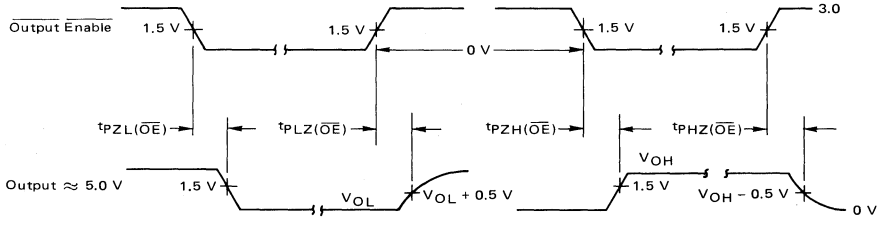


FIGURE 4 – WAVEFORMS FOR PROPAGATION DELAY TIMES – $\overline{\text{OUTPUT ENABLE}}$ TO OUTPUT



MOTOROLA

SN74LS783
MC6883

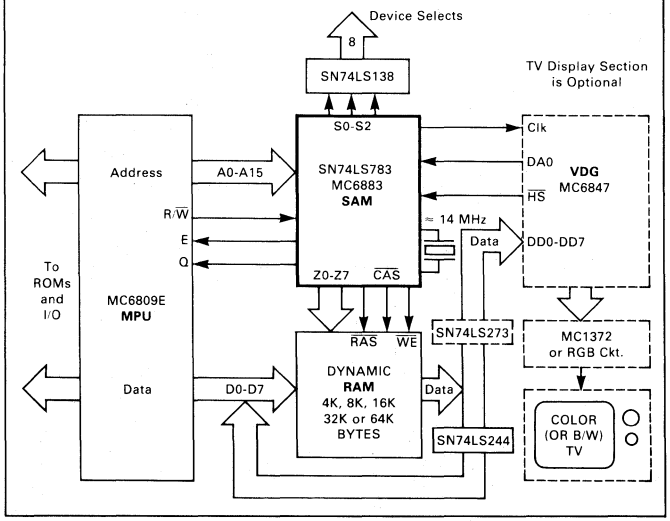
Advance Information

SYNCHRONOUS ADDRESS MULTIPLEXER

The SN74LS783/MC6883 brings together the MC6809E (MPU), the MC6847 (Color Video Display Generator) and dynamic RAM to form a highly effective, compact and cost effective computer and display system.

- MC6809E, MC6800, MC6801E, MC68000 and MC6847 (VDG) Compatible
- Transparent MPU/VDG/Refresh
- RAM size — 4K, 8K, 16K, 32K or 64K Bytes (Dynamic or Static)
- Addressing Range — 96K Bytes
- Single Crystal Provides All Timing
- Register Programmable:
  - VDG Addressing Modes
  - VDG Offset (0 to 64K)
  - RAM Size
  - Page Switch
  - MPU Rate (Crystal ÷ 16 or ÷ 8)
  - MPU Rate (Address Dependent or Independent)
- System "Device Selects" Decoded 'On Chip'
- Timing is Optimized for Standard Dynamic RAMs
- +5.0 V Only Operation
- Easy Synchronization of Multiple SAM Systems
- DMA Mode

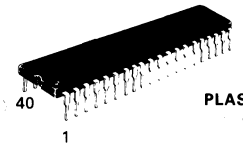
SYSTEM BLOCK DIAGRAM



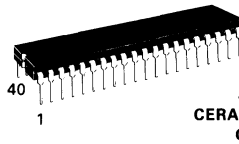
This document contains information on a new product. Specifications and information herein are subject to change without notice.

SYNCHRONOUS ADDRESS MULTIPLEXER

LOW POWER SCHOTTKY

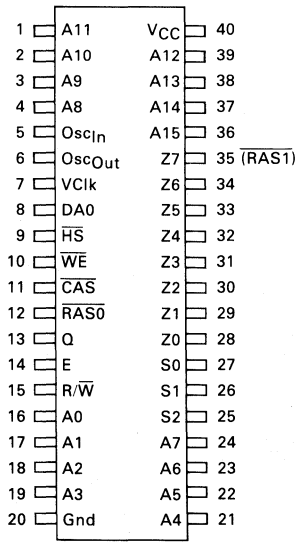


N SUFFIX
PLASTIC PACKAGE
CASE 711



J SUFFIX
CERAMIC PACKAGE
CASE 734

PIN ASSIGNMENT



MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	−0.5 to +7.0	Vdc
Input Voltage (Except Osc_{In})	V_I	−0.5 to 10	Vdc
Input Current (Except Osc_{In})	I_I	−30 to +5.0	mA
Output Voltage	V_O	−0.5 to +7.0	Vdc
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C
Input Voltage Osc_{In}	$V_{IOsc_{In}}$	−0.5 to V_{CC}	Vdc
Input Current Osc_{In}	$I_{IOsc_{In}}$	−0.5 to +5.0	mA

GUARANTEED OPERATING RANGES

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Operating Ambient Temperature Range	T_A	0	25	75	°C
Output Current High	I_{OH}				mA
$\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$		—	—	−1.0	
All Other Outputs		—	—	−0.2	
Output Current Low	I_{OL}				mA
$\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$		—	—	8.0	
VClk		—	—	0.8	
All Other Outputs		—	—	4.0	

DC CHARACTERISTICS (Unless otherwise noted specifications apply over recommended power supply and temperature ranges.)

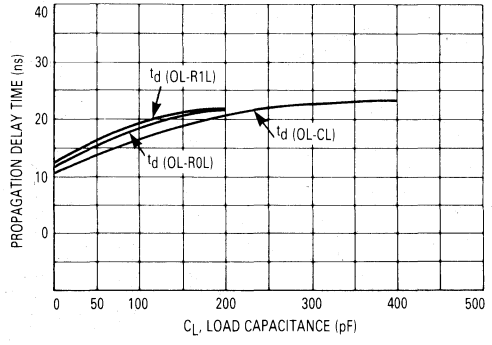
Characteristic	Symbol	Min	Typ	Max	Units
Input Voltage — High Logic State	V_{IH}	2.0	—	—	V
Input Voltage — Low Logic State	V_{IL}	—	—	0.8	V
Input Clamp Voltage (V_{CC} = Min, I_{in} = −18 mA) All Inputs Except Osc_{In}	V_{IK}	—	—	−1.5	V
Input Current — High Logic State at Max Input Voltage	I_I				μA
(V_{CC} = Max, V_{in} = 5.25 V) VClk Input		—	—	200	
(V_{CC} = Max, V_{in} = 5.25 V) DA0 Input		—	—	100	
(V_{CC} = Max, V_{in} = 5.25 V Osc_{In} = Gnd) Osc_{Out} Input		—	—	250	
(V_{CC} = Max, V_{in} = 7.0 V) All Other Inputs Except Osc_{In}		—	—	100	
Input Current High Logic State All Inputs Except VClk,	I_{IH}				μA
(V_{CC} = Max, V_{in} = 2.7 V) DA0 Osc_{In}, Osc_{Out}		—	—	20	
Input Current — Low Logic State	I_{IL}				mA
(V_{CC} = Max, V_{in} = 0.4 V) DA0 Input		—	—	−1.2	
(V_{CC} = Max, V_{in} = 0.4 V) VClk Input		—	−30	−60	
(V_{CC} = Max, V_{in} = 0.4 V, Osc_{In} = Gnd) Osc_{Out} Input		—	—	−8	
(V_{CC} = Max, V_{in} = 0.4 V) All Other Inputs Except Osc_{In}		—	—	−.4	
Output Voltage — High Logic State					V
(V_{CC} = Min, I_{OH} = −1.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$	$V_{OH(C)}$	3.0	—	—	
(V_{CC} = Min, I_{OH} = −0.2 mA) E, Q	$V_{OH(E)}$	V_{CC} − 0.75	—	—	
(V_{CC} = Min, I_{OH} = −0.2 mA) All Other Outputs	V_{OH}	2.7	—	—	
Output Voltage — Low Logic State					V
(V_{CC} = Min, I_{OL} = 8.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$	$V_{OL(C)}$	—	—	0.5	
(V_{CC} = Min, I_{OL} = 4.0 mA) E, Q Outputs	$V_{OL(E)}$	—	—	0.5	
(V_{CC} = Min, I_{OL} = 0.8 mA) VClk Output	$V_{OL(V)}$	—	—	0.6	
(V_{CC} = Min, I_{OL} = 4.0 mA) All Other Outputs	V_{OL}	—	—	0.5	
Power Supply Current	I_{CC}	—	180	230	mA
Output Short-Circuit Current	I_{OS}	30	—	225	mA

AC CHARACTERISTICS (4.75 V≤V_{CC}≤5.25 V and 0≤T_A≤70°C, unless otherwise noted).

Characteristic	Symbol	Min	Typ	Max	Units
Propagation Delay Times					ns
(See Circuit in Figure 9) Oscillator-In ↘ to Oscillator-Out ↗	$t_{d(OL-OH)}$	—	3.0	—	
Oscillator-In ↗ to Oscillator-Out ↘	$t_{d(OH-OL)}$	—	20	—	
(C_L = 195 pF) A0 thru A15 to Z0, Z1, Z2 thru Z7	$t_{d(A-Z)}$	—	28	—	
(C_L = 30 pF) A0 thru A15, R/W to S0, S1, S3	$t_{d(A-S)}$	—	18	—	
(C_L = 95 pF) Oscillator-Out ↘ to $\overline{RAS0}$ ↗	$t_{d(OL-R0H)}$	—	20	—	
(C_L = 95 pF) Oscillator-Out ↘ to $\overline{RAS0}$ ↘	$t_{d(OL-R0L)}$	—	18	—	
(C_L = 95 pF) Oscillator-Out ↘ to $\overline{RAS1}$ ↗	$t_{d(OL-R1H)}$	—	22	—	
(C_L = 95 pF) Oscillator-Out ↘ to $\overline{RAS1}$ ↘	$t_{d(OL-R1L)}$	—	20	—	
(C_L = 195 pF) Oscillator-Out ↘ to $\overline{CAS}$ ↗	$t_{d\ (OL-CH)}$	—	20	—	
(C_L = 195 pF) Oscillator-Out ↘ to $\overline{CAS}$ ↘	$t_{d(OL-CL)}$	—	20	—	
(C_L = 195 pF) Oscillator-Out ↘ to $\overline{WE}$ ↗	$t_{d(OL-WH)}$	—	22	—	
(C_L = 195 pF) Oscillator-Out ↘ to $\overline{WE}$ ↘	$t_{d(OL-WL)}$	—	40	—	
(C_L = 100 pF) Oscillator-Out ↘ to E ↗	$t_{d(OL-EH)}$	—	55	—	
(C_L = 100 pF) Oscillator-Out ↘ to E ↘	$t_{d(OL-EL)}$	—	25	—	
(C_L = 100 pF) Oscillator-Out ↘ to Q ↗	$t_{d(OL-QH)}$	—	55	—	
(C_L = 100 pF) Oscillator-Out ↘ to Q ↘	$t_{d(OL-QL)}$	—	25	—	
(C_L = 30 pF) Oscillator-Out ↗ to VClk ↗	$t_{d(OH-VH)}$	—	50	—	
(C_L = 30 pF) Oscillator-Out ↗ to VClk ↘	$t_{d(OH-VL)}$	—	65	—	
(C_L = 195 pF) Oscillator-Out ↘ to Row Address	$t_{d(OL-AR)}$	—	36	—	
(C_L = 195 pF) Oscillator-Out ↘ to Column Address	$t_{d(OL-AC)}$	—	33	—	
(C_L = 15 pF) Oscillator-Out ↘ to DA0 ↗ Earliest(1)	$t_{d(OL-DH)}$	—	−15	—	
(C_L = 15 pF) Oscillator-Out ↘ to DA0 ↗ Latest(1)	$t_{d(OL-DH)}$	—	+15	—	
(C_L = 95 pF on $\overline{RAS}$, C_L = 195 pF on $\overline{CAS}$) $\overline{CAS}$ ↘ to $\overline{RAS}$ ↗	$t_{d(CL-RH)}$	—	208	—	
Setup Time for A0 thru A15, R/$\overline{W}$ Rate = ÷16	$t_{su(A)}$	—	28	—	ns
Rate = ÷ 8		—	28		
Hold Time for A0 thru A15, R/$\overline{W}$ Rate = ÷16	$t_{h(A)}$	—	30	—	ns
Rate = ÷ 8		—	30	—	
Width of $\overline{HS}$ Low 2	$t_{WL(HS)}$	2.0	5.0	6.0	µs

Notes: 1. When using the SAM with an MC6847, the rising edge of DA0 is confined within the range shown in the timing diagrams (unless the synchronizing process is incomplete.) The synchronization process requires a maximum of 32 cycles of Osc_{Out} for completion.
2. $t_{WL(HS)}$ wider than 6.0 µs may yield more than 8 sequential refresh addresses.

FIGURE 1 — PROPAGATION DELAY TIMES VERSUS LOAD CAPACITANCE

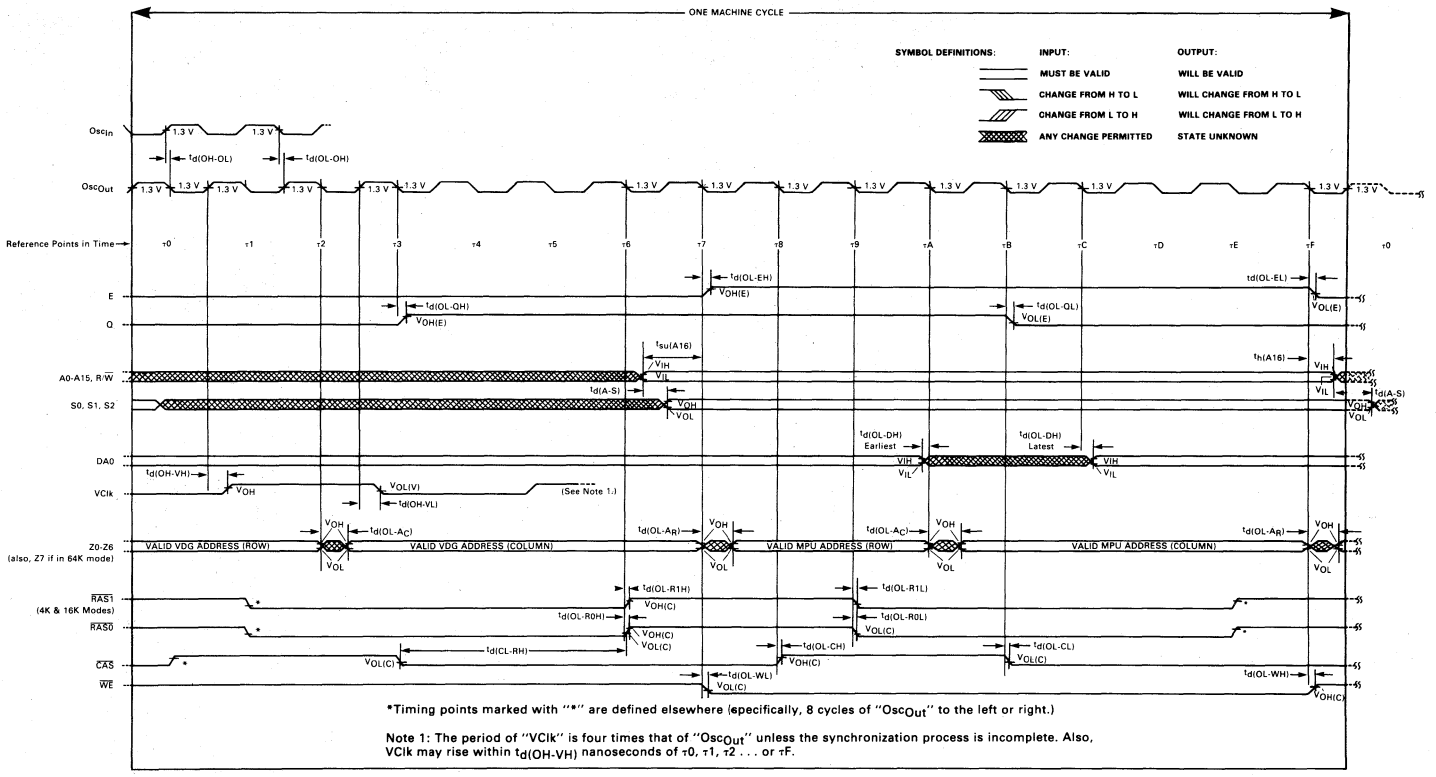


PIN DESCRIPTION TABLE

		Name	No.	Function
Input Pins	Power	V_{CC}	40	Apply + 5 volts ± 5%. SAM draws less than 230 mA.
		Gnd	20	Return Ground for +5 volts.
	MPU Address and Control	A15	36	Most Significant Bit.
		A14	37	
		A13	38	MPU address bits A0-A15. These 16 signals come directly from the MPU and are used to directly address up to 64K memory locations or to indirectly address up to 96K memory locations. (See pages 17 and 18 for memory maps). Each input is approximately equivalent to one low power Schottky load.
		A12	39	
		A11	1	
		A10	2	
		A9	3	
		A8	4	
		A7	24	
		A6	23	
		A5	22	
		A4	21	
		A3	19	
		A2	18	
		A1	17	
		A0	16	Least Significant Bit.
		R/$\overline{W}$	15	MPU READ or $\overline{WRITE}$. This signal comes directly from the MPU and is used to enable writing to the SAM control register, dynamic RAM (via $\overline{WE}$), and to enable device select #0.
	VDG Control	Osc_{In}	5	Apply 14.31818* MHz crystal and 2.5–30 pF trimmer to ground. See page 12.
		DA0	8	Display Address DA0. The primary function of this pin is to input the least significant bit of a 16-bit video display address. The more significant 15-bits are outputs from an internal 15-bit counter which is clocked by DA0. The secondary function of this pin is to indirectly input the logic level of the VDG "$\overline{FS}$" (field synchronization pulse) for vertical video address updating.
		$\overline{HS}$	9	Horizontal Synchronization. The primary function of this pin is to detect the falling edge of VDG "$\overline{HS}$" pulse in order to initiate eight dynamic RAM refresh cycles. The secondary function is to reset up to 4 least significant bits of the internal video address counter.
		VClk	7	VDG Clock. The primary function of this pin is to **output** a 3.579545 MHz square wave** to the VDG "Clk" pin. The secondary function resets the SAM when this VClk pin is pulled to logic "0" level, acting as an **input**.
Output Pins		Osc_{Out}	6	Apply 1.5 kΩ resistor to 14.31818* MHz crystal and 33 pF capacitor to ground. See page 12.
	Device Selects	S2	25	Most Significant Bit (Device Select Bits). The binary value of S2, S1, S0 selects one of eight "chunks" of MPU address space (numbers 0 through 7). Varying in length, these "chunks" provide efficient memory mapping for ROMs, RAMs, Input/Output devices, and MPU Vectors. (Requires 74LS 138-type demultiplexer).
		S1	26	
		S0	27	Least Significant Bit.
	MPU Clocks	E	14	E (Enable Clock) "E" and "Q" are 90° out of phase and are both used as MPU clocks for the MC6809E. For the MC6800 and MC6801E, only "E" is used. "E" is also used for many MC6800 peripheral chips.
		Q	13	Q (Quadrature Clock).
	RAM Address	Z7†	35	Most Significant Bit
		Z6†	34	First, the least significant address bits from the MPU or "VDG" are presented to Z0–Z5 (4K x 1 RAMs) or Z0–Z6 (16K x 1 RAMs) or Z0–Z7 (64K x 1 RAMs). Next, the most significant address bits from the MPU or "VDG" are presented to Z0–Z5 (4K x 1 RAMs) or Z0–Z6 (16K x 1 RAMs) or Z0–Z7 (64K x 1 RAMs). Note that for 4K x 1 and 16K x 1 RAMs, Z7 (Pin 35) is not needed for address information. Therefore, Pin 35 is used for a second row address select which is labeled ($\overline{RAS1}$).
		Z5†	33	
		Z4†	32	
		Z3†	31	
		Z2†	30	
		Z1†	29	
		Z0†	28	Least Significant Bit.
	RAM Control	$\overline{RAS1}$†	35	Row Address Strobe One. This pulse strobes the least significant 6,7 or 8 address bits into dynamic RAMs in Bank #1.
		$\overline{RAS0}$†	12	Row Address Strobe Zero. This pulse strobes the least significant 6,7 or 8 address bits into dynamic RAMs in Bank #0.
		$\overline{CAS}$†	11	Column Address Strobe. This pulse strobes the most significant 6,7 or 8 address bits into dynamic RAMs.
		$\overline{WE}$†	10	Write Enable. When low, this pulse enables the MPU to write into dynamic RAM.

*14.31818 MHz is 4 times 3.579545 MHz television color subcarrier. Other frequencies may be used. (See page 12.)
**When VDG and SAM are not yet synchronized the "square wave" will stretch (see page 10.)
† Due to fast transitions, ferrite beads in series with these outputs may be necessary to avoid high frequency (≈ 60 MHz) resonances.

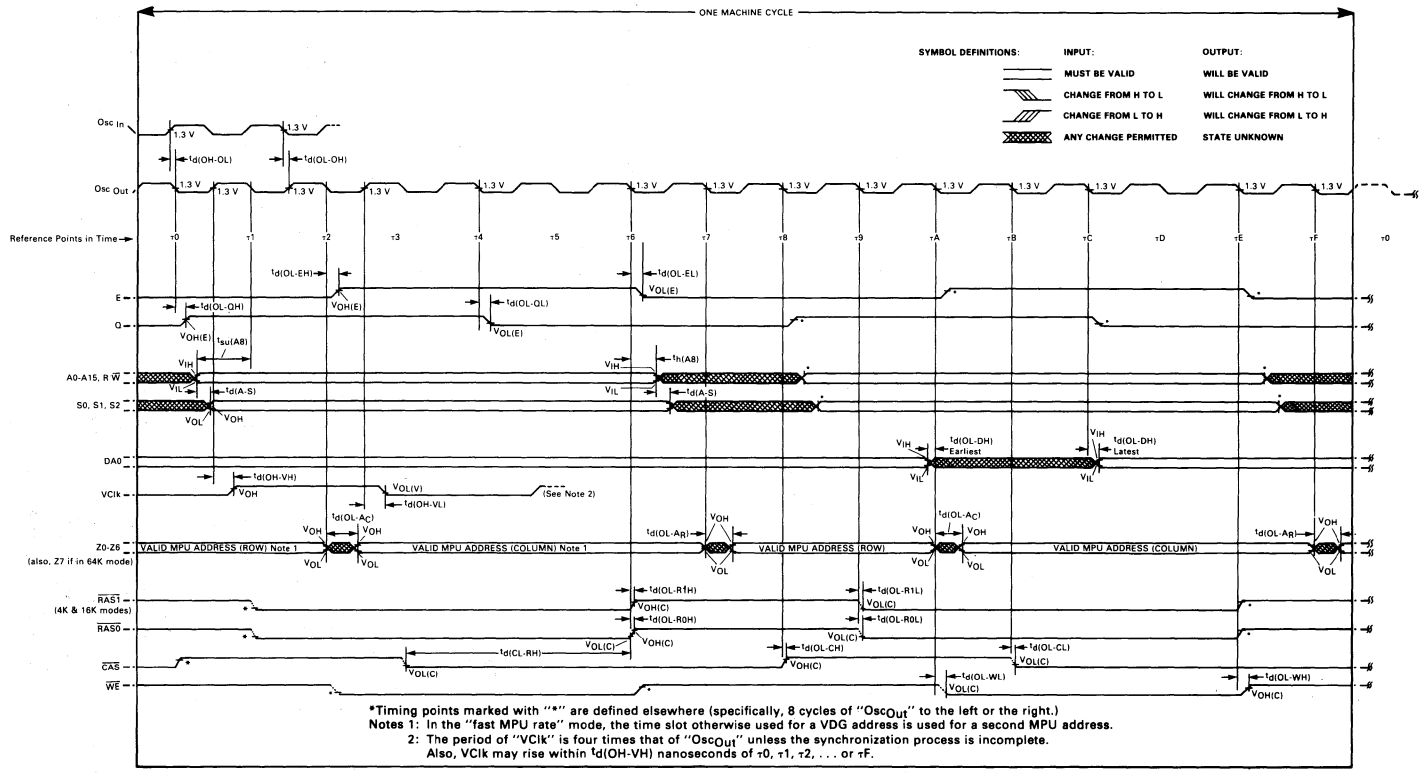
FIGURE 2 — TIMING WAVEFORMS for MPU RATE = SLOW



Timing points marked with "" are defined elsewhere (specifically, 8 cycles of "Osc_{Out}" to the left or right.)

Note 1: The period of "VClk" is four times that of "Osc_{Out}" unless the synchronization process is incomplete. Also, VClk may rise within $t_{d(OH-VH)}$ nanoseconds of τ0, τ1, τ2 . . . or τF.

FIGURE 3— TIMING WAVEFORMS for MPU RATE = FAST

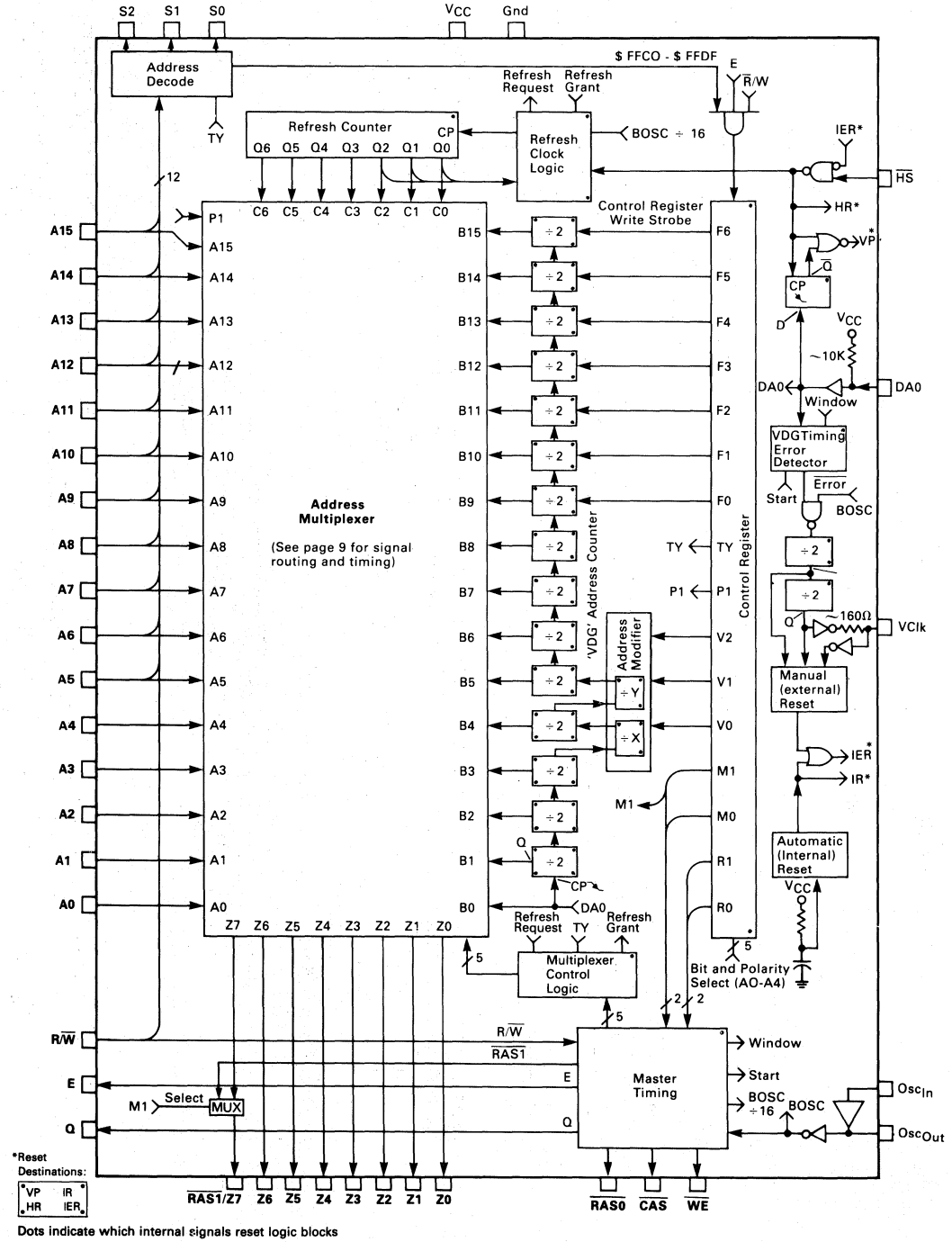


Timing points marked with "" are defined elsewhere (specifically, 8 cycles of "Osc_{Out}" to the left or the right.)

Notes 1: In the "fast MPU rate" mode, the time slot otherwise used for a VDG address is used for a second MPU address.

2: The period of "VClk" is four times that of "Osc_{Out}" unless the synchronization process is incomplete. Also, VClk may rise within $t_{d(OH-VH)}$ nanoseconds of τ0, τ1, τ2, . . . or τF.

FIGURE 4 — SAM BLOCK DIAGRAM



Dots indicate which internal signals reset logic blocks

SAM BLOCK DIAGRAM DESCRIPTION

MPU Addresses (A0 – A15):

These 16 signals come directly from the MPU and are used to directly address up to 64K memory locations (K = 1024) or to indirectly address up to 96K memory locations, by using a paging bit "P" (see pages 17 and 18 for memory maps.) Each input is approximately equivalent to one low power Schottky load.

VDG Address Counter (B0 – B15):

These 16 signals are derived from one input (DA0) which is the least significant bit of the VDG address. Most of the counter is simply binary. However, to duplicate the various addressing modes of the MC6847 VDG, ADDRESS MODIFIER logic is used. Selected by three VDG mode bits (V2, V1, and V0) from the SAM CONTROL REGISTER, eight address modifications are obtained as shown in Figure 5.

Also, notice that bits B9-B15 may be loaded from bits F0-F6 from the CONTROL REGISTER. This allows the starting address of the VDG display to be offset (in ½K increments) from $0000 to $FFFF†. B9-B15 are loaded when a VERTICAL PRE-LOAD(VP) pulse is generated. VP goes active (high) when $\overline{HS}$ from the VDG rises if DA0 is high (or a high impedance.) This condition should occur only while the TV electron beam is in vertical blanking and is simply implemented by connecting $\overline{FS}$ and $\overline{MS}$ together on the MC6847. The VP pulse also **clears** bits B1 – B8.

Finally, a HORIZONTAL RESET (HR) pulse may also affect the counter by clearing bits B1 – B3 or B1 – B4 when $\overline{HS}$ from the VDG is LOW (see Figure 5.) The HR pulse should occur only while the TV electron beam is in horizontal blanking.

In summary, DA0 clocks the VDG ADDRESS COUNTER; HR initializes the horizontal portion and VP initializes the vertical portion of the VDG ADDRESS COUNTER.

REFresh Address Counter (C0 – C6):

A seven bit binary counter with outputs labeled C0 – C6 supplies bursts of eight* sequential addresses triggered by a $\overline{HS}$ high to low transition. Thus, while the TV electron beam is in horizontal blanking, eight sequential addresses are accessed. Likewise, the next eight addresses are accessed during the next horizontal blanking period, etc. In this manner, all 128 addresses are refreshed in less than 1.1 milliseconds.

Address Multiplexer:

Occupying a large portion of the block diagram in Figure 4, is the address multiplexer which outputs bits Z0-Z7 (as addresses to dynamic RAM's.) Inputs to the address multiplexer include the VDG address (B0 – B15) the REFresh address (C0- C6) and the MPU address (A0 – A15) or (A0 – A14 plus one paging bit "P".) The paging bit "P" is one bit in the SAM CONTROL REGISTER that is used in place of A15 when memory map TYpe #0 is selected (via the SAM CONTROL REGISTER "TY" bit.)

Figure 6 shows which inputs are routed to Z0 – Z7 and **when** the routing occurs relative to one SAM machine cycle. Notice that Z7 and $\overline{RAS1}$ share the same pin. Z7 is selected if "M1" in the SAM CONTROL REGISTER IS HIGH (Memory size = 64K.)

Address Decode:

At the top left of Figure 4, is the Address Decode block. Outputs S2, S1, and S0 form a three bit encoded binary word(S). Thus S may be one of eight values (0 through 7) with each value representing a different range of MPU addresses. (To enable peripheral ROM's or I/O, decode the S2, S1, and S0 bits into eight seperate signals by using a 74LS138, 74LS155 or 74LS156. Notice that S2, S1, and S0 are **not** gated with any timing signals such as E or Q.)

Along with the A5 – A15 inputs is the MEMORY MAP TYpe bit (TY.) This bit is soft-programmable (as are **all** 16 bits in the SAM CONTROL REGISTER,) and selects one of two memory maps. Memory map #0 is intended to be used in systems that are primarily **ROM** based. Whereas, memory map #1 is intended for a primarily **RAM** based system with 64K contiguous RAM locations (minus 256 locations.) The various meanings of S2, S1, S0 are tabulated in Figure 16 (page 19) and again on pages 17 and 18.

In addition to S2, S1, and S0 outputs is a decode of $FFC0 through $FFDF which, when gated with E and $\overline{R/W}$, results in the write strobe for the SAM CONTROL REGISTER.

SAM Control Register

As shown in Figure 4, the CONTROL REGISTER has 16 "outputs":

VDG Addressing Modes:	V2, V1, V0	MPU Rate:	R1, R0
VDG Address OFFset:	F6, F5, F4, F3, F2, F1, F0	Memory Size (RAM):	M1, M0
32K Page Switch:	P	Memory Map TYpe:	TY

When the SAM is reset (see page 10,) all 16 bits are cleared. To **set** any one of these 16 bits, the MPU simply **writes** to a unique** **odd** address (within $FFC1 through $FFDF.) To **clear** any one of these 16 bits, the MPU

* If $\overline{HS}$ is held low longer than 8 μs, then the number of sequential addresses in one refresh "BURST" is proportional to the time interval during which $\overline{HS}$ is low.

** See pages 17 or 18 for specific addresses.

† In this document, the "$" symbol always preceeds hexidecimal characters.

simply writes to a unique** **even** address (within $FFCO through $FFDE.) Note that the **data** on the MPU data bus is irrelevant.

Inputs to the control register include A4, A3, A2, A1 (which are used to select **which one** of 16 bits is to be cleared or set), A0 (which determines the polarity . . . clear or set,) and R/W, E and $FFCO – $FFDF (which restrict the method, timing and addresses for changing one of the 16 bits.) For more detailed descriptions of the purposes of the 16 control bits, refer to related sections in the BLOCK DIAGRAM DESCRIPTION (pages 8 through 12) and the PROGRAMMING GUIDE (pages 14 through 18).

** See pages 17 or 18 for specific addresses.

FIGURE 5 — VDG ADDRESS MODIFIER

Mode			Division Variables		Bits Cleared by $\overline{HS}$ (low)
V2	V1	V0	X	Y	
0	0	0	1	12	B1–B4
0	0	1	3	1	B1–B3
0	1	0	1	3	B1–B4
0	1	1	2	1	B1–B3
1	0	0	1	2	B1–B4
1	0	1	1	1	B1–B3
1	1	0	1	1	B1–B4
1	1	1	1	1	None (DMA MODE)

FIGURE 6 — SIGNAL ROUTING for ADDRESS MULTIPLEXER

Memory Size		Signal	Row/Column	Signals Routed to Z0–Z7								Timing
M1	M0	Source		Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0	(Figure 2)
4K	**0**	**0**	MPU	ROW	*	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	*	L	A11	A10	A9	A8	A7	A6	TA-TF
			VDG	ROW	*	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	*	L	B11	B10	B9	B8	B7	B6	T2-T7
			REF	ROW	*	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	*	L	L	L	L	L	L	L	T2-T7
16K	**0**	**1**	MPU	ROW	*	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	*	A13	A12	A11	A10	A9	A8	A7	TA-TF
			VDG	ROW	*	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	*	B13	B12	B11	B10	B9	B8	B7	T2-T7
			REF	ROW	*	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	*	L	L	L	L	L	L	L	T2-T7
64K (dynamic)	**1**	**0**	MPU	ROW	A7	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	P/A15***	A14	A13	A12	A11	A10	A9	A8	TA-TF
			VDG	ROW	B7	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	B15	B14	B13	B12	B11	B10	B9	B8	T2-T7
			REF	ROW	L	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	L	L	L	L	L	L	L	L	T2-T7
64K (static)	**1**	**1**	MPU	ROW	A7	A6	A5	A4	A3	A2	A1	A0	T7-**T9**
				COL	P/A15***	A14	A13	A12	A11	A10	A9	A8	**T9**-TF
			VDG	ROW	B7	B6	B5	B4	B3	B2	B1	B0	TF-**T1**
				COL	B15	B14	B13	B12	B11	B10	B9	B8	**T1**-T7
			REF	ROW	L	C6	C5	C4	C3	C2	C1	C0	TF-**T1**
				COL	L	L	L	L	L	L	L	L	**T1**-T7

Notes: "L" implies logical LOW level.
*Z7 functions as RAS1 and its level is address dependent. For example, when using two banks of 16K x 1 RAMs, $\overline{RAS0}$ is active for addresses $0000 to $3FFF and $\overline{RAS1}$ is active for addresses $4000 to $7FFF.
***If Map TYpe = 0, then page bit "P" is the output (otherwise A15).

Internal Reset

By lowering V_{CC} below 0.6 volts for at least one millisecond, a **complete** SAM reset is initiated and is completed within 500 nanoseconds after V_{CC} rises above 4.25 volts.

NOTE: In some applications, (for example, multiple "VDG-RAM" systems controlled by a single MPU) **multiple** SAM ICs can be synchronized as follows:

- Drive all SAM's from one external oscillator.
- Stop external oscillator.
- Lower V_{CC} below 0.6 volts for at least 1.0 millisecond.
- Raise V_{CC} to 5.0 volts.
- Start external oscillator.
- Wait at least 500 nanoseconds.

Now, the "E" clocks from all SAM's should be in-phase.

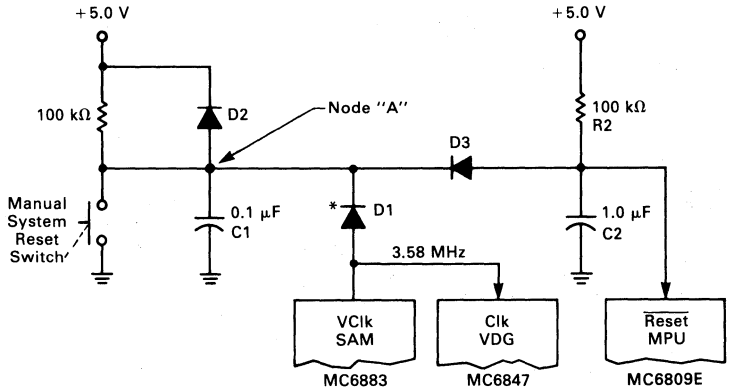
External Reset

When the VClk pin on SAM is forced below 0.8 volts for at least eight cycles of "oscillator-out", the SAM becomes **partially** reset. That is, all bits in the SAM control register are cleared. However, signals such as $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, E or Q are **not** stopped (as they are with an **internal** reset), since the SAM must maintain dynamic RAM refresh even during this external reset period.

Figure 7 shows how VClk can be pulled low through diode D1 when node "A" is low.* When node "A" is high, only the backbiased capacitance of diode D1 loads the 3.58 MHz on VClk. Diode D2 helps discharge C1 (Power-on-Reset capacitor) when power is turned off. Diode D3 allows the MPU reset time constant R2C2 to be greater than the SAM reset time constant. Thereby, ensuring **release** of the SAM reset **prior to** attempting to program the SAM control register.

3

FIGURE 7 — EXTERNAL RESET CIRCUITRY



VDG Synchronization

In order for the VDG and MPU to share the same dynamic RAM (see page 13,) the **VDG clock must be stopped** until the VDG data fetch and MPU data fetch are synchronized as shown in Figure 12. Once synchronized, the VDG clock resumes its 3.579545 MHz rate and is not stopped again unless an extreme temperature change (or SAM reset) occurs. When stopped, the VDG clock remains stopped for **no more than** 32 Osc_{Out} cycles (approximately 2 microseconds.)

In the block diagram in Figure 4, DA0 enters a block labeled VDG Timing Error Detector. If DA0 rises **between** time reference points** τ_A and τ_C, then $\overline{Error}$ is high and VClk is the result of dividing BOSC (Buffered Osc_{Out} ≈ 14 MHz) by four. However, if DA0 rises **outside** the time Window τ_A to τ_C, then $\overline{Error}$ goes LOW and the VDG stops. A START pulse at time reference point τ_B (center of Window) restarts the VDG . . . properly synchronized.

*Use a diode with sufficiently low forward voltage drop to meet V_{IL} requirement at VClk.

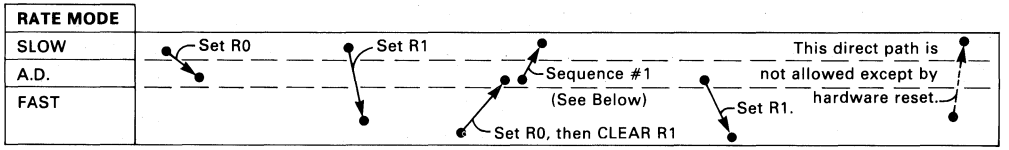
**See timing diagrams on page 5 and 6.

Changing the MPU Rate (by changing SAM control register bits R0, R1).

Two bits in the SAM control register determine the period of both "E" and "Q" MPU clocks. Three rate modes are implemented as follows:

RATE MODE	R1	R0	
SLOW	0	0	The frequency of "E" (and "Q") is f crystal ÷ 16. This rate mode is automatically selected when the SAM is reset. Note that system timing is least critical in this "SLOW" rate mode.
A.D. (Address Dependent)	0	1	The frequency of "E" (and "Q") is either f crystal ÷ 16 or f crystal ÷ 8, depending on the address the MPU is presenting.
FAST	1	X	The frequency of "E" (and "Q") is f crystal ÷ 8. This is accomplished by stealing the time that is normally used for VDG/REFRESH, and using this time for the MPU. Note: Neither VDG display nor dynamic RAM refresh are available in the "FAST" rate mode. (Both are available in SLOW and A.D. rate modes).

When changing between any two of the three rate modes, the following procedures must be followed to ensure that MPU timing specifications are met:



May be ANY address from $0000 to $7FFF.

SEQUENCE #1:

```
7D 00 00  TST #$0000 . . . Synchronizes STA instruction to write during T2-TG (See Figure #8).*
21 00     BRN 00
B7 FF D6  STA #$FFD6 . . . Clears bit R0
```

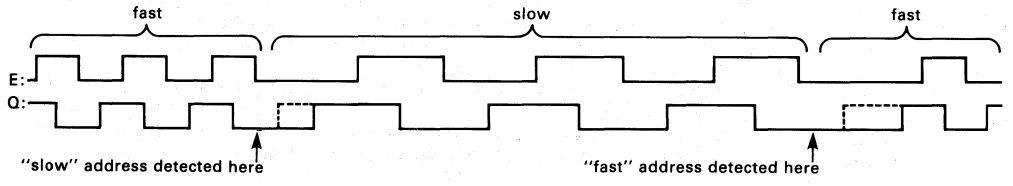
*Note: "TST" instruction affects MC6809E condition code register.

Changing the MPU Rate (In Address Dependent Mode)

When the SAM control register bits "R1", and "R0" are programmed to "0" and "1", respectively, the Address Dependent Rate Mode is selected. In this mode, the ÷ 16 MPU rate is automatically used when addressing within $0000 to $7FFF* or $FF00 to $FF1F ranges. Otherwise the ÷ 8 MPU rate is automatically used. (Refer to Figure 8 for sample "E" and "Q" waveforms yielding ÷ 8 to ÷ 16 and ÷ 16 to ÷ 8 rate changes). This mode often nearly doubles the MPU throughput while still providing transparent VDG and dynamic, RAM refresh functions. For example, since much of the MPU's time may be spent performing internal MPU functions (address = $FFFF)**, accessing ROM (address = $8000 to $FEFF) or accessing I/O (address = $FF20 — $FF5F), the faster f crystal ÷ 8 MPU rate may be used much of the time.

Note: The VDG operates normally when using the SLOW or A.D. rate modes. However, in the FAST rate mode, the VDG is not allowed access to the dynamic RAM.

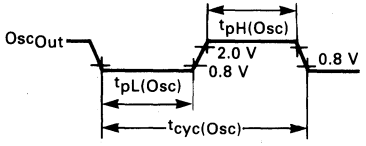
FIGURE 8 — RATE CHANGE E AND Q WAVEFORMS



*When using Memory Map 0, addresses $0000 to $7FFF may access Dynamic RAM.
**The MC6809 outputs $FFFF on A0–A15 when no other valid addresses are being presented.

Oscillator

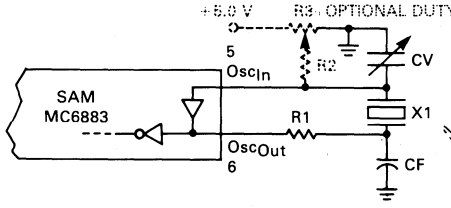
In Figure 4, an amplifier between Osc_{In} and Osc_{Out} provides the gain for oscillation (using a crystal as shown in Figure 9.) Alternately, Pin 5 (Osc_{In}) may be grounded while Pin 6 (Osc_{Out}) may be driven at low-power Schottky levels as shown in Figure 10. Also, see V_{IH}, V_{IL} on page 2.



AC Specifications*

	Max	Typ	Min	Units
$t_{pH(Osc)}$	—	30	22	ns
$t_{pL(Osc)}$	—	30	22	ns
$t_{cyc(Osc)}$	—	70	62.4	ns

FIGURE 9 — CRYSTAL OSCILLATOR



Suggested Component Values

Freq. MHz	CV*	CF*	R1*	R2*	R3*	X1
14.31818	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*
16.0000	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*

Recommended Crystal Parameters

	14.31818 MHz**	16.0000 MHz**
R_S	10 Ω ± 2.0 Ω	10 Ω ± 2.0 Ω
CO	5.0 pF ± 1.5 pF	6.0 pF ± 1.0 pF
C1	0.0245 pF ± 15%	0.0319 pF ± 15%
L1	5.05 mH	3.1 mH
Q	50K ± 10K	40K ± 10K

Calibration Tolerance: 0.002% at 26°C
Temperature Tolerance: 0.001% 0°C to 70°C

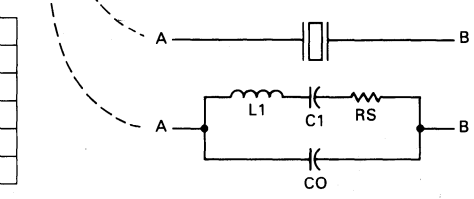
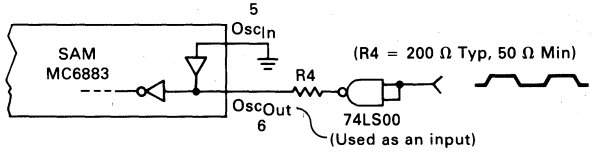


FIGURE 10 — TTL CLOCK INPUT



Typical input capacitances are 3.0 pF for Pin 5 and 5.5 pF for Pin 6.

*Optimum values depend on characteristics of the crystal (X1). For many applications, VClk must be 3.579545 MHz ± 50 Hz! Hence, Osc_{Out} must be made similarly "drift resistant" (by balancing temperature coefficients of X1, CV, CF, R1, R2 and R3).
**Specifically cut for MC6883 are International Crystal Manufacturing, Inc. Crystals (#167568 for 14.31818 MHz or #167569 for 16.0 MHz). However, other crystals may be used.

THEORY OF OPERATION

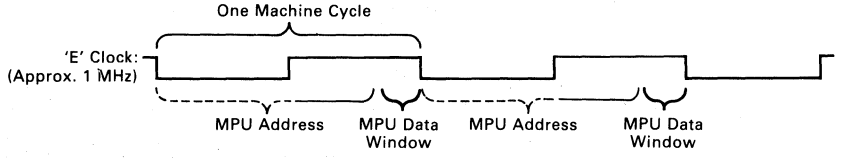
Video or No Video

Although the MC6883 may be used as a dynamic RAM controller **without** a video display*, most applications are likely to include a MC6847 video display generator (VDG). Therefore, this document emphasizes MC6883 with MC6847 systems.

Shared RAM (with interleaved DMA)

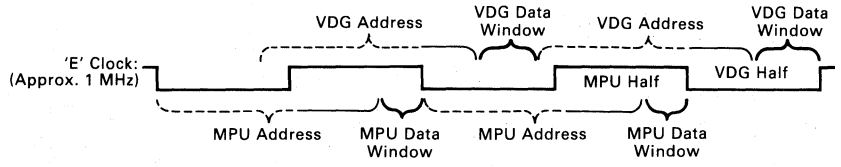
To minimize the number of RAM and interface chips, both the MPU and VDG share common dynamic RAM. Yet, the use of common RAM creates an apparent difficulty. That is, the MPU and VDG must both access the RAM without contention. This difficulty is overcome by taking advantage of the timing and architecture of Motorola MPU's (MC6800, MC6801E, MC6809E, MC68000). Specifically, **all** MPU accesses of external memory **always** occur in the **latter half** of the machine cycle, as shown below:

FIGURE 11 — MOTOROLA MPU TIMING



Similarly, the MC6847 (non-interlaced) VDG transfers a data byte in a half machine cycle (E or Φ2). Thus, when properly positioned, VDG and MPU RAM accesses interleave without contention as shown below:

FIGURE 12 — MOTOROLA MPU WITH VDG TIMING



This Interleaved Direct Memory Access (IDMA) is synchronized via the MC6883 by centering the VDG data window half-way between MPU data windows.**

The result is a shared RAM system without MPU/VDG RAM access contention, with both MPU and VDG running uninterrupted at normal operating speed, each transparent to the other.

RAM Refresh

Dynamic RAM refresh is accomplished by accessing eight*** sequential addresses every 64*** microseconds until 128 consecutive addresses have been accessed. To avoid RAM access contention between REFRESH and MPU, each of the 128 refresh accesses occupies the "VDG half" of the interleaved DMA (IDMA). Furthermore, refresh accesses occur only during the television retrace period (at which time the VDG doesn't need to access RAM).

In summary, the VDG, MPU and MC6883's Refresh Counter all transparently access the common dynamic RAM without contention or interruption.

Why IDMA?

Use of the interleaved direct memory access results in fast modification to variable portions of display RAM, by the MPU, without any distracting flashes on the screen (due to RAM access contention.) In addition, the MPU is not slowed down nor stopped by the MC6883; thereby, assuring accurate software timing loops without costly additional hardware timers. Furthermore, additional hardware and software to give "access permission" to the MPU is eliminated since the MPU may access RAM at **any** time.

* Only 1 pin, (DA0) out of 40 pins is dedicated to the video display.
** See VDG synchronization (page 10) for more detail.
*** When not using a MC6847, HS may be wired low for continuous transparent refresh.

"Systems On Silicon" Concept

Total Timing

For most applications, the SAM can supply complete system timing from its on-chip precision 14.31818 MHz oscillator. This includes buffered MPU clocks (E and Q), VDG clock, color subcarrier (3.58 MHz), row address select ($\overline{RAS}$), column address select ($\overline{CAS}$) and write enable ($\overline{WE}$).

Total Address Decode

For most applications, the SAM plus a "1 of 8 decoder" chip completely decodes I/O, ROM and RAM chip selects without wasting memory address space and without needlessly chopping-up contiguous address space. Chip selects are positioned in address space to allow three types of memory (RAM, local ROM and cartridge ROM) independent room for growth. For example, RAM may grow from address $0000-up, cartridge ROM may grow from address $FEFF-down and local ROM may grow from $FBFF-down. Alternately, if the application requires minimum ROM and maximum contiguous RAM, a second choice of two memory maps places RAM from $0000 to $FEFF. (See pages 17 and 18.)

In both memory maps all I/O, MPU vectors, SAM control registers, and some reserved address spaces are efficiently contained between addresses $FF00 and $FFFF.

How Much RAM?

Using nine SAM pins (Z0 – Z7 and $\overline{RAS0}$) the following combinations require no additional address logic.

FIGURE 13 — RAM CONFIGURATIONS

Address: MSB LSB	Chip Select:	
Z5Z4Z3Z2Z1Z0	$\overline{RAS0}$	One or two banks of 4K x 8 (like MCM4027's)
Z5Z4Z3Z2Z1Z0	$\overline{RAS1}$ (=Z7)	
Z6Z5Z4Z3Z2Z1Z0	$\overline{RAS0}$	One or two banks of 16K x 8 (like MCM4116's)
Z6Z5Z4Z3Z2Z1Z0	$\overline{RAS1}$ (=Z7)	
Z7Z6Z5Z4Z3Z2Z1Z0	$\overline{RAS0}$	One bank of 64K x 8 (like MCM6665's)

PROGRAMMING GUIDE

SAM — Programmability

The SAM contains a 16-bit control register which allows the MC6809E to program the SAM for the following options:

VDG Addressing Mode	3-bits
VDG Address Offset	7-bits
32K Page Switch	1-bit
MPU Rate	2-bits
Memory Size	2-bits
Map Type	1-bit

Note that when the SAM is **reset** by first applying power or by manual hardware reset,† all control register bits are **cleared** (to a logic "0").

VDG Addressing Mode

Three bits (V2, V1, V0) control the sequence of DISPLAY ADDRESSES generated by the SAM (which are used to scan dynamic RAM for video information). For example, if you wish to display Dynamic RAM data as INTERNAL ALPHANUMERICS VIDEO, you should program‡ the MC6847 for the INTERNAL ALPHANUMERICS MODE **and** CLEAR BITS V2, V1 and V0 in the SAM. The table on the following page summarizes the available modes:

† See Figure 7 for manual reset circuit.

‡ Typically, part of a PIA (MC6821) at location $FF22 is used to control MC6847 modes. (See MC6847 Data Sheet.)

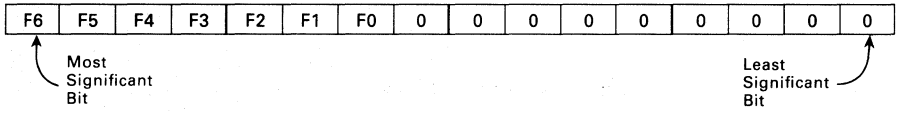
Mode Type	MC6847 Mode					SAM Mode		
	G/$\overline{A}$	GM2	GM1	GM0 EXT/$\overline{I}$	CSS	V2	V1	V0
Internal Alphanumerics	0	X	X	0	X	0	0	0
External Alphanumerics	0	X	X	1	X	0	0	0
OSemigraphics — 4	0	X	X	0	X	0	0	0
Semigraphics — 6	0	X	X	1	X	0	0	0
Semigraphics — 8*	0	X	X	0	X	0	1	0
Semigraphics — 12*	0	X	X	0	X	1	0	0
Semigraphics — 24*	0	X	X	0	X	1	1	0
Full Graphics — 1C	1	0	0	0	X	0	0	1
Full Graphics — 1R	1	0	0	1	X	0	0	1
Full Graphics — 2C	1	0	1	0	X	0	1	0
Full Graphics — 2R	1	0	1	1	X	0	1	1
Full Graphics — 3C	1	1	0	0	X	1	0	0
Full Graphics — 3R	1	1	0	1	X	1	0	1
Full Graphics — 6C	1	1	1	0	X	1	1	0
Full Graphics — 6R	1	1	1	1	X	1	1	0
Direct Memory Access†	X	X	X	X	X	1	1	1

*S8, S12, & S24 modes are **not** described in the MC6847 Data Sheet. See appendix "A".
†DMA is identical to 6R except as shown in Figure 5 on page 9.

3

VDG Address Offset

Seven bits (F6, F5, F4, F3, F2, F1 and F0) determine the **Starting Address** for the video display. The "Starting Address" is defined as "the address corresponding to data displayed in the **Upper Left** corner of the TV screen". The "Starting Address" is shown below in binary:



Note that the "Starting Address" may be placed anywhere within the 64K address space with a resolution of ½K (the size of one alphanumeric page).
The F6–F0 bits take effect during the TV vertical synchronization pulse (i.e., when $\overline{FS}$ from MC6847 is low).

Page Switch

One bit (P1) is used "in place of" A15 from the MC6809E in order to refer access within $0000-$7FFF to one of two 32K byte **pages** of RAM. If the system does **not** use more than 32K bytes of RAM, P1 can be ignored.**

**When using 4K x 1 RAMS, two banks of eight IC's are allowed. This accounts for Addresses $0000-1FFF. Also, this same RAM can be addressed at $2000-$3FFF, $4000-$5FFF and $6000-$7FFF.

MPU Rate

Two bits (R1, R0) control the clock rate to the MC6809E MPU. The options are:

RATE (FREQUENCY OF "E" CLOCK)	R1	R0
0.9 MHz (Crystal Frequency ÷ 16) Slow	0	0
0.9/1.8 MHz (Address Dependent Rate)	0	1
1.8 MHz (Crystal Frequency ÷ 8) Fast	1	X
(Typical Crystal Frequency = 14.31818 MHz)		

In the "address dependent rate" mode, accesses to $0000-$7FFF and $FF00-$FF1F are slowed to 0.9 MHz (crystal frequency ÷ 16) and all other addresses are accessed at 1.8 MHz (crystal frequency ÷ 8.)

Note: "Slow" (0.9 MHz) operation can be accomplished using 1.0 MHz MC6809E and MC6821 devices. For "Fast" (1.8 MHz) operation, 2.0 MHz MC68B09E and MC68B21 devices must be used.

Memory Size

Two bits (M1 and M0) determine RAM memory size. The options are:

SIZE	M1	M0
One or two banks of 4K × 1 dynamic RAMs	0	0
One or two banks of 16K × 1 dynamic RAMs	0	1
One bank of 64K × 1 dynamic RAMs	1	0
Up to 64K static RAM*	1	1

*Requires a latch for demultiplexing the RAM address.

IMPORTANT!

Note: Be sure to program the SAM for the correct memory size **before** using RAM (i.e., for a subroutine stack).

Map Type

One bit (TY) is used to select between two memory map configurations.

Refer to pages 17, 18 and 19 for details. Early versions of the SAM did not allow the "Fast" MPU rate to be used in conjunction with Map Type "TY = 1". Devices manufactured after January 1, 1983 allow both "Fast" and "Slow" MPU rates to be used with Map Type "TY = 1." (Date of manufacture is marked on devices as YYWW where YY is the year and WW is the week of manufacture.)

Writing To The SAM Control Register

Any bit in the control register (CR) may be set by writing to a specific unique address. Each bit has two unique addresses . . . writing to the **even #** address **clears** the bit and writing to the **odd #** address **sets** the bit. (Data on the data bus is irrelevant in this procedure.) The specific addresses are tabulated on pages 17 and 18.

If desired, a short routine may be written to program the SAM CR "a word at a time". For example, the following routine copies "B" bits from "A" register to SAM CR addresses beginning with address "X".

```
SAM1    46          ROR   A
        24  06      BCC   SAM2
        30  01      INX   (LEAX1,X)
        A7  80      STA   O,X+
        20  02      BRA   SAM3
SAM2    A7  81      STA   O,X++
SAM3    5A          DEC   B
        26  F2      BNE   SAM1
        39          RTS
```

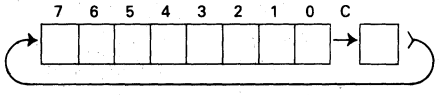
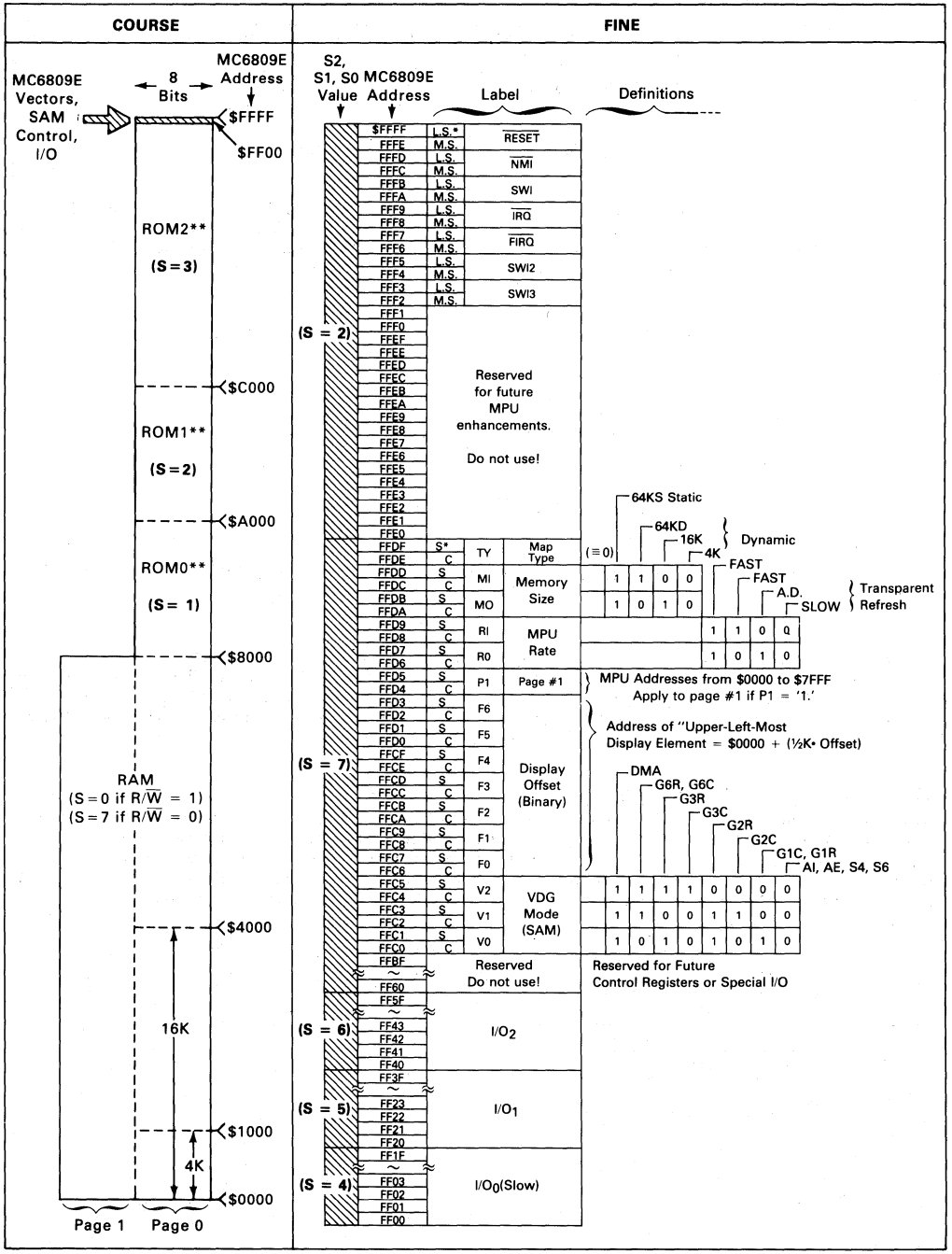


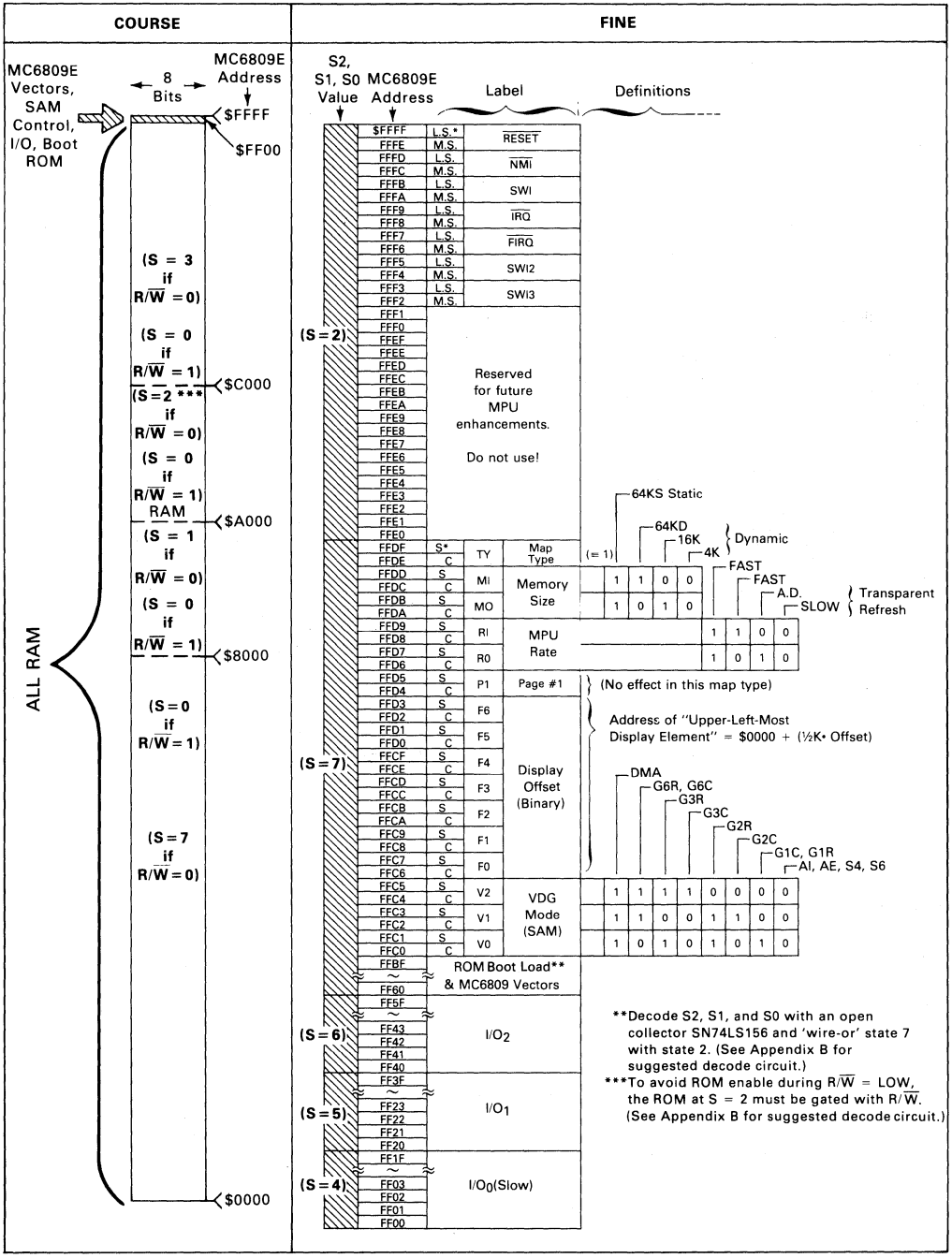
FIGURE 14 — MEMORY MAP (TYPE #0)



*Note:
M.S. = Most Significant
L.S. = Least Significant
S = Set Bit
C = Clear Bit
(All bits are cleared when SAM is reset.)
S = Device Select value = 4 x S2 + 2 x S1 + 1 x S0

**May also be RAM

FIGURE 15 — MEMORY MAP (TYPE #1)



*Note:
M.S. = Most Significant
L.S. = Least Significant
S = Set Bit
C = Clear Bit
(All bits are cleared when SAM is reset.)
S = Device Select value = 4 x S2 + 2 x S1 + 1 x S0

FIGURE 16 — MEMORY ALLOCATION TABLE
(Also, see the memory MAPs on pages 17 and 18.)

Type # 0: (Primarily for ROM based systems)

Address Range	S=4(S2)+2 (S1)+S0 S Value	Intended Use
$FFF2 to FFFF	2	MC6809E Vectors: $\overline{\text{Reset}}$, $\overline{\text{NMI}}$, SWI, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, SWI2, SWI3.
FFE0 to FFF1	2	Reserved for future MPU enhancements.
FFC0 to FFDF	7	SAM Control Register: V0, – V2, F0 – F6, P, R0, R1, M0, M1, TY.
FF60 to FFBF	7	Reserved for future control register enhancements.
FF40 to FF5F	6	I/O_2: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF20 to FF3F	5	I/O_1: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF00 to FF1F	4	I/O_0: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
C000 to FEFF	3	ROM2: 16K addresses. External cartridge ROM*.
A000 to BFFF	2	ROM1: 8K addresses. Internal ROM*. Note that MC6809E vector addresses select this ROM*.
8000 to 9FFF	1	ROM0: 8K addresses. Internal ROM*.
0000 to 7FFF	0 if R/$\overline{\text{W}}$ = 1 7 if R/$\overline{\text{W}}$ = 0	RAM: 32K addresses. RAM shared by MPU and VDG.

*Not restricted to ROM. For example, RAM or I/O may be used here.

Type # 1: (Primarily for RAM based systems)

Address Range	S=4(S2)+2 (S1)+S0 S Value	Intended Use
$FFF2 to FFFF	2	MC6809E Vectors: $\overline{\text{Reset}}$, $\overline{\text{NMI}}$, SWI, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, SWI2, SWI3.
FFE0 to FFF1	2	Reserved for future MPU enhancements.
FFC0 to FFDF	7	SAM Control Register: V0 – V2, F0 – F_6, P, R0, R1, M0, M1, TY.
FF60 to FFBF	7	Small ROM: Boot load program and initial MC6809 vectors.
FF40 to FF5F	6	I/O_2: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0-A4.
FF20 to FF3F	5	I/O_1: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF00 to FF1F	4	I/O_0: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A2 – A4.
0000 to FEFF	0 if R/$\overline{\text{W}}$ = 1	RAM: 64K(–256) addresses, shared by MPU and VDG. (If R/$\overline{\text{W}}$ = 0 then S = 3 for $C000–$FEFF; S = 2 for $A000–$BFFF; S = 1 for $8000–$9FFF and S = 7 for $0000–$7FFF.)

APPENDIX A

VDG/SAM Video Display System Offers 3 New Modes

by

Paul Fletcher

There are three new modes created when the VDG and SAM are used together in a video display system. These modes offer alphanumeric compatibility with 8 color low-to-high resolution graphics, 64Hx64V, 64Hx96V, 64Hx192V. The new modes S8, S12, and S24 are created by placing the VDG in the Alpha Internal mode and having the SAM in a 2K, 3K or 6K full color graphics mode. In all modes the VDG's S/$\overline{A}$ and Inv. pins are connected to data bits DD7 and DD6 to allow switching on the fly between Alpha and Semigraphics and between inverted and non-inverted alpha. This method is used in most VDG systems to obtain maximum flexibility.

The three modes divide the standard 8*12 dot box used by the VDG for the standard alpha and semigraphics modes into eight 4*3 dot boxes for the S8 mode, twelve 4*2 dot boxes for the S12 mode, and twenty-four 4*1 dot boxes for the S24 mode. Figure 17 shows the arrangement of these boxes. One byte is needed to control two horizontally consecutive boxes. It therefore takes four bytes for the S8, six bytes for the S12, and 12 bytes for the S24 mode to control the entire 8*12 dot box. These two horizontally consecutive boxes have four combinations of luminance controlled by bits B0 – B3. For convenience B2 should be made equal to B0 and B3 should be made equal to B1. This eliminates a screen placement problem which would cause other codes to change patterns when moved vertically on the screen. The illuminated boxes can be one of eight colors which are controlled by B4 – B6 (see Figure 18). The bytes needed to control all the boxes in the 8*12 dot box must be spaced 32 address spaces apart in the display RAM because of the addressing scheme orginally used in the VDG and duplicated by the SAM. This means to place an alphanumeric character on the TV screen it requires 4, 6, or 12 bytes depending on the mode used. These bytes are placed 32 memory locations apart in the display RAM (see Figure 18). This multiple byte format allows the mixing of character rows of different characters in the same 8*12 dot box creating new characters and symbols. It also allows overlining and underlining in eight colors by switching to semigraphics at the correct time.

These new modes optimize the memory versus screen density tradeoffs for RF performance on color TVs. This could make them the most versatile of all the modes depending on the users creativity and the software sophistication.

APPENDIX B

Memory Decode for "MAP TYPE = 1"

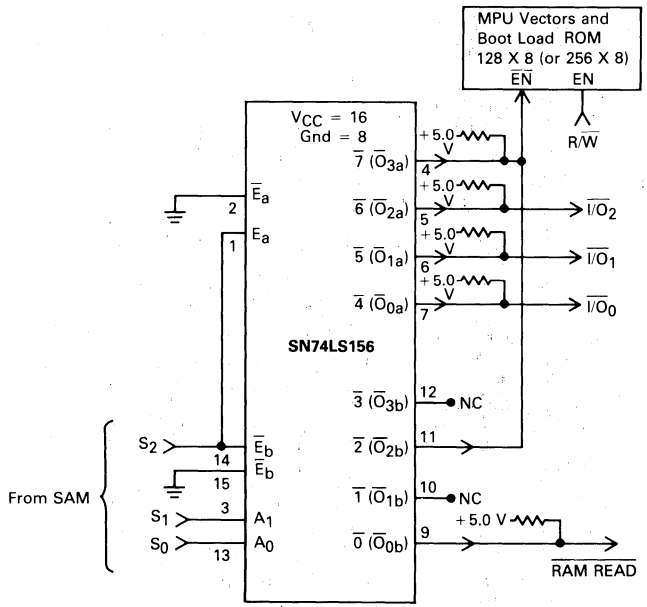
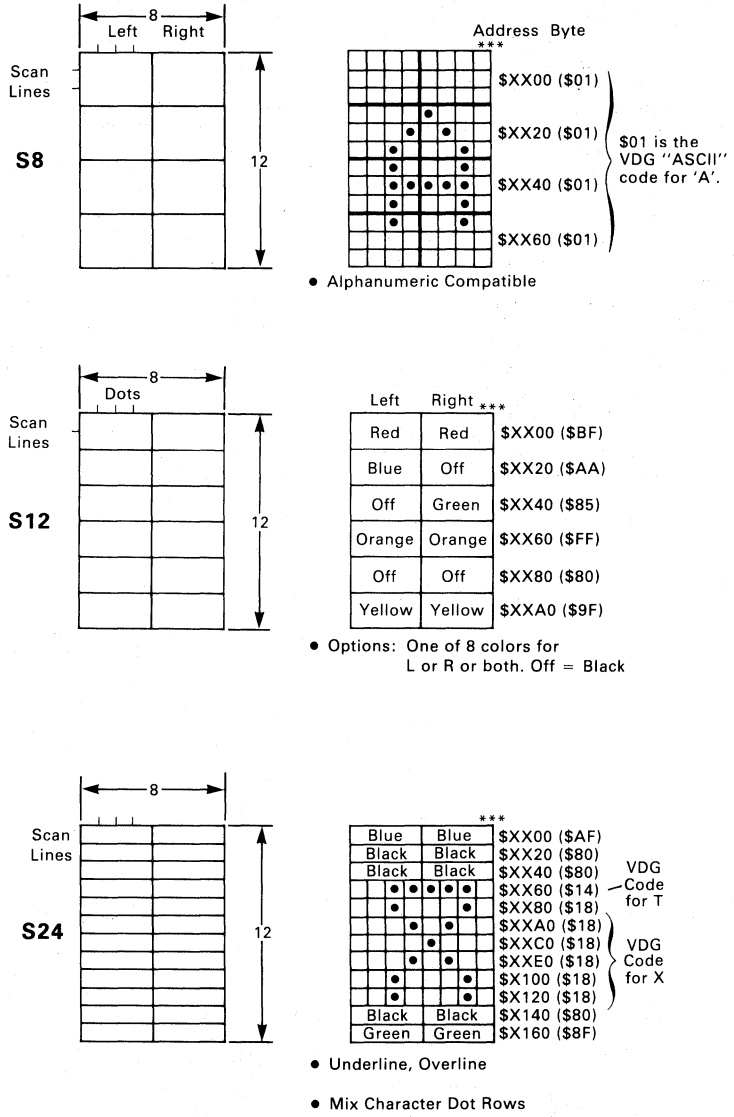
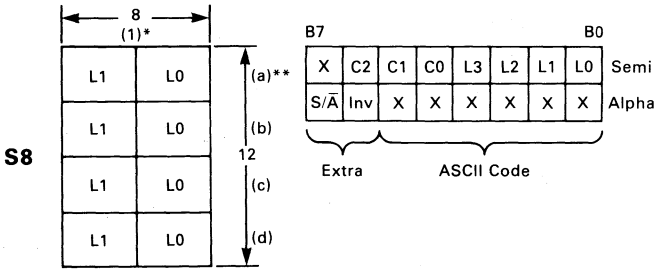


FIGURE 17 — DISPLAY MODES S8, S12, S24
Bit/Visible Dot Correlation



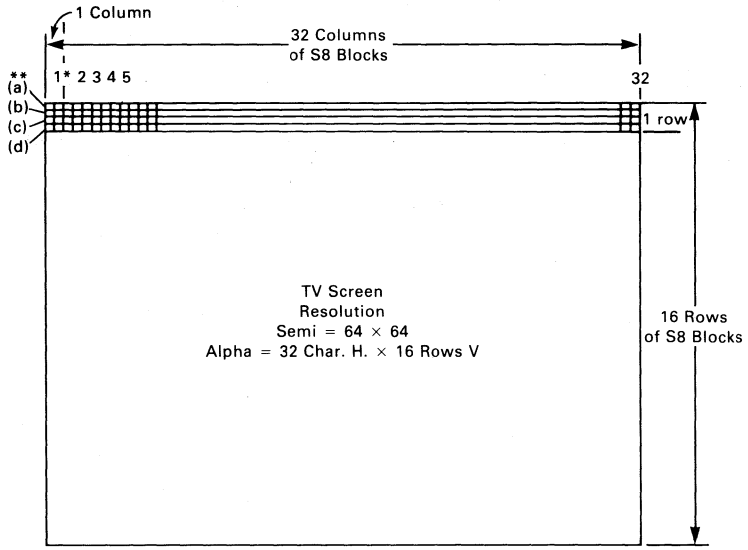
*** Characters will always remain in standard VDG positions.

FIGURE 18 — S8 DISPLAY FORMAT EXAMPLES



LX	C2	C1	C0	Color
0	X	X	X	Black
1	0	0	0	Green
1	0	0	1	Yellow
1	0	1	0	Blue
1	0	1	1	Red
1	1	0	0	Buff
1	1	0	1	Cyan
1	1	1	0	Magenta
1	1	1	1	Orange

B3,B1	B2,B0		4	
0	0	=	Off	Off
0	1	=	Off	Color
1	0	=	Color	Off
1	1	=	Color	



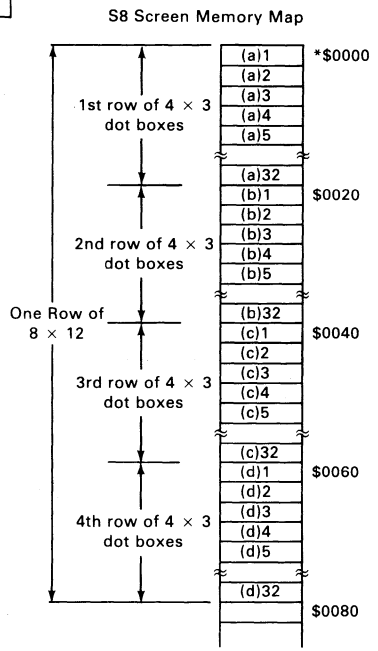
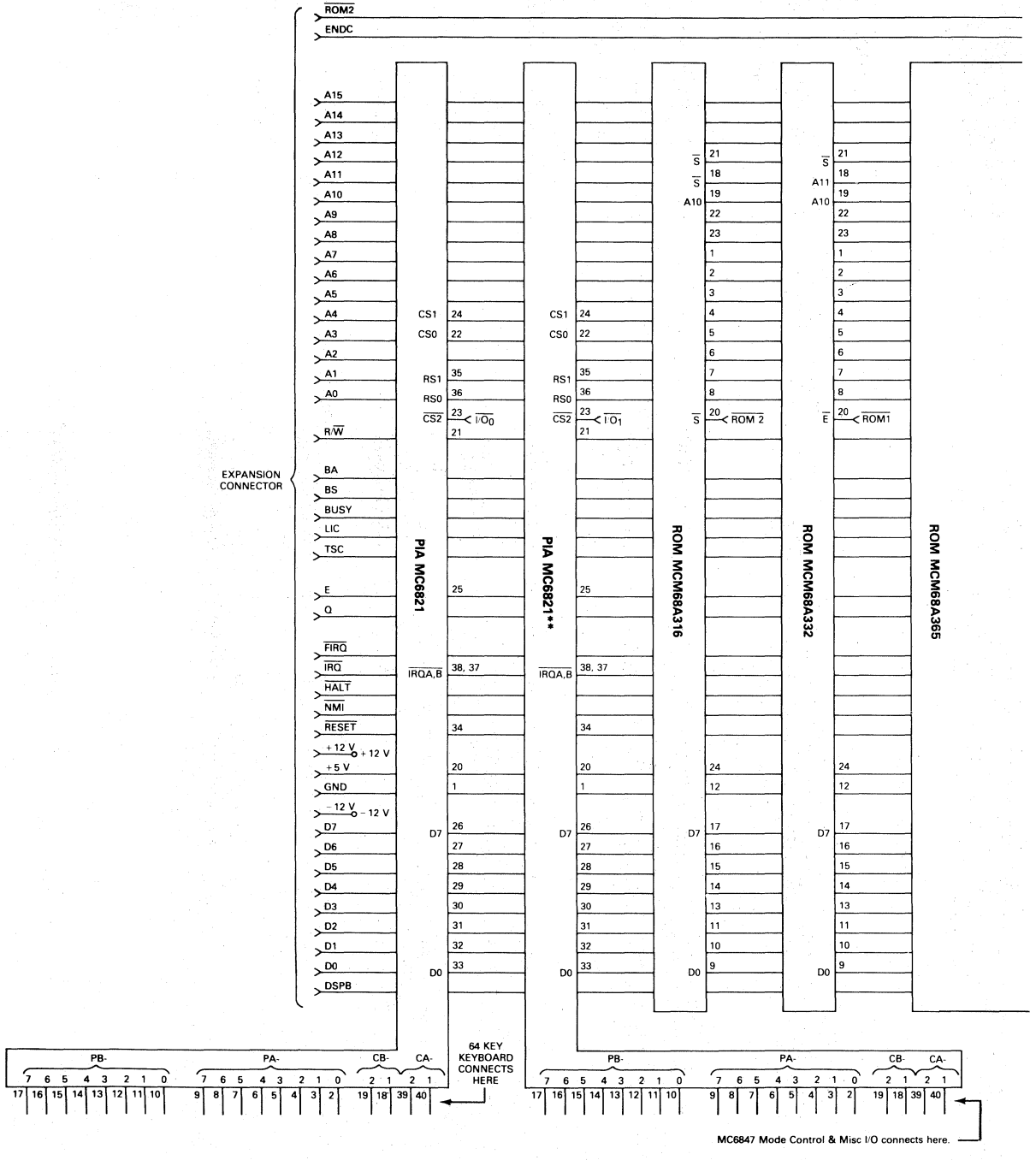
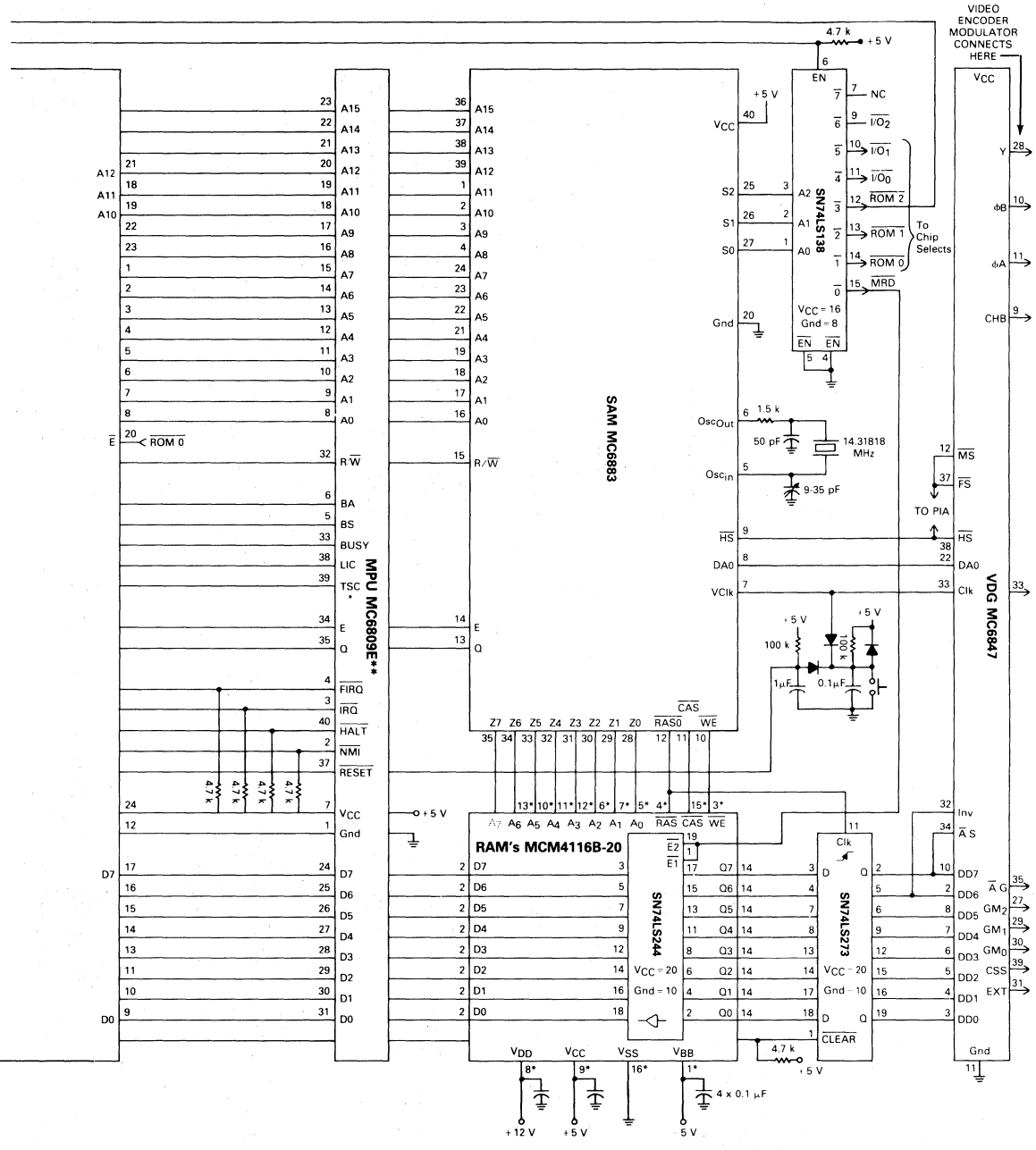


FIGURE 19 — EXAMPLE of MC6809E, MC6883 and MC6847 COMPUTER





*This pin number on 8 different RAM chips is connected to this point.
**See text . . . page 16

FIGURE 20 — EQUIVALENT OF OSCILLATOR INPUT AND OUTPUT

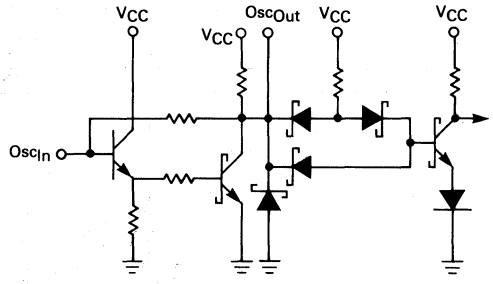


FIGURE 21 — DA0 INPUT

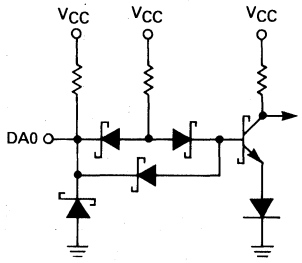


FIGURE 22 — VClk INPUT/OUTPUT

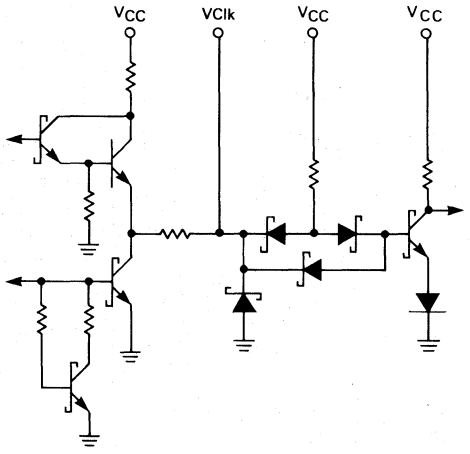


FIGURE 23 — E AND Q OUTPUTS

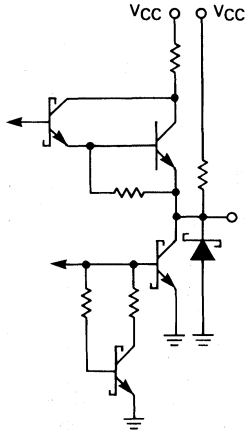


FIGURE 24 — TYPICAL INPUT

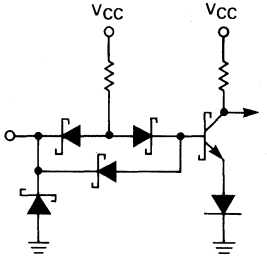
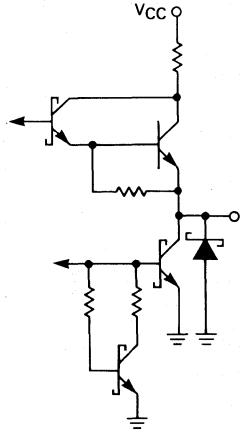


FIGURE 25 — TYPICAL OUTPUT



MOTOROLA

MC8T95/MC6885
MC8T96/MC6886
MC8T97/MC6887
MC8T98/MC6888

HEX THREE-STATE BUFFER INVERTERS

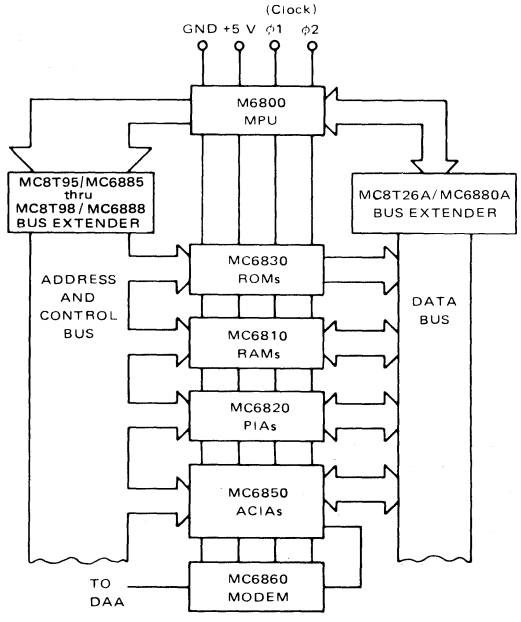
This series of devices combines three features usually found desirable in bus-oriented systems: 1) High impedance logic inputs insure that these devices do not seriously load the bus; 2) Three-state logic configuration allows buffers not being utilized to be effectively removed from the bus; 3) Schottky technology allows high-speed operation.

The devices differ in that the non-inverting MC8T95/MC6885 and inverting MC8T96/MC6886 provide a two-input Enable which controls all six buffers, while the non-inverting MC8T97/MC6887 and inverting MC8T98/MC6888 provide two Enable inputs – one controlling four buffers and the other controlling the remaining two buffers.

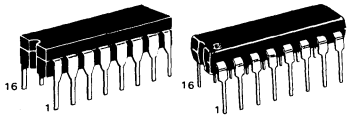
The units are well-suited for Address buffers on the M6800 or similar microprocessor application.

- High Speed – 8.0 ns (Typ)
- Three-State Logic Configuration
- Single +5 V Power Supply Requirement
- Compatible with 74LS Logic or M6800 Microprocessor Systems
- High Impedance PNP Inputs Assure Minimal Loading of the Bus

MICROPROCESSOR BUS EXTENDER APPLICATION

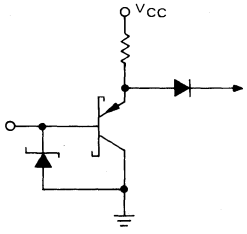


HEX THREE-STATE BUFFER/INVERTERS

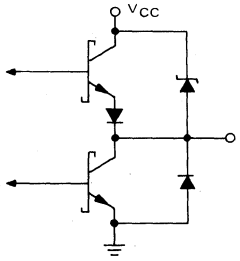


CASE 620 CASE 648

INPUT EQUIVALENT CIRCUIT



OUTPUT EQUIVALENT CIRCUIT



ORDERING INFORMATION

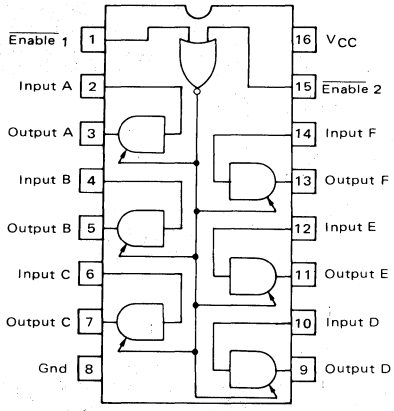
(Temperature Range for the following devices = 0 to +75°C)

DEVICE	ALTERNATE	PACKAGE
MC8T95L	MC6885L	Ceramic DIP
MC8T96L	MC6886L	Ceramic DIP
MC8T97L	MC6887L	Ceramic DIP
MC8T98L	MC6888L	Ceramic DIP
MC8T95P	MC6885P	Plastic DIP
MC8T96P	MC6886P	Plastic DIP
MC8T97P	MC6887P	Plastic DIP
MC8T98P	MC6888P	Plastic DIP

3

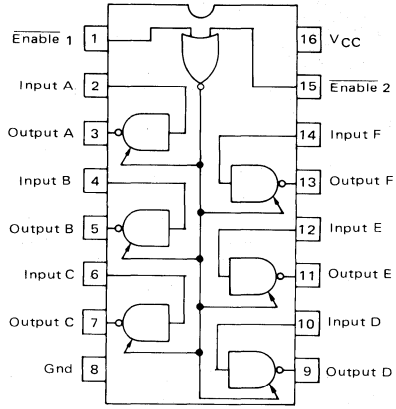
PIN CONNECTIONS AND TRUTH TABLES

MC8T95/MC6885



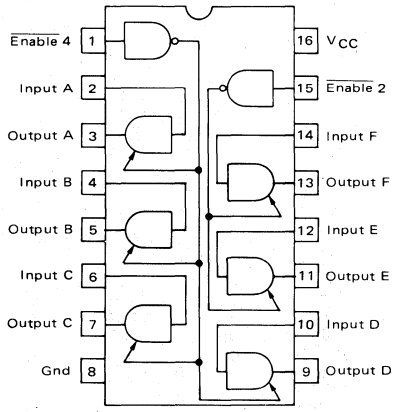
Enable 2	Enable 1	Input	Output
L	L	L	L
L	L	H	H
L	H	X	Z
H	L	X	Z
H	H	X	Z

MC8T96/MC6886



Enable 2	Enable 1	Input	Output
L	L	L	H
L	L	H	L
L	H	X	Z
H	L	X	Z
H	H	X	Z

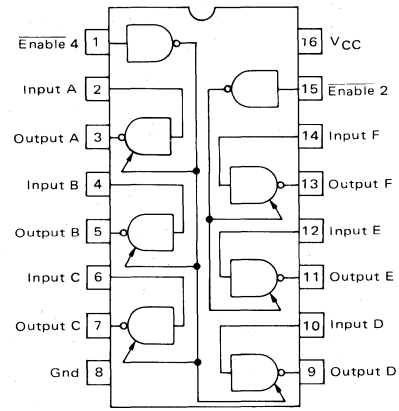
MC8T97/MC6887



Enable	Input	Output
L	L	L
L	H	H
H	X	Z

L = Low Logic State
H = High Logic State
Z = Third (High Impedance) State
X = Irrelevant

MC8T98/MC6888



Enable	Input	Output
L	L	H
L	H	L
H	X	Z

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Operating Junction Temperature Plastic Package Ceramic Package	T_J	 150 175	°C

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, $0^{o}C \leq T_A \leq 75^{o}C$ and $4.75\ V \leq V_{CC} \leq 5.25\ V$)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Voltage – High Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IH}	2.0	–	–	V
Input Voltage – Low Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IL}	–	–	0.8	V
Input Current – High Logic State (V_{CC} = 5.25 V, V_{IH} = 2.4 V)	I_{IH}	–	–	40	μA
Input Current – Low Logic State (V_{CC} = 5.25 V, V_{IL} = 0.5 V, $V_{IL(\overline{E})}$ = 0.5 V)	I_{IL}	–	–	-400	μA
Input Current – High Impedance State (V_{CC} = 5.25 V, $V_{IL(I)}$ = 0.5 V, $V_{IH(\overline{E})}$ = 2.0 V)	$I_{IH(\overline{E})}$	–	–	-40	μA
Output Voltage – High Logic State (V_{CC} = 4.75 V, I_{OH} = -5.2 mA)	V_{OH}	2.4	–	–	V
Output Voltage – Low Logic State (I_{OL} = 48 mA)	V_{OL}	–	–	0.5	V
Output Current – High Impedance State (V_{CC} = 5.25 V, V_{OH} = 2.4 V)	I_{OZ}	–	–	40	μA
(V_{CC} = 5.25 V, V_{OL} = 0.5 V)		–	–	-40	
Output Short-Circuit Current (V_{CC} = 5.25 V, V_O = 0) (only one output can be shorted at a time)	I_{OS}	-40	-80	-115	mA
Power Supply Current	I_{CC}				mA
(V_{CC} = 5.25 V) MC8T95, MC8T97, MC6885, MC6887		–	65	98	
MC8T96, MC8T98, MC6886, MC6888		–	59	89	
Input Clamp Voltage (V_{CC} = 4.75 V, I_{IC} = -12 mA)	V_{IC}	–	–	-1.5	V
Output V_{CC} Clamp Voltage (V_{CC} = 0, I_{OC} = 12 mA)	V_{OC}	–	–	1.5	V
Output Gnd Clamp Voltage (V_{CC} = 0, I_{OC} = -12 mA)	V_{OC}	–	–	-1.5	V
Input Voltage (I_I = 1.0 mA)	V_I	5.5	–	–	V

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted.)

		MC8T95/97 MC6885/87			MC8T96/98 MC6886/88			
Characteristic	Symbol	Min	Typ	Max	Min	Typ	Max	Unit
Propagation Delay Time – High to Low State	t_{PHL}							ns
(C_L = 50 pF)		3.0	–	12	4.0	–	11	
(C_L = 250 pF)		–	16	–	–	15	–	
(C_L = 375 pF)		–	20	–	–	18	–	
(C_L = 500 pF)		–	23	–	–	22	–	
Propagation Delay Time – Low to High State	t_{PLH}							ns
(C_L = 50 pF)		3.0	–	13	3.0	–	10	
(C_L = 250 pF)		–	25	–	–	22	–	
(C_L = 375 pF)		–	33	–	–	28	–	
(C_L = 500 pF)		–	42	–	–	35	–	
Transition Time – High to Low State	t_{THL}							ns
(C_L = 250 pF)		–	10	–	–	10	–	
(C_L = 375 pF)		–	11	–	–	13	–	
(C_L = 500 pF)		–	14	–	–	15	–	
Transition Time – Low to High State	t_{TLH}							ns
(C_L = 250 pF)		–	32	–	–	28	–	
(C_L = 375 pF)		–	42	–	–	38	–	
(C_L = 500 pF)		–	60	–	–	53	–	

SWITCHING CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C unless otherwise noted.)

Characteristic	Symbol	MC8T95/97 MC6885/87 Min	Typ	Max	MC8T96/98 MC6886/88 Min	Typ	Max	Unit
Propagation Delay Time – High State to Third State (C_L = 5.0 pF)	$t_{PHZ(\overline{E})}$	–	–	10	–	–	10	ns
Propagation Delay Time – Low State to Third State (C_L = 5.0 pF)	$t_{PLZ(\overline{E})}$	–	–	12	–	–	16	ns
Propagation Delay Time – Third State to High State (C_L = 50 pF)	$t_{PZH(\overline{E})}$	–	–	25	–	–	22	ns
Propagation Delay Time – Third State to Low State (C_L = 50 pF)	$t_{PZL(\overline{E})}$	–	–	25	–	–	24	ns

FIGURE 1 – TEST CIRCUIT FOR SWITCHING CHARACTERISTICS

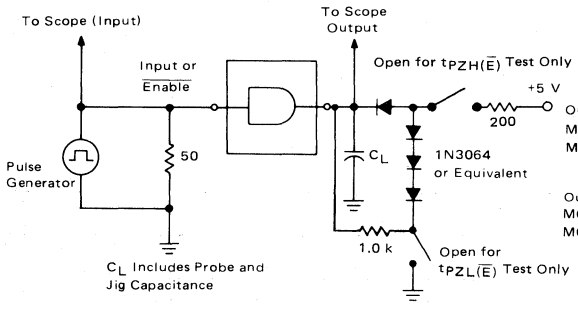


FIGURE 2 – WAVEFORMS FOR PROPAGATION DELAY TIMES INPUT TO OUTPUT

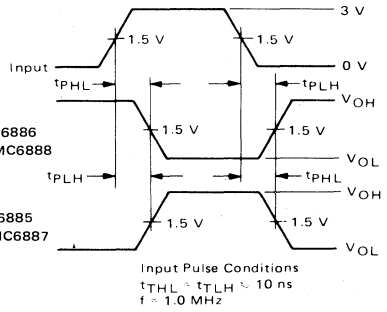
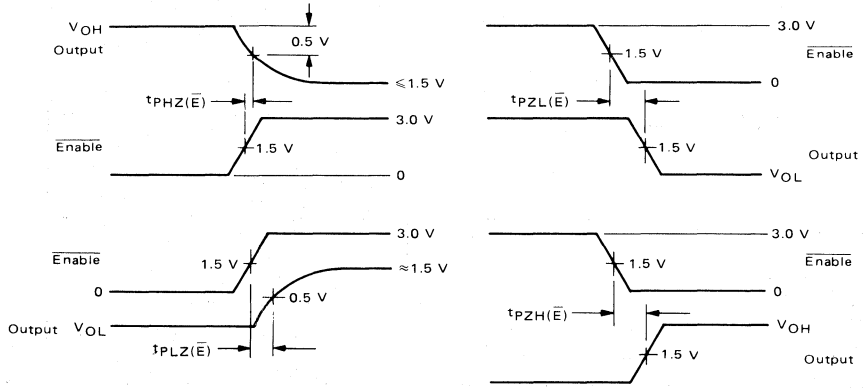
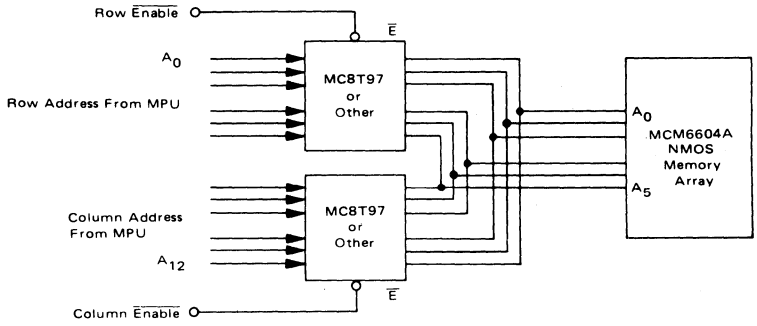


FIGURE 3 – WAVEFORMS FOR PROPAGATION DELAY TIMES – $\overline{\text{ENABLE}}$ TO OUTPUT



H = High-Logic State, L = Low-Logic State, Z = High Impedance State

FIGURE 4 – ADDRESS MULTIPLEXER FOR 16-PIN 4K NMOS MEMORY



MOTOROLA

MC8T28
MC6889

NONINVERTING
QUAD THREE-STATE BUS TRANSCEIVER

This quad three-state bus transceiver features both excellent MOS or MPU compatibility, due to its high impedance PNP transistor input, and high-speed operation made possible by the use of Schottky diode clamping. Both the -48 mA driver and -20 mA receiver outputs are short-circuit protected and employ three-state enabling inputs.

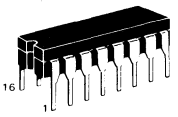
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier diode clamps to suppress excessive undershoot voltages.

Propagation delay times for the driver portion are 17 ns maximum while the receiver portion runs 17 ns. The MC8T28 is identical to the NE8T28 and it operates from a single +5 V supply.

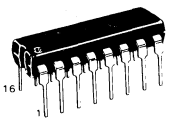
- High Impedance Inputs
- Single Power Supply
- High Speed Schottky Technology
- Three-State Drivers and Receivers
- Compatible with M6800 Family Microprocessor
- Non-Inverting

NONINVERTING
BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

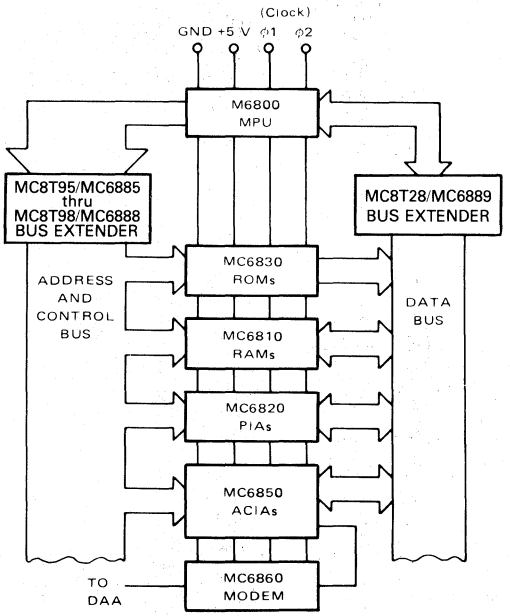


L SUFFIX
CERAMIC PACKAGE
CASE 620-02

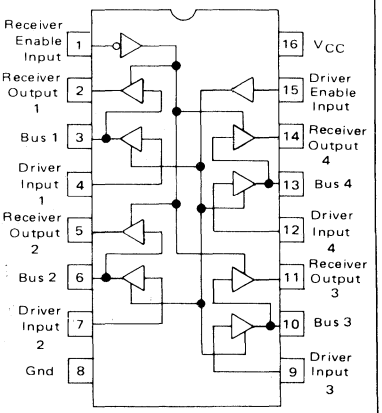


P SUFFIX
PLASTIC PACKAGE
CASE 648-05

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS — MC8T28 MC6889



ORDERING INFORMATION

Device	Alternate	Temperature Range	Package
MC8T28L	MC6889L	0 to +75°C	Ceramic DIP
MC8T28P	MC6889P	0 to +75°C	Plastic DIP

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Junction Temperature Ceramic Package Plastic Package	T_J	 175 150	°C
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (4.75 V ≤ V_{CC} ≤ 5.25 V and 0°C ≤ T_A ≤ 75°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current – Low Logic State					
(Receiver Enable Input, $V_{IL(\overline{RE})}$ = 0.4 V)	$I_{IL(\overline{RE})}$	-		-200	μA
(Driver Enable Input, $V_{IL(DE)}$ = 0.4 V)	$I_{IL(DE)}$			-200	
(Driver Input, $V_{IL(D)}$ = 0.4 V)	$I_{IL(D)}$			-200	
(Bus (Receiver) Input, $V_{IL(B)}$ = 0.4 V)	$I_{IL(B)}$			-200	
Input Disabled Current – Low Logic State					
(Driver Input, $V_{IL(D)}$ = 0.4 V)	$I_{IL(D)\ DIS}$	–	–	-25	μA
Input Current-High Logic State					
(Receiver Enable Input, $V_{IH(\overline{RE})}$ = 5.25 V)	$I_{IH(\overline{RE})}$			25	μA
(Driver Enable Input, $V_{IH(DE)}$ = 5.25 V)	$I_{IH(DE)}$			25	
(Driver Input, $V_{IH(D)}$ = 5.25 V)	$I_{IH(D)}$			25	
Input Voltage – Low Logic State					
(Receiver Enable Input)	$V_{IL(\overline{RE})}$			0.85	V
(Driver Enable Input	$V_{IL(DE)}$			0.85	
(Driver Input)	$V_{IL(D)}$			0.85	
(Receiver Input)	$V_{IL(B)}$		–	0.85	
Input Voltage – High Logic State					
(Receiver Enable Input)	$V_{IH(\overline{RE})}$	2.0			V
(Driver Enable Input)	$V_{IH(DE)}$	2.0			
(Driver Input)	$V_{IH(D)}$	2.0			
(Receiver Input)	$V_{IH(B)}$	2.0	–	–	
Output Voltage – Low Logic State					
(Bus Driver) Output, $I_{OL(B)}$ = 48 mA)	$V_{OL(B)}$	-	–	0.5	V
(Receiver Output, $I_{OL(R)}$ = 20 mA)	$V_{OL(R)}$	-	-	0.5	
Output Voltage – High Logic State					
(Bus (Driver) Output, $I_{OH(B)}$ = -10 mA)	$V_{OH(B)}$	2.4	3.1	-	V
(Receiver Output, $I_{OH(R)}$ = -2.0 mA)	$V_{OH(R)}$	2.4	3.1	-	
(Receiver Output, $I_{OH(R)}$ = -100/μA, V_{CC} = 5.0 V)		3.5	–	–	
Output Disabled Leakage Current – High Logic State					
(Bus Driver) Output, $V_{OH(B)}$ = 2.4 V)	$I_{OHL(B)}$	–	-	100	μA
(Receiver Output, $V_{OH(R)}$ = 2.4 V)	$I_{OHL(R)}$	–	-	100	
Output Disabled Leakage Current – Low Logic State					
(Bus Output, $V_{OL(B)}$ = 0.5 V)	$I_{OLL(B)}$	–	–	-100	μA
(Receiver Output, $V_{OL(R)}$ = 0.5 V)	$I_{OLL(R)}$	–	–	-100	
Input Clamp Voltage					
(Driver Enable Input $I_{ID(DE)}$ = -12 mA)	$V_{IC(DE)}$	–	–	-1.0	V
(Receiver Enable Input $I_{IC(RE)}$ = +12 mA)	$V_{IC(RE)}$	–	–	-1.0	
(Driver Input $I_{IC(D)}$ = -12 mA)	$V_{IC(D)}$	–	–	-1.0	
Output Short-Circuit Current, V_{CC} = 5.25 V (1)					
(Bus (Driver) Output)	$I_{OS(B)}$	-50	–	-150	mA
(Receiver Output)	$I_{OS(R)}$	-30	–	-75	
Power Supply Current	I_{CC}	–	–	110	mA
(V_{CC} = 5.25 V)					

(1) Only one output may be short-circuited at a time.

SWITCHING CHARACTERISTICS (Unless otherwise noted, V_{CC} = 5.0 V and T_A = 25°C)

Characteristic	Symbol	Min	Max	Unit
Propagation Delay Time—Receiver (C_L = 30 pF)	$t_{PLH(R)}$ $t_{PHL(R)}$	–	17 17	ns
Propagation Delay Time—Driver (C_L = 300 pF)	$t_{PLH(D)}$ $t_{PHL(D)}$	–	17 17	ns
Propagation Delay Time—Enable (C_L = 30 pF) – Receiver	$t_{PZL(R)}$ $t_{PLZ(R)}$	– –	23 18	ns
– Driver Enable (C_L 300 pF)	$t_{PZL(D)}$ $t_{PLZ(D)}$	– –	28 23	

3

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

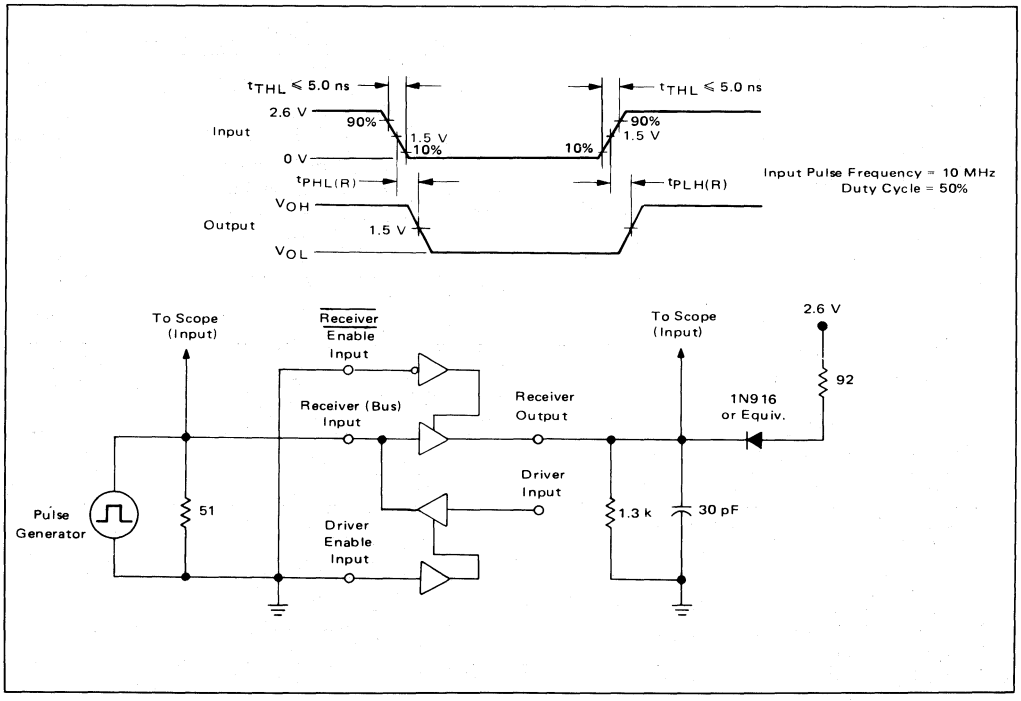


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

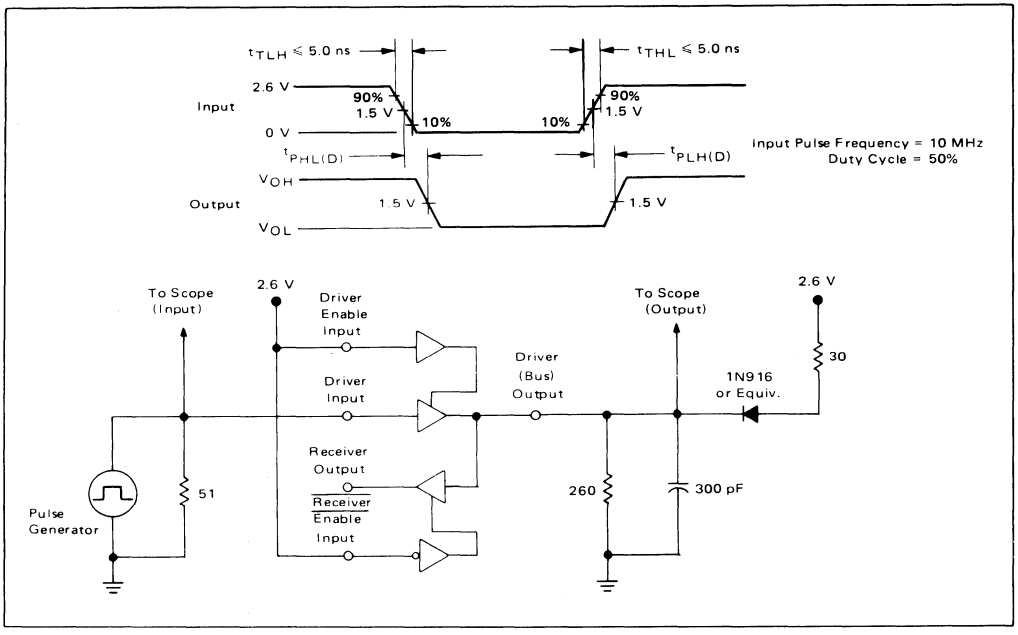


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

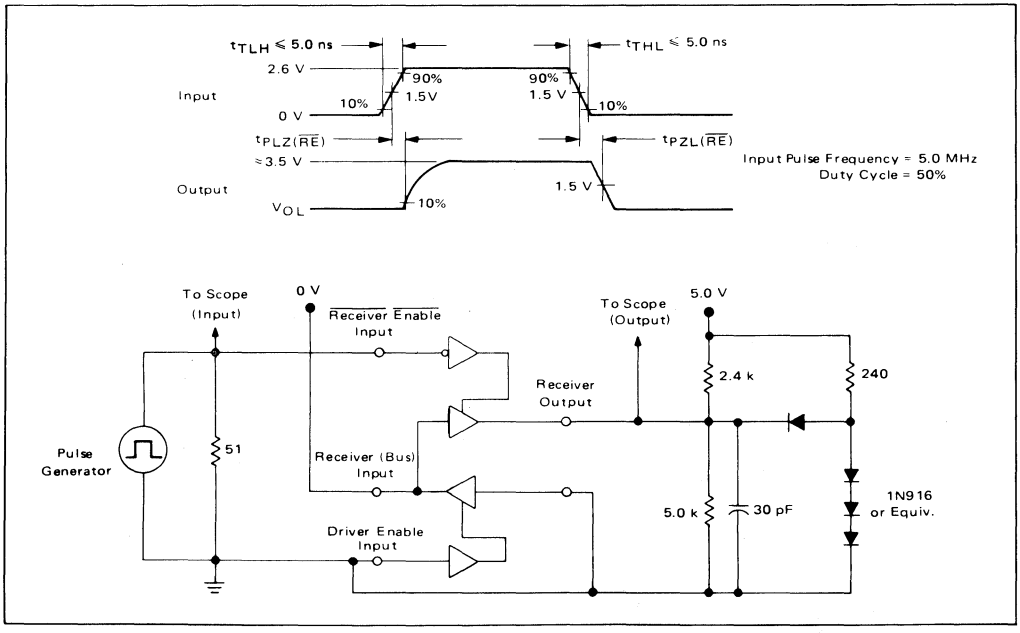


FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

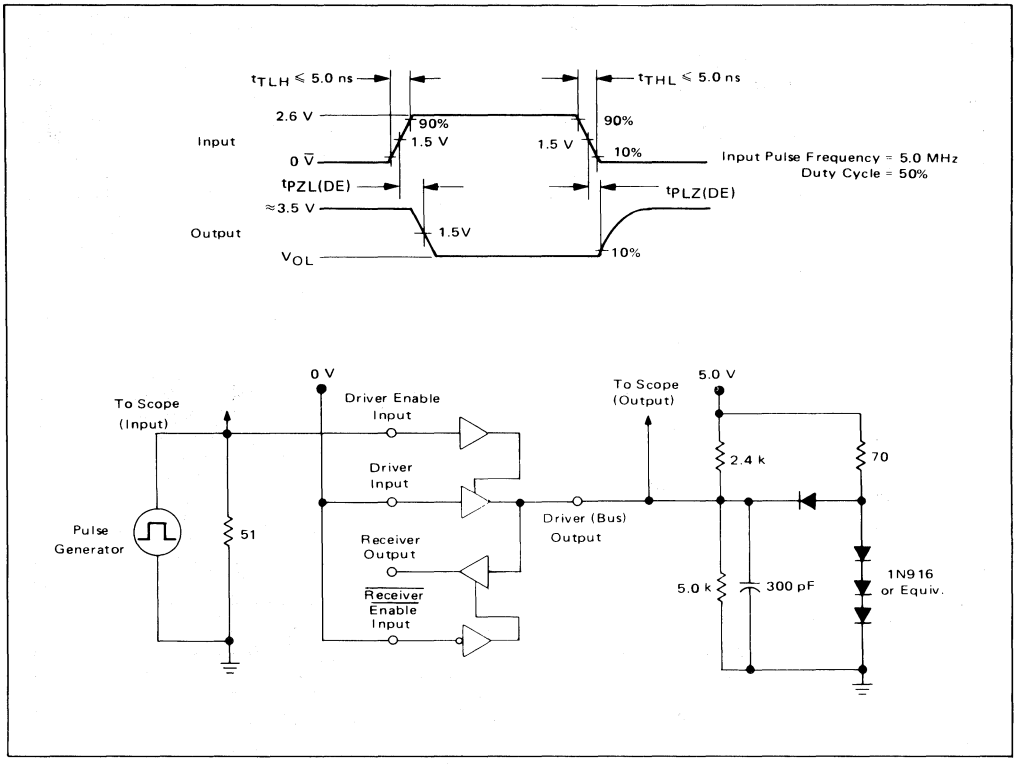
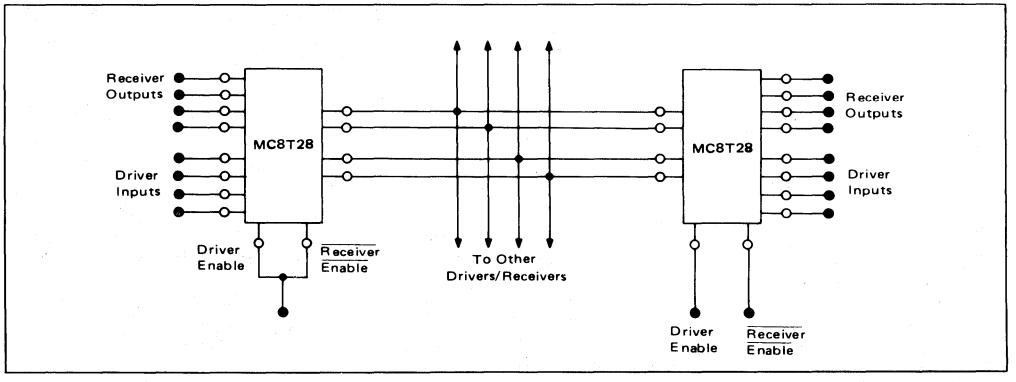


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS



MOTOROLA

MC6890

Advance Information

MPU-BUS-COMPATIBLE 8-BIT D-TO-A CONVERTER

The MC6890 is a self-contained, bus-compatible, 8 bit (±0.19% accuracy) D-to-A converter system capable of interfacing directly with 8-bit microprocessors.

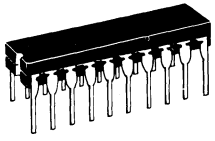
Available in both commercial and military temperature ranges, this monolithic converter contains master/slave registers to prevent transparency to data transitions during active enable; a laser-trimmed, low-TC, 2.5 V precision bandgap reference; and high stability, laser-trimmed, thin-film resistors for both reference input and output span and bipolar offset control.

A reset pin provides for overriding stored data and forcing I_{out} to zero.

- Direct Data Bus Link with All Popular TTL Level MPU's
- ±1/2 LSB Nonlinearity Over Temperature
- Fast Settling Time: 200 ns Typ
- Internal 2.5-V Precision Laser-Trimmed Voltage Reference (May Also Be Used Externally)
- Minimum $\overline{\text{Enable}}$ Pulse Width: 70 ns
- Fast $\overline{\text{Enable}}$: 10 ns Maximum Data Hold Time
- $\overline{\text{Reset}}$ Pin to Override Data
- Output Voltage Ranges: +5, +10, +20, or ±2.5, ±5, ±10 Volts
- Low Power: 90 mW Typ
- +5 V and -5 V to -15 V Supplies

8-BIT MPU-BUS-COMPATIBLE DAC

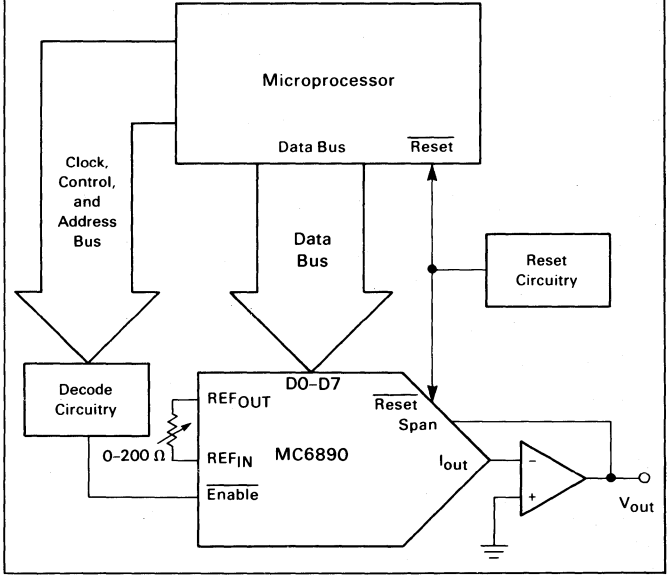
SILICON MONOLITHIC INTEGRATED CIRCUIT



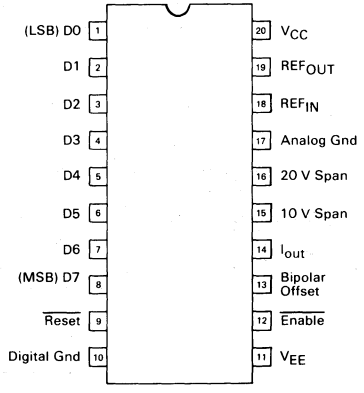
L SUFFIX
CASE 732-03

3

OPERATION WITH AN MPU



PIN CONNECTIONS



ORDERING INFORMATION

Device	Temperature Range	Package
MC6890L	0° to +70°C	Ceramic DIP
MC6890AL	-55° to +125°C	Ceramic DIP

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

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC} V_{EE}	+7.0 -18	Vdc
Digital Input Voltage, Pins 1-8, 12 Pin 9	V_{in}	-3.0 to +7.0 0 to +7.0	Vdc
Applied Output Voltage	V_{14}	V_{EE} +2.0 to V_{EE} +24	Vdc
Reference Amplifier Input	V_{18}	±7.5	Vdc
Operating Temperature Range MC6890L, MC6890AL	T_A	 0 to +70 -55 to +125	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Junction Temperature	T_J	+150	°C

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V, V_{EE} = -12 V, Pin 18 loaded only by Pin 19 through 100 Ω. Reset high, $T_A = T_{low}$ to T_{high}[1], unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Digital Input Logic Levels High Level, Logic 1 Low Level, Logic 0	 V_{IH} V_{IL}	 2.0 —	 — —	 — 0.8	Vdc
Digital Input Current Data (V_{IH} = 3.0 V) (V_{IL} = 0.4 V) Enable (V_{IH} = 3.0 V) (V_{IL} = 0.4 V) Reset ($V_{IH} = V_{CC}$) (V_{IL} = 0.4 V)	 I_{IH} I_{IL} I_{IH} I_{IL} I_{IH} I_{IL}	 — — — — — —	 0.001 0.5 0.001 -6.5 0.001 -1.0	 1.0 -10 1.0 -100 1.0 -15	 μA μA μA μA μA μA
Full Scale Output Current — Unipolar	I_O	-1.50	-1.992	-2.50	mA
Unipolar Zero Output — All Bits Off (T_A = 25°C)	—	—	0.010	0.20	μA
Output Voltage Temperature Coefficient Unipolar Zero Bipolar Zero Full Scale Range	TC_{VO}	 — — —	 ±1.0 ±5.0 ±20	 ±2.0 ±15 ±50	ppm of FSR/°C
Output Voltage, Full Scale Range (See Figure 3) (T_A = 25°C) (10 V Span) (20 V Span) (5.0 V Span)	V_O	 9.861 19.722 4.930	 9.961 19.922 4.980	 10.061 20.122 5.030	Vdc
Output Voltage, Bipolar Zero (MSB on) (See Figure 4) (T_A = 25°C) (10 V Span) (20 V Span) (5.0 V Span)	V_O	 — — —	 0 0 0	 ±20 ±40 ±10	mV
DAC Output Resistance — Exclusive of Span Resistors (T_A = 25°C) (See Figure 5)	R_O	1.0	5.0	—	MΩ
Resolution	—	8.0	8.0	8.0	Bits
Nonlinearity — Relative Accuracy (See Terminology)	NL	—	—	±0.19 (±1/2 LSB)	%
Differential Nonlinearity	Monotonicity Guaranteed				
Differential Nonlinearity (T_A = 25°C) (See Terminology)	—	—	—	±0.29 (±3/4 LSB)	%
Reference Input Resistor	R_{REF}	3800	4900	6800	Ω
Reference Output Voltage (T_A = 25°C)	V_{REF}	2.470	2.500	2.530	Vdc
Reference Output Impedance (T_A = 25°C) I_{load} = 0-3.0 mA	—	—	0.3	1.0	Ω
Reference Short Circuit Current (T_A = 25°C)	I_{REF}	15	30	50	mA
Reference Output Voltage Temperature Coefficient	$TC_{VO(REF)}$	—	±20	—	ppm/°C
Power Supply Range	V_{CC} V_{EE}	4.5 -16.5	5.0 -12	5.5 -4.5	Vdc
Power Supply Current — All Bits Low (V_{CC} = 5.0 V) (V_{EE} = -5.0 V) (V_{EE} = -15 V)	 I_{CC} I_{EE} I_{EE}	 — — —	 10 -10 -10	 20 -15 -15	mA
Power Supply Rejection (T_A = 25°C) To V_{CC} (V_{CC} = 4.5 to 5.5 V) To V_{EE} (V_{EE} = -4.5 V to -16.5 V)	PSR	 — —	 0.010 0.10	 ±1/10 ±1/2	LSB
Power Dissipation — All Bits Low For V_{CC} = 4.5 V, V_{EE} = -4.5 V For V_{CC} = 5.5 V, V_{EE} = -16.5 V	P_D	 — —	 90 220	 158 358	mW

NOTE 1: T_{low} = -55°C for MC6890A, 0° for MC6890
T_{high} = +125°C for MC6890A, +70°C for MC6890

AC SPECIFICATIONS (V_{CC} = 5.0 V, V_{EE} = -12 V, T_A = 25°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Current Settling Time (Enable Positive Edge to ±1/2 LSB Output)	t_s	—	200	300*	ns
Data Setup Time	$t_{su(D)}$	70	40	—	ns
Data Hold Time	$t_{h(D)}$	10	0	—	ns
Pulse Widths					ns
Enable	$t_{W(\overline{E})}$	70	20	—	
Reset	$t_{W(\overline{R})}$	100*	—	—	
Propagation Delays					ns
Enable, Low to High	$t_{PLH(\overline{E})}$	—	100	—	
Reset, High to Low (I_O < 1.0 μA)	$t_{PHL(\overline{R})}$	—	250	—	

*Not 100% tested, guaranteed by design

FIGURE 1 — TIMING DIAGRAM

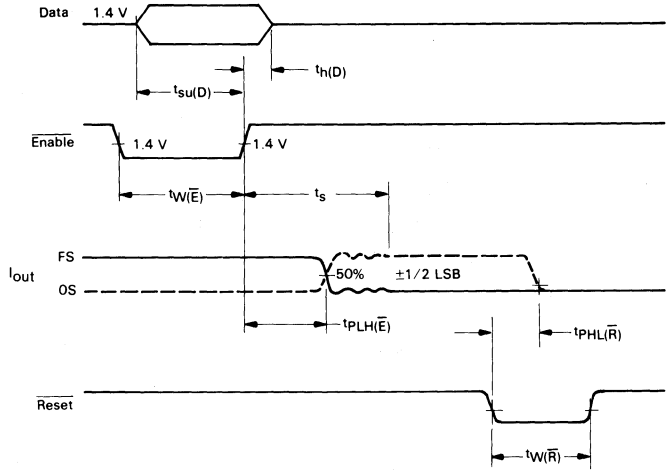
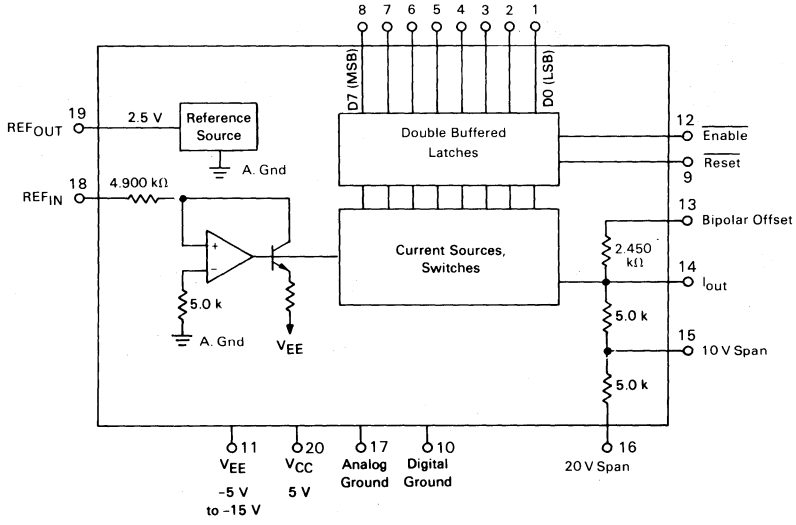


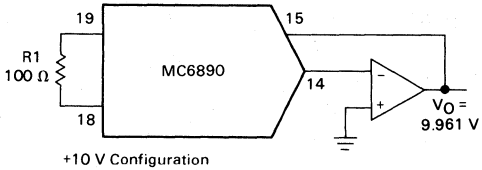
FIGURE 2 — BLOCK DIAGRAM



TEST FIGURES

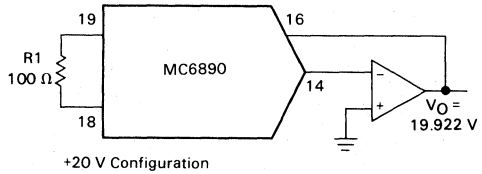
UNIPOLAR CONFIGURATIONS

FIGURE 3A



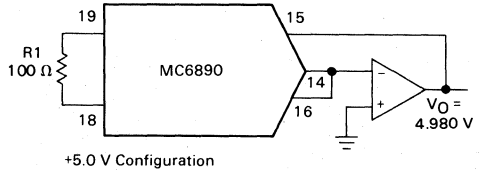
+10 V Configuration
Latched Input Code: 11111111

FIGURE 3B



+20 V Configuration
Latched Input Code: 11111111

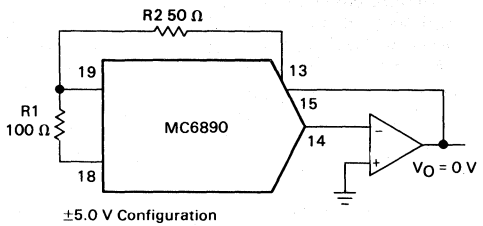
FIGURE 3C



+5.0 V Configuration
Latched Input Code: 11111111

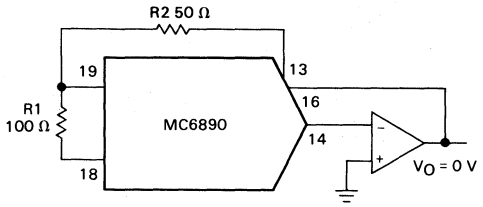
BIPOLAR CONFIGURATIONS

FIGURE 4A



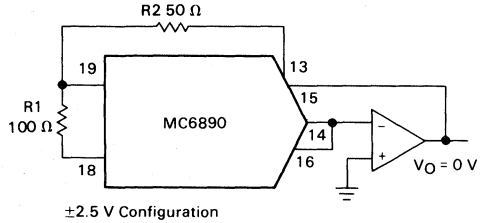
±5.0 V Configuration
Latched Input Code: 10000000

FIGURE 4B



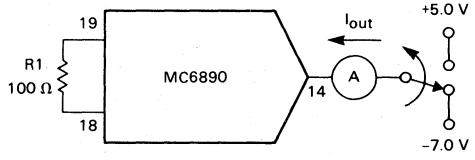
±10 V Configuration
Latched Input Code: 10000000

FIGURE 4C



±2.5 V Configuration
Latched Input Code: 10000000

FIGURE 5 TEST CONFIGURATION FOR DAC OUTPUT IMPEDANCE



Latched Input Code: 11111111

$$R_{out} = \frac{12\ V}{\Delta I_{out}}$$

TERMINOLOGY

Nonlinearity (Relative Accuracy) — Maximum output deviation from ideal straight line connecting zero and full-scale readings, expressed as a fraction of LSB or percent of full scale.

Differential Nonlinearity — Maximum deviation in the readings of any two adjacent input bit codes from the ideal LSB step, expressed in fractions of LSB or percentage of full scale. A differential nonlinearity value greater than 1 LSB may lead to non-monotonic operation.

Monotonicity — For every increase in the input digital word, the output current either remains the same or increases. The MC6890 is guaranteed to be monotonic over temperature.

Settling Time — The elapsed time from the $\overline{\text{Enable}}$ positive transition until the output has settled within an error band about its final value.

The worst case switching condition occurs when all bits are latched "on," which corresponds to a low-to-high transition for all bits. This time is typically 200 ns for the current output to settle to within ±1/2 LSB for 8 bit accuracy. These times apply when the output swing is limited to a small (<0.5 V) swing and the external output capacitance is under 10 pF.

Gain Error — The difference between the actual full scale range and the ideal full scale range. Based on a 0 to 10 V output configuration, the ideal FSR is $\frac{255}{256} \times 10\text{ V} = 9.961\text{ V}$.

Gain error is laser trimmed to less than ±1.0% with R1 = 100 Ω (Figure 3) and can be user trimmed to zero error with R1 = 200 Ω pot.

Bipolar Zero — Using the configuration shown in Figure 6 with R1 = 100 Ω, R2 = 50 Ω, with the MSB on and all other bits off, the output voltage reading compared to analog ground is expressed as a percentage of the full-scale range. Offset voltage of the output op amp must be nulled. Bipolar Zero error is laser trimmed to less than 0.20% and can be user trimmed to zero with R2 = 100 Ω pot.

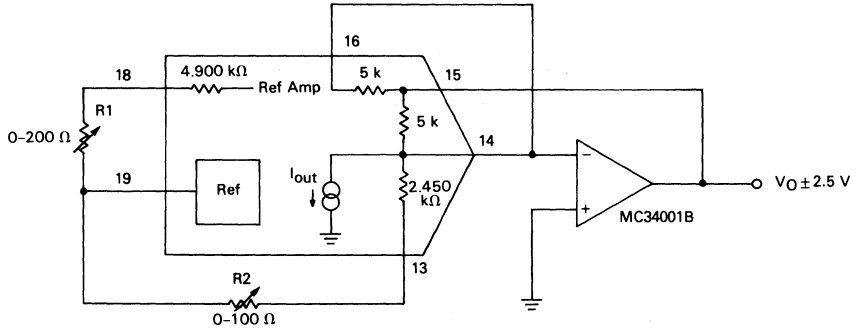
Temperature Coefficients — (Unipolar zero, Bipolar zero, Gain and Reference Output). The maximum deviation of the particular parameter over the specified temperature range, divided by the temperature range, expressed in parts per million of Full Scale Range per degree C.

Power Supply Rejection — The change in full scale current caused by the specified change in V_{EE} or V_{CC} is expressed in LSB's.

$\overline{\text{Reset}}$ Function — The MC6890 has a $\overline{\text{Reset}}$ pin (9) that will force the DAC's registers, and therefore the DAC output current, to zero. This input is active low and should not occur simultaneously with an active $\overline{\text{Enable}}$ signal although no harm would result to the converter. The power dissipation increases slightly during $\overline{\text{Reset}}$ low. $\overline{\text{Reset}}$ should not be allowed to become more negative than ground.

3

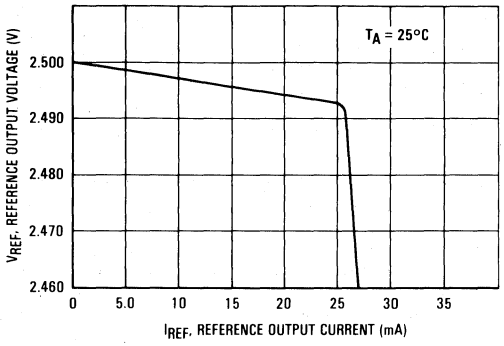
FIGURE 6 — MC6890 IN TYPICAL BIPOLAR ±2.5 V OPERATION



D7	D6	D5	D4	D3	D2	D1	D0	V_O (Volts)	
								R2 ≅ 60 Ω	R2 ≅ 50 Ω
1	1	1	1	1	1	1	1	+2.490	+2.480
1	1	1	1	1	1	1	0	+2.470	+2.460
1	0	0	0	0	0	0	0	+0.010	+0.000
0	1	1	1	1	1	1	1	−0.010	−0.020
0	0	0	0	0	0	0	1	−2.470	−2.480
0	0	0	0	0	0	0	0	−2.490	−2.500

TYPICAL PERFORMANCE CURVES

FIGURE 7 — REFERENCE VOLTAGE versus EXTERNAL LOAD CURRENT*



*External load current is in addition to Reference Input Current (Pin 18) of D/A converter.

FIGURE 8 — DIGITAL INPUT CHARACTERISTICS

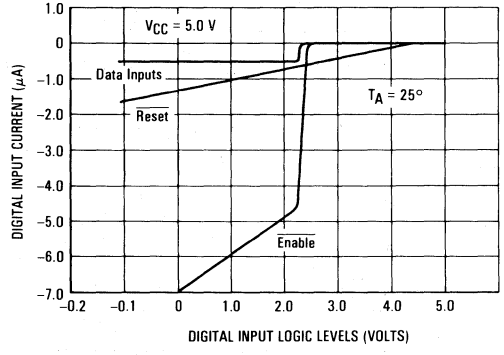
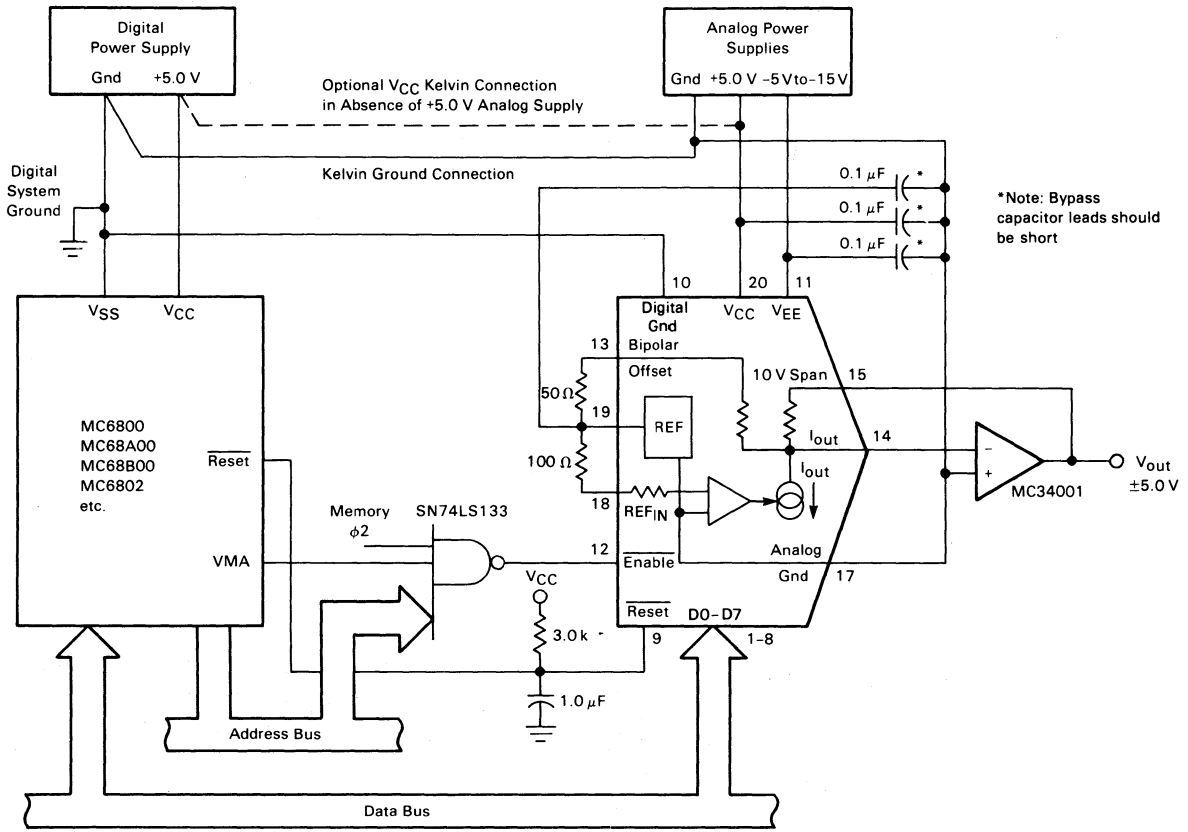


FIGURE 9 — TYPICAL APPLICATION OF THE MC6890 IN A MC6800 SERIES MPU SYSTEM



MOTOROLA

MC68120
MC68121

Advance Information

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

INTELLIGENT PERIPHERAL CONTROLLER

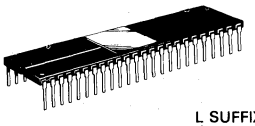
3

INTELLIGENT PERIPHERAL CONTROLLER

The MC68120/MC68121 Intelligent Peripheral Controller (IPC) is a general purpose, mask programmable peripheral controller. The IPC provides the interface between an M68000 or M6800 Family microprocessor and the final peripheral devices through a system bus and control lines. System bus data is transferred to and from the IPC via dual-port RAM while the software utilizes the semaphore registers to control RAM tasking or any other shared resource. Multiple operating modes range from a single chip mode with 21 I/O lines and 2 control lines to an expanded mode supporting an address space of 64K bytes. The MC68120 has 2K bytes of on-chip ROM to make full use of all operating modes. The MC68121 utilizes only the expanded address modes, due to the absence of on-chip ROM.

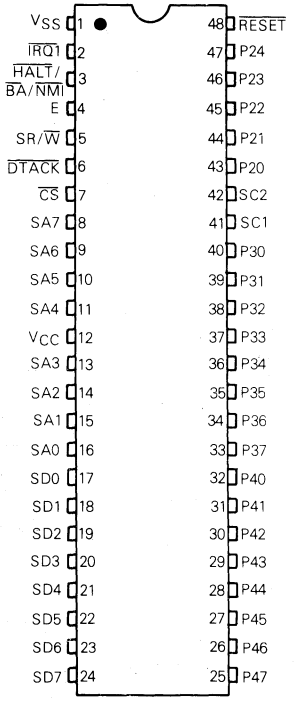
A serial communications interface, 16-bit timer, dual-ported RAM and semaphore registers are available for use by the IPC in all operating modes.

- System Bus Compatible with the Asynchronous M68000 Family
- System Bus Compatible with the MC6809 and Other M6800 Family Processors/Peripherals
- Local Bus Allows Interface with all M6800 Peripherals
- MC6801 Source and Object Code Compatible
- Upward Compatible with MC6800 Source and Object Code
- 2048 Bytes of ROM (MC68120 Only)
- 128 Bytes of Dual-Ported RAM
- Multiple Operation Modes Ranging from Single Chip to Expanded, with 64K Byte Address Space
- Six Shared Semaphore Registers
- 21 Parallel I/O Lines and 2 Handshake Lines (5 I/O Lines on MC68121)
- Serial Communications Interface (SCI)
- 16-Bit Three-Function Timer
- 8-Bit CPU and Internal Bus
- Halt/Bus Available Capability Control
- 8×8 Multiply Instruction
- TTL Compatible Inputs and Outputs
- External and Internal Interrupts



L SUFFIX
CERAMIC PACKAGE
CASE 740

PIN ASSIGNMENT

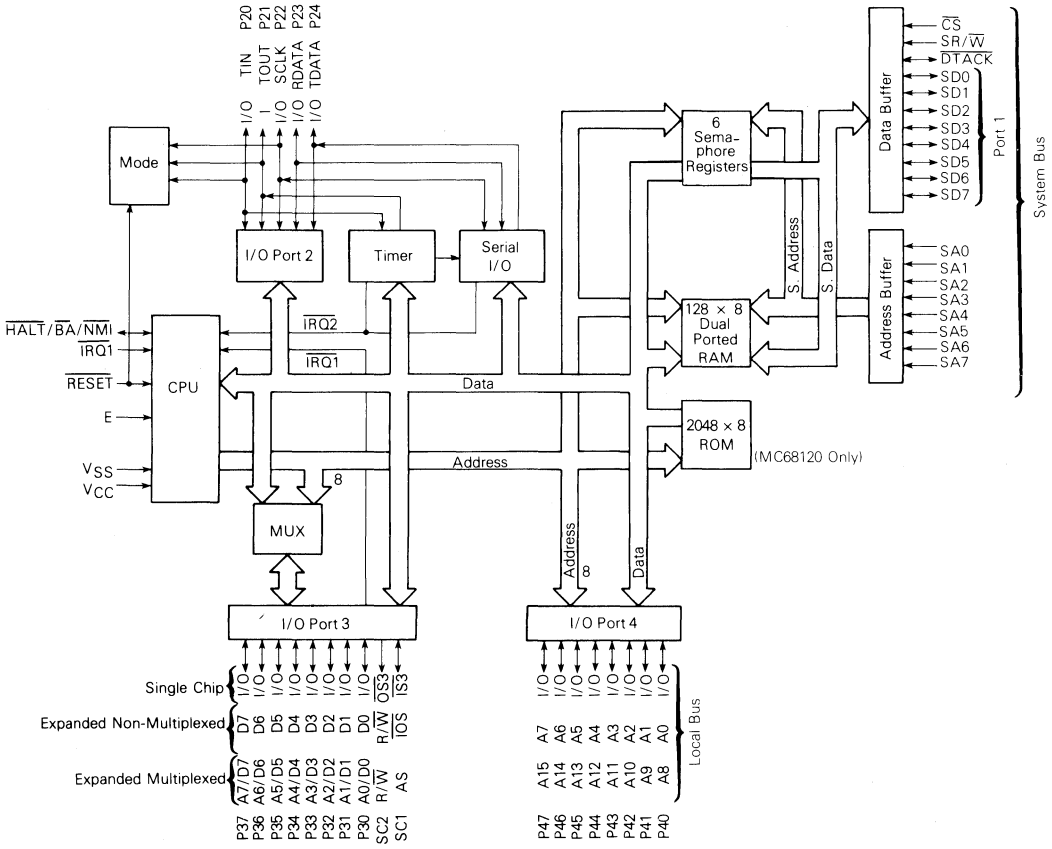


GENERIC INFORMATION
(T_A = 0°C to 70°C)

Package Type	Frequency (MHz)	Generic Number
Ceramic L Suffix	1.0	MC68120L1 (Unicorn ROM)
	1.0	MC68121L
	1.25	MC68120L1-1 (Unicorn ROM)
	1.25	MC68121L-1

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

MC68120/MC68121 INTELLIGENT PERIPHERAL CONTROLLER — BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

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

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

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

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

DC LOCAL BUS ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0° to 70°C unless otherwise noted)
(Refer to Figures 1 and 2)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	E	V_{EIH}	$V_{CC}-0.75$	–	V_{CC}	V
Input Low Voltage	E	V_{EIL}	$V_{SS}-0.3$	–	$V_{SS}+0.6$	V
Input High Voltage	$\overline{RESET}$ Other Inputs*	V_{IH}	$V_{SS}+4.0$ $V_{SS}+2.0$	– –	V_{CC} V_{CC}	V
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Load Current (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	–	0.5	mA
Input Leakage Current (V_{in} = 0 to 5.25 V)	SCI, $\overline{HALT}/\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	1.5	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	SD0-SD7, P20-P24, P30-P37	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{load} = – 65 μA, V_{CC} = min) (I_{load} = – 100 μA, V_{CC} = min)	P40-P47, SC1, SC2 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$	– –	– –	V
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Internal Power Dissipation (measured at T_A = 0°C)		P_{INT}	–	–	1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	E P30-P37, P40-P47, SC1 Other Inputs	C_{in}	– – –	– – –	60.0 12.5 10.0	pF

*Except Mode Programming Levels; See Figure 29.

FIGURE 1 – CMOS LOAD

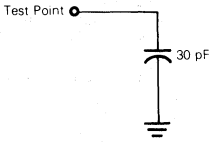
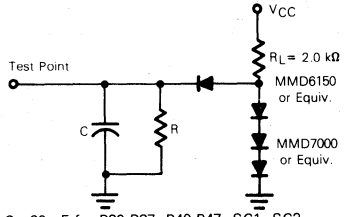


FIGURE 2 – TIMING TEST LOAD PORTS 2, 3, 4



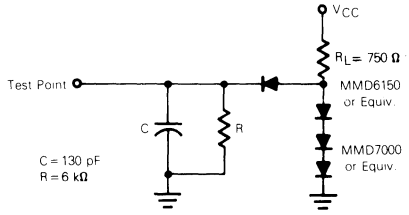
C = 90 pF for P30-P37, P40-P47, SC1, SC2
= 30 pF for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$, P30-P37

DC SYSTEM BUS ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 70°C unless otherwise noted) (Refer to Figure 3)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IL}	V_{SS} – 0.3	–	V_{SS} + 0.8	V
Output High Voltage (I_{Load} = – 400 μA, V_{CC} = min)	$\overline{DTACK}$, SD0-SD7	V_{OH}	V_{SS} + 2.4	–	–	V
Output Low Voltage (I_{Load} = 5.3 mA, V_{CC} = min)	$\overline{DTACK}$, SD0-SD7	V_{OL}	–	–	V_{SS} + 0.5	V

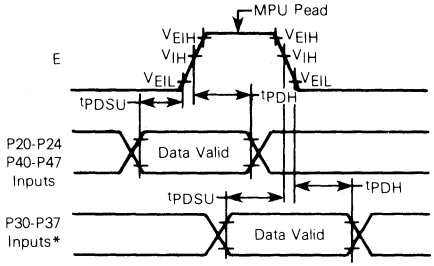
FIGURE 3 – TIMING TEST LOAD SD0-SD7, $\overline{DTACK}$



PERIPHERAL PORT TIMING (Refer to Figures 4 through 7)

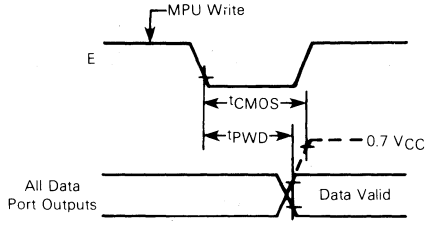
Characteristics	Symbol	Min	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid (Ports 2, 3, 4)	t_{PWD}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	ns
Input Data Hold Time	t_{IH}	60	–	ns
Input Data Setup Time	t_{IS}	20	–	ns
Input Capture Pulse Width (Timer Function)	t_{PWIC}	2	–	E_{cyc}

FIGURE 4 – DATA SETUP AND HOLD TIMES (MPU READ LOCAL BUS)



*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

FIGURE 5 – DATA SETUP AND HOLD TIMES (MPU WRITE LOCAL BUS)



Notes:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

FIGURE 6 – PORT 3 OUTPUT STROBE TIMING (SINGLE CHIP MODE)

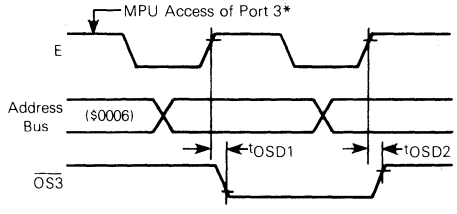
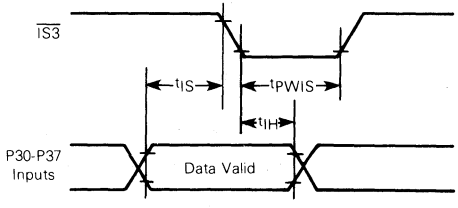


FIGURE 7 – PORT 3 LATCH TIMING (SINGLE CHIP MODE)



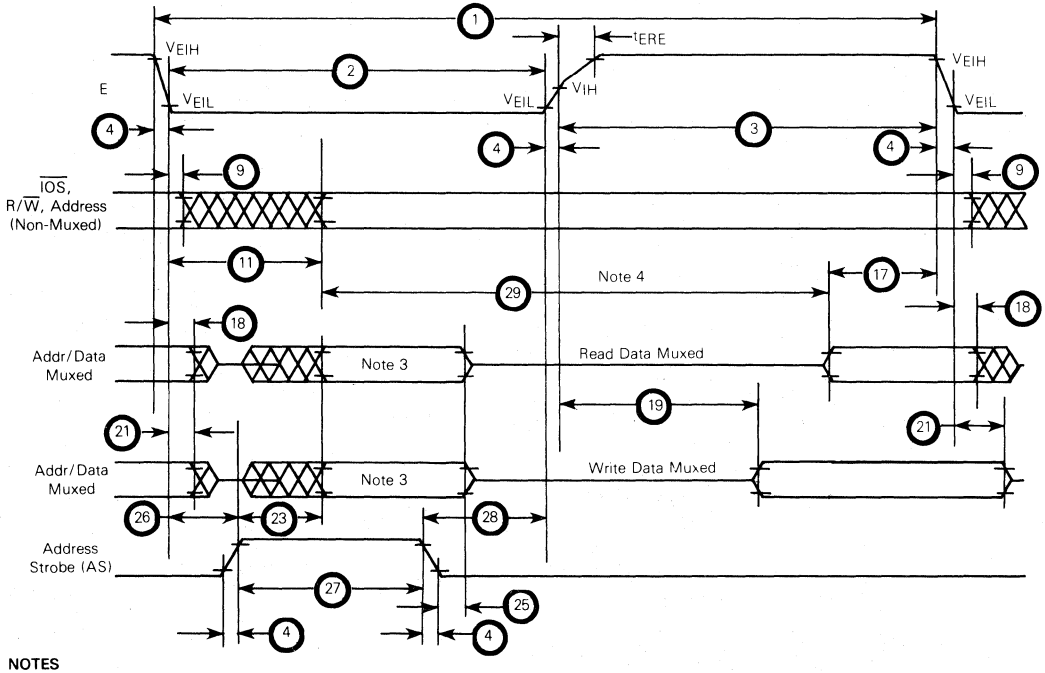
*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

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

LOCAL BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC68120/ MC68121		MC68120-1/ MC68121-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Non-Muxed Address Hold Time	t_{AH}	20	–	20	–	ns
11	Address Delay From E Low	t_{AD}	–	260	–	220	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	ns
23	Muxed Address Delay from AS	t_{ADM}	–	90	–	80	ns
25	Muxed Address Hold Time	t_{AHL}	20	110	20	110	ns
26	Delay Time E to AS Rise	t_{ASD}	100	–	80	–	ns
27	Pulse Width, AS High	PW_{ASH}	220	–	170	–	ns
28	Delay Time AS to E Rise	t_{ASED}	100	–	80	–	ns
29	Usable Access Time (Note 4)	t_{ACC}	570	–	435	–	ns
	Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns
	Processor Control Setup Time	t_{PCS}	200	–	200	–	ns
	Processor Control Hold Time	t_{PCH}	20	40	20	40	ns

FIGURE 8 – LOCAL BUS TIMING



NOTES

1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Address valid on the occurrence of the latest of 11 or 23.
4. Usable access time is computed by: 1 – (4 + 11 + 17).

ASYNCHRONOUS SYSTEM BUS TIMING (Refer to Figures 9, 10, 11 and 12)

Characterisic	Symbol	Min	Typ	Max	Unit
Cycle Time	t_{cyc}	0.8	–	2.0	μs
System Address Setup	t_{SAS}	30	–	–	ns
System Address Hold	t_{SAH}	0	–	–	ns
System Data Delay Read Semaphore	t_{SDDR}	0.3	–	0.3+1.5 t_{cyc}*	μs
RAM	t_{SDDR}	–	315	–	ns
System Data Valid	t_{SDV}	0	–	–	ns
System Data Hold Read	t_{SDHR}	0	–	100	ns
System Data Delay Write Semaphore	t_{SDDW}	**	–	**	ns
RAM	t_{SDDW}	–	–	60	ns
System Data Hold Write	t_{SDHW}	0	–	–	ns
Data Acknowledge Semaphore	t_{DAL}	0.5	–	0.5+1.5 t_{cyc}*	μs
RAM	t_{DAL}	–	315	–	ns
Data Acknowledge High	t_{DAH}	–	–	60	ns
Data Acknowledge Three-State	t_{DAT}	–	–	90	ns
Data Acknowledge Low to $\overline{CS}$ High	t_{DCS}	60	–	–	ns

* Actual value dependent upon clock period.
** Data need not be valid on write to Semaphore Registers.

3

FIGURE 9 — ASYNCHRONOUS READ OF SEMAPHORE REGISTER

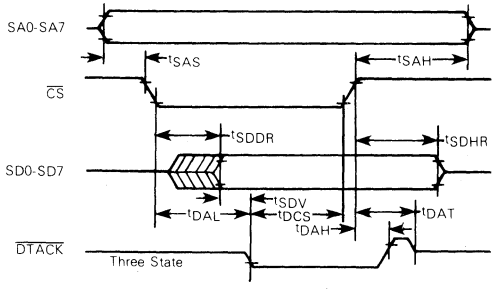


FIGURE 10 — ASYNCHRONOUS WRITE OF SEMAPHORE REGISTER

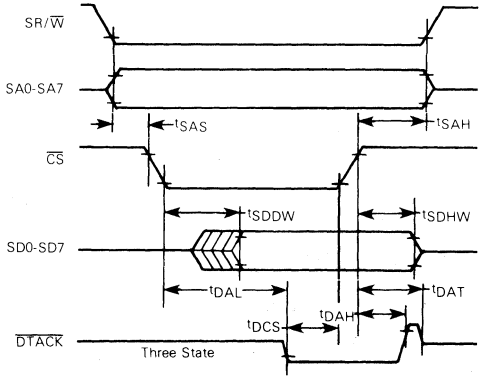


FIGURE 11 — ASYNCHRONOUS READ OF RAM

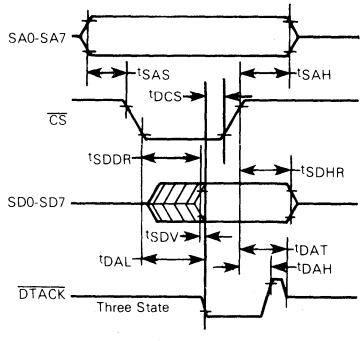
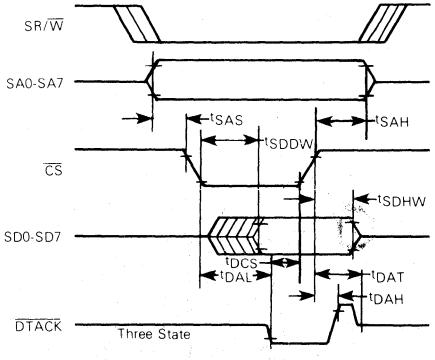


FIGURE 12 — ASYNCHRONOUS WRITE OF RAM



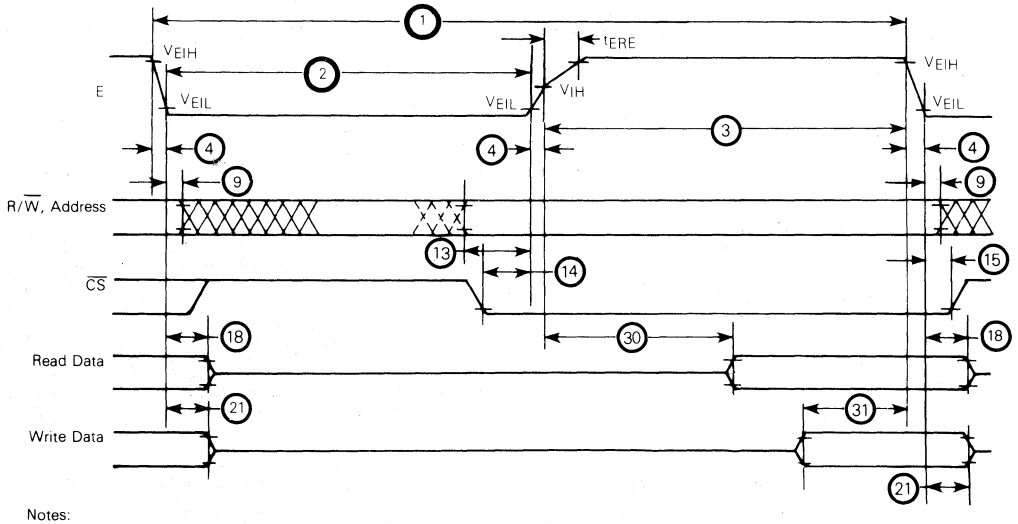
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

SYNCHRONOUS SYSTEM BUS TIMING (See Notes 1 and 2)

Ident Number	Characteristic	Symbol	MC68120/ MC68121		MC68120-1 MC68121-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.80	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	360	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	360	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Address Hold Time	t_{AH}	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	70	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	70	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	30	100	30	85	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	250	ns
31	Input Data Setup Time	t_{DSW}	165	–	120	–	ns
	Clock Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns

3

FIGURE 13 – SYNCHRONOUS SYSTEM BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

INTRODUCTION

The MC68120/MC68121 is an 8-bit Intelligent Peripheral Controller (IPC) which can be configured to function in a wide variety of applications. This extraordinary flexibility is provided by its ability to be hardware programmed into eight different operating modes. These operating modes allow the IPC to operate on its local bus and communicate with an external system bus through the internal dual-ported RAM. The operating mode controls the configuration of 18 of the 48 pins on the IPC, the available on-chip resources, the memory map, the location (internal or external) of interrupt vectors, and the type of local bus. The configuration of the remaining 30 pins is not controlled by the operating mode.

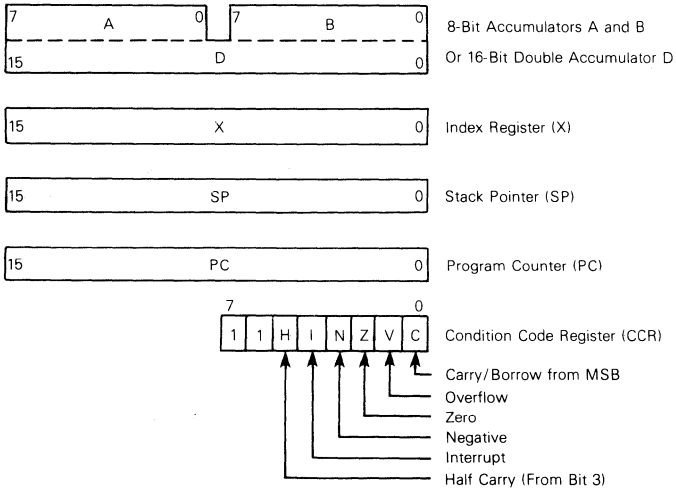
The dual-ported RAM provides a vehicle for devices on two separate buses to exchange data without directly affecting the devices on the other bus. The dual-ported RAM is accessible from the MC68120/MC68121 CPU and accessible synchronously or asynchronously to the system bus through Port 1. Semaphore registers are provided as a software tool to arbitrate shared resources such as the dual-ported RAM. The semaphore registers are accessible from both buses in the same way each bus accesses the dual-ported RAM.

The remaining ports (2, 3, and 4) are I/O ports. Each port is controlled by its Data Direction Register. The CPU has direct access to the port pins of each port through its Data Register. Port pins are labeled as P_{ij} where i identifies one of three ports and j indicates the particular bit. Port 2 is a 5-bit port which may be configured for I/O or for use of the on-chip timer and Serial Communications Interface (SCI). Ports 3 and 4 may be used as 16 bits of I/O or may form a local address and data bus with control lines allowing communications with external memory and peripherals.

The IPC contains an enhanced M6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800 and directly compatible with the MC6801. The programming model is depicted in Figure 14, where accumulator D is a concatenation of accumulators A and B.

The MC68121 has all of the features of the MC68120 with the exception of on-chip ROM. Thus the MC68121 normally operates in the modes utilizing external ROM (modes 2 and 3). Therefore, modes 0, 1, 4, 5, 6 and 7 should not be used.

FIGURE 14 — PROGRAMMING MODEL



DUAL-PORTED RAM AND SEMAPHORE REGISTERS

The dual-ported RAM may be accessed from both the MC68120/MC68121 CPU and the external system bus. The six semaphore registers are tools provided for the programmer's use in arbitrating simultaneous accesses of the same resource.

For the internal CPU, the dual-ported RAM is located from $0080 through $00FF in all modes except 3 and 4. In mode 3, the dual-ported RAM has been relocated in high memory from $C080 through $C0FF thus allowing use of direct addressing mode on external memory/peripherals. Note that no direct addressing of internal control registers is possible in mode 3. In mode 4, the internal RAM is not fully decoded and appears in locations $XX80 through $XXFF. From the external system bus, the dual-ported RAM is found in locations %10000000-11111111, as shown below in Table 1.

TABLE 1 — LOCATION OF SEMAPHORE REGISTERS AND DUAL-PORTED RAM

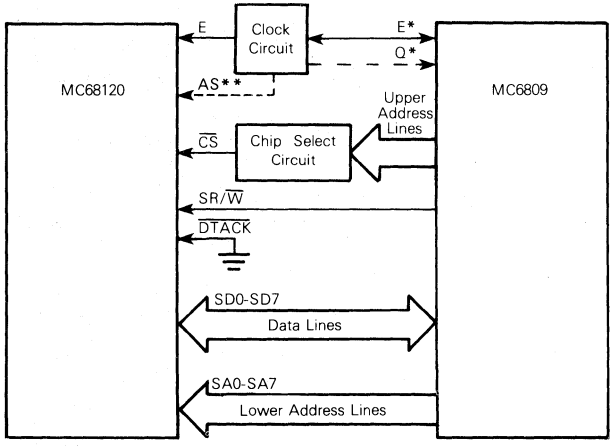
System Bus Address (SA7-SA0)	Feature	IPC Address*
%0000 0000 — 0001 0110	Reserved	------
---- ---- — ---- ----	Internal Registers	$00-16
0001 0111 — 0001 1100	Semaphore Registers	17-1C
0001 1101 — 0111 1111	Reserved	1D-1F
---- ---- — ---- ----	External Mem./Unusable*	20-7F
1000 0000 — 1111 1111	Dual-Ported RAM	80-FF

% = Binary; $ = Hexadecimal
* Mode Dependent

The reserved memory areas %0-0001 0110 and %0001 1101-%0111 1111 cannot be written to from the System bus. If read from the System bus these memory locations return a value of $FF.

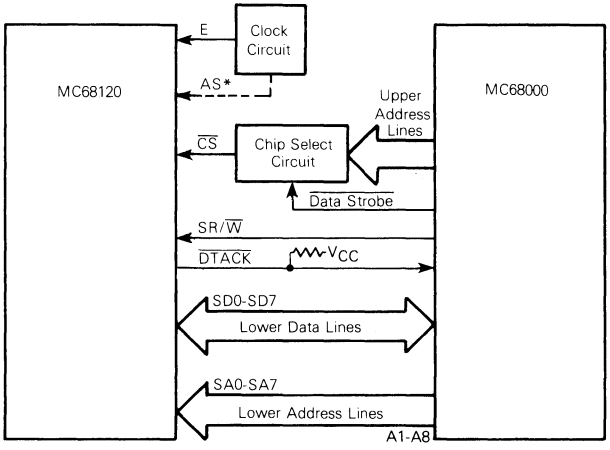
The dual-ported RAM is accessed from the external System bus by way of eight address lines (SA0-SA7) and eight data lines (SD0-SD7). Three control lines provide for synchronous or asynchronous access to the dual-ported RAM through Port 1. Figure 15 shows an example of a synchronous interface (using MC6809) and Figure 16 shows an example of an asynchronous interface (using MC68000). The dual-ported RAM is selected in each case by address lines SA0-SA7 and Chip Select ($\overline{CS}$) from the system bus. The direction of data transfer is selected by the System Read/Write (SR/$\overline{W}$) line. The Data Transfer Acknowledge ($\overline{DTACK}$) signal is the asynchronous handshake required by an MC68000. Refer to $\overline{DTACK}$ under Functional Pin Description for more information. $\overline{DTACK}$ can be used to control a Memory Ready signal on the M6800 Family processor where Memory Ready capability is provided (see Figure 17). The latter would allow the M6800 Family processor to run asynchronously with the MC68120/MC68121. It should be noted that if the Memory Ready signal (on M6800 processors) is to be used with the $\overline{DTACK}$ signal, the system clock must be faster than or equal to the clock driving the IPC. Example clock circuits are shown in Figures 18 and 19.

FIGURE 15 — SYNCHRONOUS SYSTEM BUS ACCESS INTERFACE



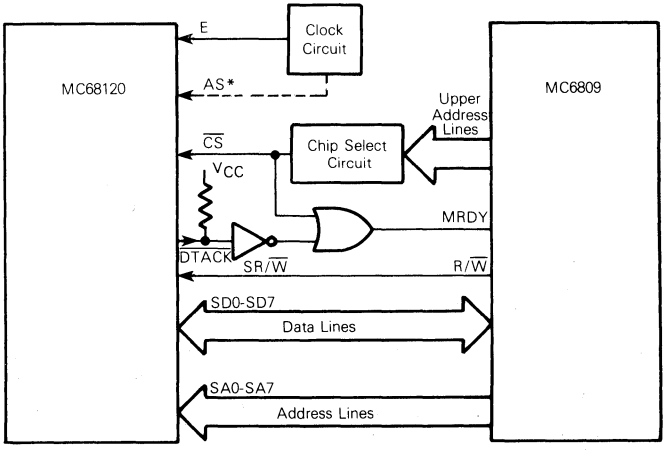
*E and Q are inputs for MC6809E
**Only needed in expanded multiplexed modes.

FIGURE 16 — ASYNCHRONOUS SYSTEM BUS INTERFACE



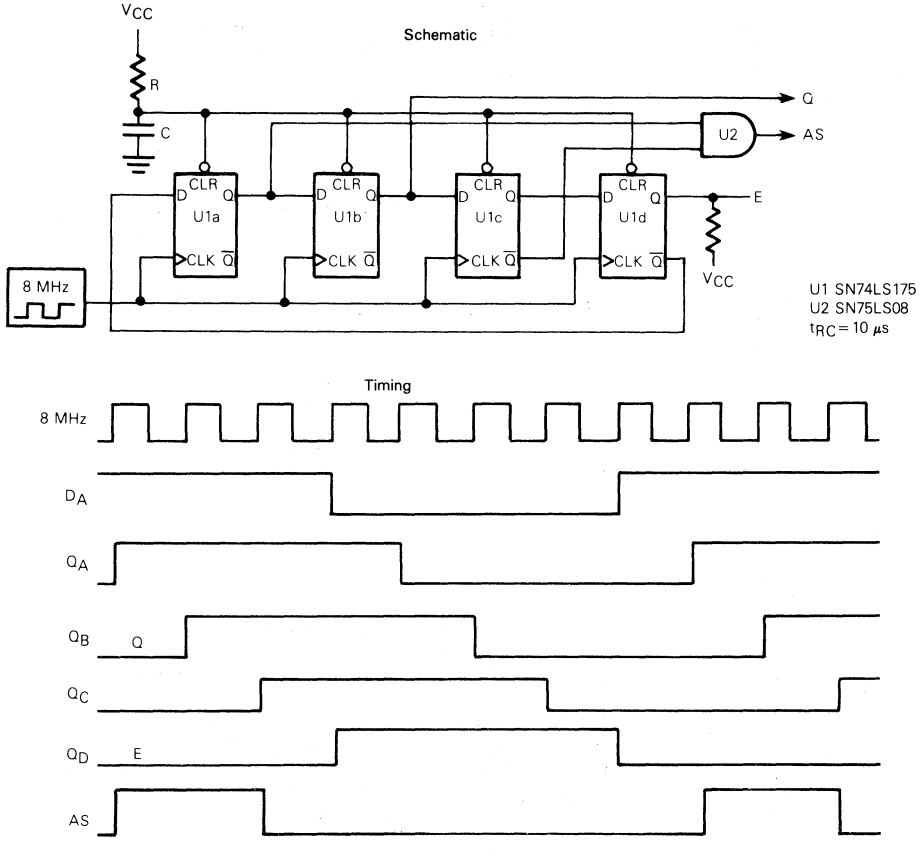
* Only needed in expanded multiplexed modes.

FIGURE 17 — MEMORY READY — DTACK CONFIGURATION



* Only needed in expanded multiplexed modes.

FIGURE 18 — CLOCK CIRCUIT EXAMPLE 1 — SCHEMATIC AND TIMING



The semaphore registers allow arbitration between shared resources, which may be part or all of the dual-port RAM, or a peripheral. The semaphore registers may also be used to indicate that non-reentrant code is in use or that a task is in process or is complete. To prevent the writing or reading of erroneous data from the dual-ported RAM, all simultaneous accesses involving a write to the same byte in the dual-ported RAM should be avoided. The responsibility for mutual exclusion resides in software. The semaphore registers are a convenient means for the software to control the simultaneous accesses involving a write to the dual-ported RAM. Each of the six semaphore registers consist of a semaphore bit (SEM, bit 7) and an ownership bit (OWN, bit 6). The remaining six bits (b0-b5) will read all zeros.

SEMAPHORE REGISTER

7	6	5	4	3	2	1	0
SEM	OWN	0	0	0	0	0	0

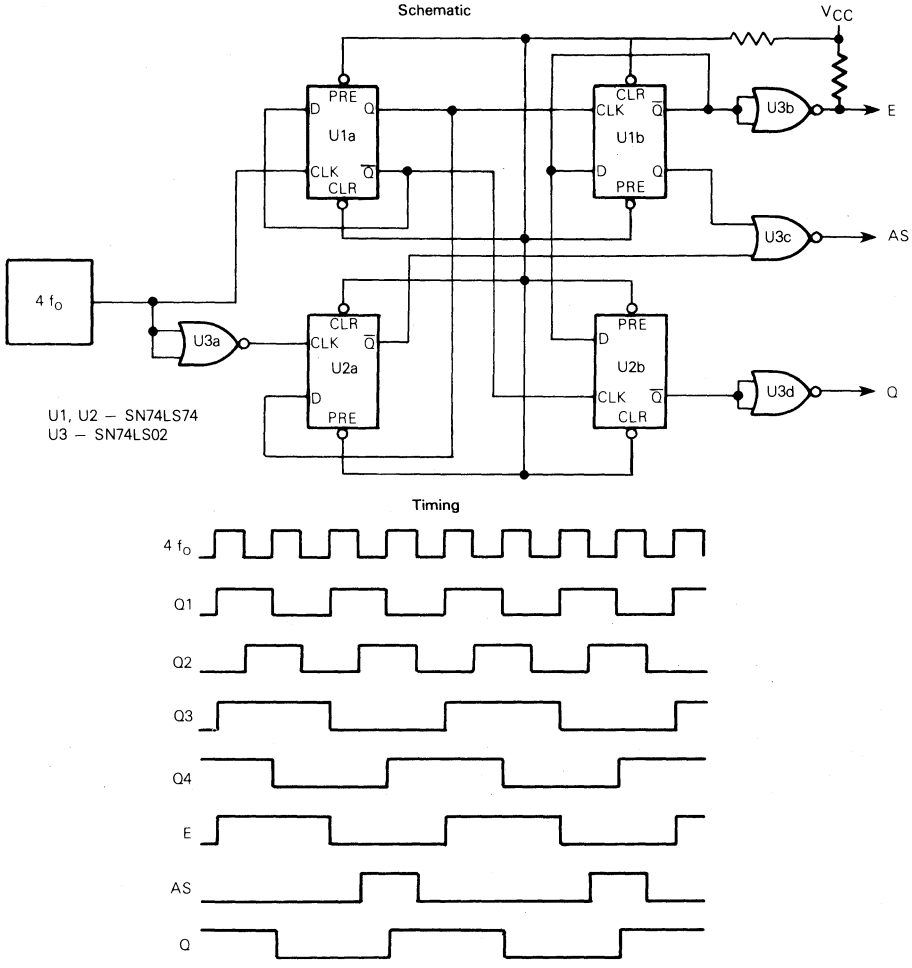
The semaphore bits are test and set bits with hardware arbitration during simultaneous accesses. Basically, the semaphore bit is cleared when written and set when read, during a single processor access. This is shown in Table 2.

TABLE 2 — SINGLE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	R/$\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	W	—	0
1	W	—	0

*0 — Resource Available
1 — Resource Not Available

FIGURE 19 — CLOCK CIRCUIT EXAMPLE 2 — SCHEMATIC AND TIMING



The data written is disregarded and the information obtained from the Read may be interpreted as: 0 — resource available; 1 — resource not available. Thus, any write to a semaphore clears the semaphore bit and makes the associated resource "available."

An access where both the IPC and system processors attempt to read or write the same semaphore register simultaneously is a contested access. During a contested access, the hardware decides which processor reads a clear semaphore bit and which reads a set semaphore bit. Table 3 describes contested operation of a semaphore bit.

The IPC always reads the actual semaphore bit; the system processor reads the semaphore bit in all cases except the simultaneous read of a clear semaphore bit. This arbitration during a simultaneous read ensures that only one processor reads a clear bit and therefore controls the resource; that processor is arbitrarily the IPC.

In Table 3, the first four states are considered proper and they occur in correctly written software. The last four states are improper and only exist in improperly written software.

The ownership bit is a read-only bit that indicates which processor sets the semaphore bit. If the semaphore bit is set, the ownership bit indicates which processor set it. If the semaphore bit is not set, the ownership bit indicates which processor last set the semaphore bit; OWN = 0, the other processor set SEM; OWN = 1, this processor set SEM.

The reset state of the semaphore and ownership bits is defined in Table 4. All of the semaphore bits are set after an MC68120/MC68121 reset. The IPC owns all of them except the second semaphore which is owned by the system processor. This configuration should prevent the system processor from reading a clear semaphore and implying the system processor set it when the IPC $\overline{\text{RESET}}$ is held low.

TABLE 3 — DUAL PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	IPC R/$\overline{W}$	IPC Data Read	System R/$\overline{W}$	System Data Read	Resulting SEM Bit	
0	R	0*	R	1*	1	PROPER
1	R	1*	W	—	0	
1	W	—	R	1*	0	
1	R	1	R	1*	1	
0	W	—	W	—	0	IMPROPER
0	R	0*	W	—	1	
1	W	—	W	—	0	
0	W	—	R	0*	1	

*0 — Resource Available
1 — Resource Not Available

TABLE 4 — RESET STATE OF SEMAPHORE REGISTER

SEM Reg No.	IPC Sem	IPC Own	System Sem	System Own
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

3

PROGRAM STORAGE MEMORY — ROM

The standard MC68120 comes preprogrammed with a monitor in the ROM. Custom programs are placed in ROM by special order (see Appendix A).

The MC68120 contains 2048 bytes of on-chip, mask programmable read-only memory (ROM) in memory locations $F800 through $FFFF. The contents of this ROM allows the IPC to perform a custom function for the user. The interrupt vectors $FFF0-$FFFF are decoded to provide vectors at the top of resident ROM. Address $FFEF is reserved for the checksum value for the ROM. This value is the complement of the "Exclusive OR" of the 2047 bytes of mask programmed ROM. An IPC without ROM is also available as the MC68121. The MC68121 should only be used in modes 2 and 3 to access external ROM after reset.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power and ground to the IPC. The power supply should provide +5 volts (±5%) to V_{CC} and V_{SS} should be tied to ground. Total power dissipation should not exceed P_D milliwatts.

RESET

The reset function is used for three purposes. The first is to provide the IPC with an orderly and defined start-up procedure from a powerdown condition. The second is to return to start-up conditions without an intervening powerdown condition. The third is to provide a control signal to latch the operating mode.

During reset (low logic level on $\overline{RESET}$ pin), execution of the current instruction is suspended and the CPU enters a "reset state." The register contents are not pushed onto the stack and their contents become undefined during reset. The "reset state" initializes the IPC, as shown in Table 5.

On the positive edge of $\overline{RESET}$, the IPC latches the operating mode from P22, P21 and P20, and then configures Port 3, Port 4, SC1 and SC2. The restart vector is then fetched and transferred to the program counter, then instruction execution begins.

Reset timing is illustrated in Figure 20. The $\overline{RESET}$ line must be held low for a minimum of three E-cycles for the IPC to complete its entire reset sequence. An external RC-network may be used to obtain the required timing.

ENABLE — E

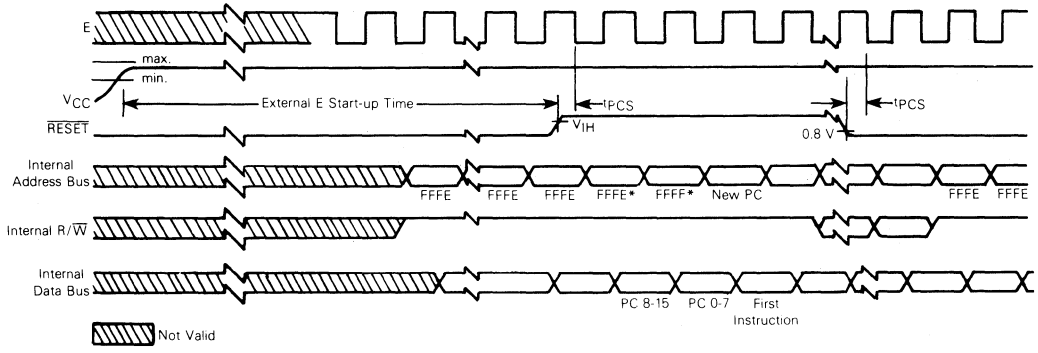
The E clock input is required for timing to synchronize Data Bus transfers. A "CPU E-cycle" (or bus cycle) consists of a negative half-cycle of E followed by a positive half-cycle. For any given bus cycle, the address is valid during the negative half-cycle of E and the selected device must be enabled to the Data Bus during the next positive half-cycle. The data bus is active only while E is high. It should be noted

TABLE 5 — STATE OF IPC DURING RESET

Bits or Registers	Effective State
CPU I-Bit	set (IRQ1 and IRQ2 disabled)
NMI Interrupt Latch	cleared (NMI disabled)
Halt Control Bit	cleared (HALT/BA selected)
All Data Direction Registers	cleared
SCI Rate and Mode Control Register	cleared
Receive Data Register	cleared
Timer Control and Status Register	cleared
Free Running Counter	cleared
Buffer for LSB of Counter	cleared
Port 3 Control and Status Register	cleared
Port 2, 3, 4 Data Registers	undefined after Power-up Reset; and not changed after Reset
SCI Transmit/Receive Control and Status Register	Preset to $20
Output Compare Register	Preset to $FFFF
Semaphore Bits	Preset to 1's
Ownership Bit of Semaphore Register 2	Preset to System Ownership
All other Ownership Bits	Preset to IPC Ownership
All Ports 2 and 3 Lines	High Impedance (inputs)
All Port 4 Lines	High Impedance (inputs) with pullup resistors
SC1*	High Impedance with pullup resistors
SC2	Active High

*If in mode 5, SC1 will go active high; otherwise it will remain in the high impedance state.

FIGURE 20 — RESET TIMING



*Mode 0 — $BFFE, BFFF
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

that this input should have some provision to obtain the specified logical high level which is greater than standard TTL levels.

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

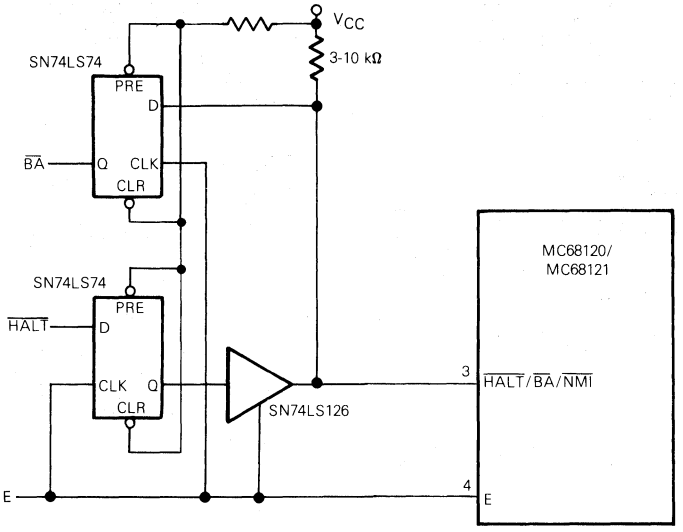
HALT/BUS AVAILABLE/NON-MASKABLE INTERRUPT — HALT/BA/NMI

The HALT/BA/NMI (pin 3) serves one of two functions. These functions are $\overline{NMI}$ or Halt/BA and the function selected is determined by the Halt Control (HC, bit 2) bit of the Functional Control Register (location $14). If the HC bit is set (to a "1"), then the $\overline{NMI}$ function is activated. Alternately, if HC is cleared (to a "0" as it is during reset), the Halt/BA function is activated. An external pullup resistor to V_{CC} is required on pin 3 for either function. Typical pullup resistor values range from 3K to 10K depending on the drive capability of the external device.

When the $\overline{NMI}$ function is implemented, pin 3 is configured as an input. A negative edge on pin 3 then requests an IPC non-maskable interrupt sequence, but the current instruction will be completed before responding to this request. To assure an interrupt under all conditions, $\overline{NMI}$ must be held low for at least one E-cycle. $\overline{NMI}$ may be used to cause the IPC to exit the Wait instruction. For interrupt timing specifications, see the interrupt portion of the Operating Mode Section.

When configured to utilize the Halt/BA function of this pin, such as after reset, the circuit of Figure 21 is recommended to detect and supply continuous $\overline{HALT}$ and $\overline{BA}$

FIGURE 21 — $\overline{HALT}/\overline{BA}$ DEMULTIPLEXING CIRCUIT

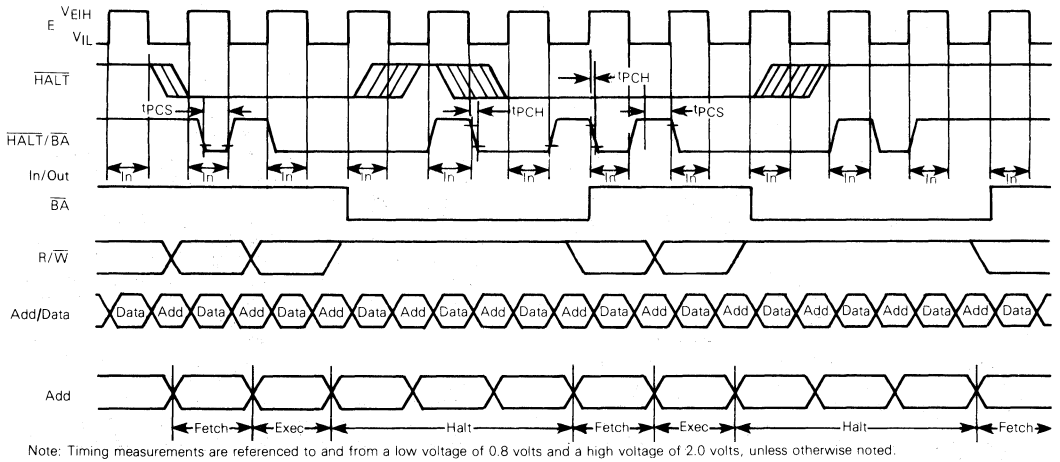


signals. Figure 22 shows the appropriate timing diagram for Halt/BA with the recommended circuit. The pullup resistor shown in the circuit maintains a high logic level when $\overline{HALT}$ is not active. During a positive half-cycle of E, pin 3 is an input sampled to determine if the Halt State is requested (active low). During the negative half cycle of E, the $\overline{BA}$ signal is output through pin 3. After the request for Halt State signal is detected and the processor completes its current instruction, the CPU is halted and the active low $\overline{BA}$ signal is output through pin 3 during the negative half cycle of E. The local bus is then available for other devices to utilize until the Halt State signal has returned to a high level, thus allowing the IPC back on the local bus. During the Halt State, the R/$\overline{W}$ is high, and the address bus displays the address of the next instruction.

When single instruction operation is desired, in program debug for instance, it is advantageous to single step through instructions. After $\overline{BA}$ goes low, $\overline{HALT}$ must be brought high for one E-cycle and returned low again to single step through instructions. Figure 22 illustrates the timing involved while single stepping through a single byte, two bus cycle instruction, such as CLRA.

$\overline{BA}$ is not output in response to the Wait instruction. If interrupts are to be utilized in removing the processor from a

FIGURE 22 — $\overline{HALT}/\overline{BA}$ TIMING DIAGRAM



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

Wait State while in the Halt/BA mode then, $\overline{IRQ1}$ and $\overline{IRQ2}$ are the only interrupts which may do so; therefore, their masks must be cleared before entering the Wait State.

MASKABLE INTERRUPT REQUEST 1 — $\overline{IRQ1}$

This level-sensitive input can be used to request an interrupt sequence. The IPC will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the IPC will begin an interrupt sequence: a vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is continued at the new location. This is explained in greater detail in the Interrupt Section.

$\overline{IRQ1}$ typically requires an external resistor (3K to 10K depending on external devices drive capability) to V_{CC} for wire-OR applications. $\overline{IRQ1}$ has no internal pullup resistor.

STROBE CONTROL 1 AND 2 — SC1 and SC2

The functions of SC1 and SC2 depend on the operating mode. SC1 is configured as an input in all modes except the Expanded Non-Multiplexed Mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

Single Chip Modes — In these modes, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as an input strobe ($\overline{IS3}$) and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{IS3}$ are controlled by the Control and Status Register for Port 3 and are discussed in the Port 3 description.

SC2 is configured as an output strobe ($\overline{OS3}$) and can be used to strobe output data or acknowledge input data for Port 3. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{OS3}$ timing is shown in Figure 6.

Expanded Non-Multiplexed Mode — In this mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select (IOS) and is asserted (active-low) only when addresses $0100 through $01FF are accessed. SC2 is configured as R/$\overline{W}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{W}$ and E are high.

Expanded Multiplexed Modes — In these modes, SC1 is configured as an input and SC2 is configured as an output. In the expanded multiplexed modes, the IPC has the ability to access a 64K byte address space. SC1 functions as an input, Address Strobe, which controls demultiplexing and enabling of the eight least significant addresses and the data buses.

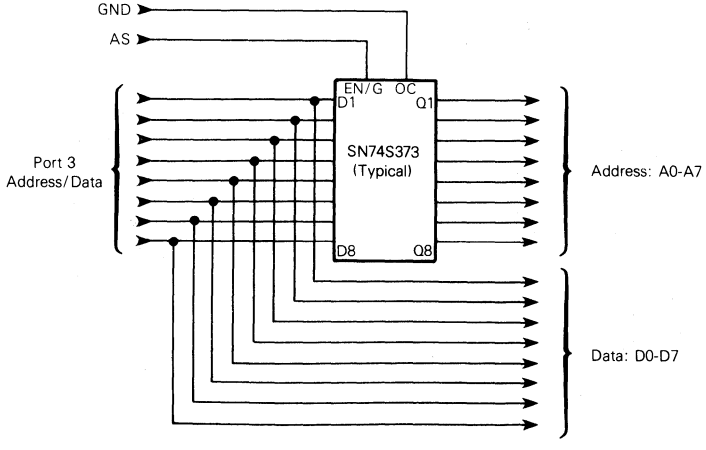
By using a transparent latch such as an SN74LS373 or MC6882, Address Strobe (AS) can also be used to demultiplex the two buses external to the IPC. (See Figure 23.) SC2 provides the local Data Bus control signal called Read/Write (R/$\overline{W}$). SC2 is configured as R/$\overline{W}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{W}$ and E are high.

SYSTEM BUS INTERFACE

Port 1 is a mode-independent 8-bit data port which permits the external system bus to access the dual-ported RAM and semaphore registers either asynchronously or synchronously with respect to the E clock. In addition to the eight data lines (SD0-SD7), eight address (SA0-SA7) and three control lines (SR/$\overline{W}$, $\overline{CS}$, $\overline{DTACK}$) are used to access the dual-ported RAM and semaphore registers.

Port 1 Data Lines (SD0-SD7) — These data lines are bidirectional data lines which allow data transfer between the dual-ported RAM or the semaphore registers, and the system bus. The data bus output drivers are three-state devices which remain in the high-impedance state except

FIGURE 23 — TYPICAL LATCH ARRANGEMENT



during a read of the IPC dual-ported RAM or semaphore registers by the system processor.

System Address Lines (SA0-SA7) — The address lines together with the Chip Select signal allow any of the 128 bytes of RAM or six semaphore registers to be uniquely selected from the system bus. The address lines must be valid before the $\overline{CS}$ signal goes low for the asynchronous interface and valid before the E signal goes high for the synchronous interface. The system interface must be deselected between reads or between writes for the asynchronous operation.

System Read/Write (SR/$\overline{W}$) — This signal is generated by the system bus to control the direction of data transfer on the data bus. With the IPC selected, a low on the SR/$\overline{W}$ line enables the input buffers, and data is transferred from the system processor to the IPC. When SR/$\overline{W}$ is high and the chip is selected, the data output buffers are turned on and data is transferred from the IPC to the system bus.

Chip Select ($\overline{CS}$) — This signal is a TTL compatible input signal, used to activate the system bus interface and allows transfer of data between the IPC and the system processor during synchronous or asynchronous accesses. $\overline{CS}$ provides the synchronizing signal for the Semaphore registers during access by the system bus.

Data Transfer Acknowledge ($\overline{DTACK}$) — This bidirectional control line is used to determine synchronous or asynchronous system bus accesses and to provide the data acknowledge signal for asynchronous data transfers.

As an input, it is sampled on the falling edge of $\overline{CS}$ by the IPC to determine if the system bus is being accessed synchronously or asynchronously with respect to the E clock.

If $\overline{DTACK}$ is low when sampled, the system bus is synchronous and data will be transferred during E high as shown in Figure 13.

If $\overline{DTACK}$ is high when sampled, the system bus is asynchronous. In this mode $\overline{DTACK}$ becomes an output that is asserted low when data is on the bus during a system read or when a data transfer is completed during a system write. Refer to Figures 9 through 12.

$\overline{DTACK}$ requires an external pullup resistor when the system bus is run asynchronously since it is then a bidirectional handshake line for information transfer on the system data bus.

PORT 2 — P20-P24

Port 2 is a mode independent 5-bit I/O port where each line is configured by its Data Direction Register. During reset, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, or CMOS devices using external pullup resistors. P20, P21 and P22 must always be connected to provide the operating mode.

PORT 2 DATA REGISTER

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

Inputs on P20, P21 and P22 determine the operating mode which is latched into the Program Control Register on the positive edge of $\overline{RESET}$. The mode may be read from the Port 2 Data Register (PC2 is latched from pin 45).

Port 2 also provides an interface for the Serial Communications Interface and Timer. Bit 1, if configured as an output, is dedicated to the Timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

PORT 3 — P30-P37

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

Single Chip Modes — In these modes, Port 3 is an 8-bit I/O port where each line is configured by the Port 3 Data Direction Register. Associated with Port 3 are two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control Port 3 data transfers.

Three Port 3 options, controlled by the Port 3 Control and Status Register and available only in the Single Chip Modes are: 1) Port 3 input data can be latched using $\overline{IS3}$ as a control signal, 2) $\overline{OS3}$ can be generated by either an IPC read or write to the Port 3 Data Register, and 3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 7.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 FLAG	IS3 IRQ1 ENABLE	X	OSS	LATCH ENABLE	X	X	X	$0F

Bits 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared by Reset.

Bit 4 OSS (Output Strobe Select). This bit determines whether $\overline{OS3}$ will be generated by a read or write of the Port 3 Data Register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared by Reset.

Bit 5 Not used.

Bit 6 $\overline{IS3}$-$\overline{IRQ1}$ ENABLE. When set, an $\overline{IRQ1}$ interrupt will be enabled whenever $\overline{IS3}$ FLAG is set; when clear, the interrupt is inhibited. This bit is cleared by Reset.

Bit 7 $\overline{IS3}$ FLAG. This read-only status bit is set by an $\overline{IS3}$ negative edge. It is cleared by a read of the Port 3 Control and Status Register (with $\overline{IS3}$ FLAG set) followed by a read or write to the Port 3 Data Register or by Reset.

Expanded Non-Multiplexed Mode — In this mode, Port 3 is configured as a bi-directional data bus (D0-D7). The direction of data transfers is controlled by R/$\overline{W}$ (SC2). Data transfers are clocked by E (Enable).

Expanded Multiplexed Modes — In these modes, Port 3 is configured as a time-multiplexed address (A0-A7) and data bus (D0-D7). Address Strobe (AS) must be input on SC1, and can be used externally to de-multiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent potential bus conflicts.

PORT 4 — P40-P47

Port 4 is configured as 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors.

Single Chip Modes — In these modes, Port 4 functions as an 8-bit I/O port where each line is configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Expanded Non-Multiplexed Mode — In this mode, Port 4 is configured from reset as an 8-bit input port, where the Data Direction Register can be written, to provide any or all of address lines A0-A7. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured.

Expanded Multiplexed Mode — In all these modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port; the Port 4 Data-Direction Register must be written to provide any or all of address lines, A8 to A15. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured (bit 0 controls A8, etc.).

3

OPERATING MODES

The IPC provides eight different operating modes which are selectable by hardware programming and referred to as Modes 0 through 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC1 and SC2 and the address location of the interrupt vectors.

FUNDAMENTAL MODES

The eight modes of the IPC can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single Chip includes Modes 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. A system utilizing three MC68120's, one in each of the fundamental operating modes, is shown in Figure 24. Table 6 summarizes the characteristics of the operating modes.

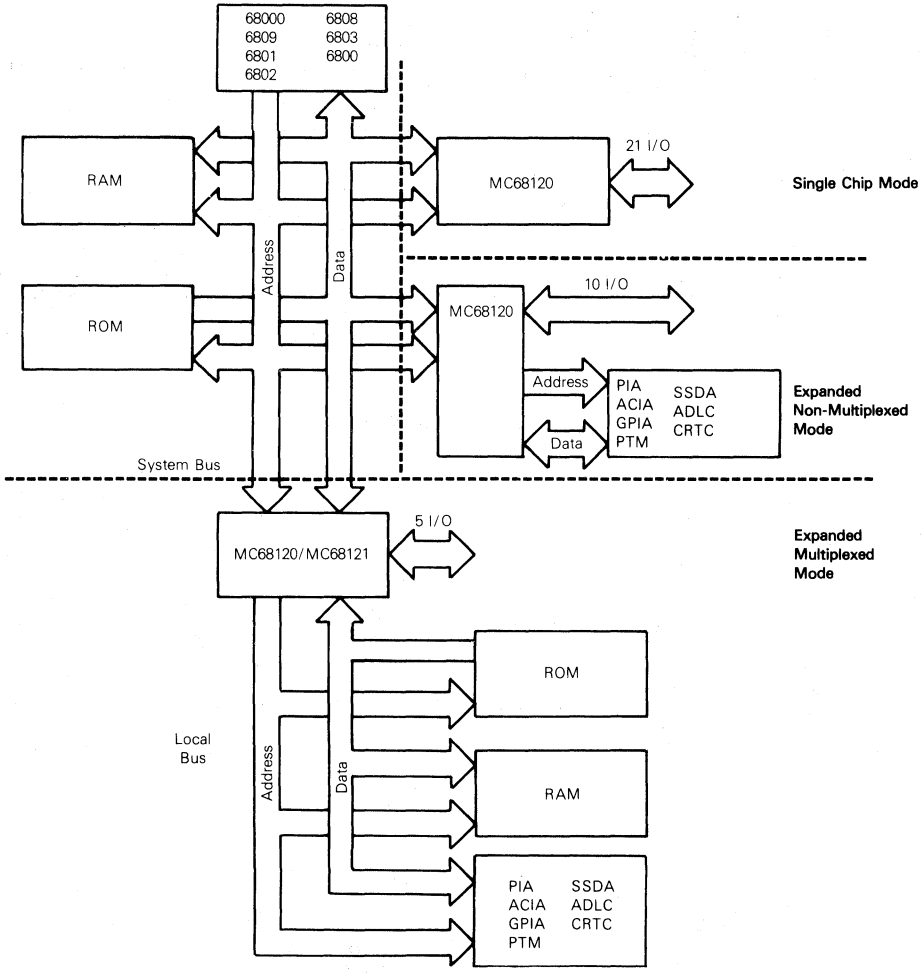
Single Chip Modes (4, 7) — In Single Chip Mode, three of the four IPC ports are configured as parallel input/output data ports, as shown in Figure 25. The IPC functions as a complete microcomputer in these two modes without external address or data buses. A maximum of 21 I/O lines and two Port 3 control lines are provided.

In Single Chip Test Mode (4), the RAM responds to addresses $XX80 (X = don't care) through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using Modes 0, 1, 2, or 6. If the IPC is reset and then programmed into Mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from Mode 4 without going through reset by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Port 3 and 4 in the Single Chip and Non-Multiplexed Modes.

TABLE 6 — SUMMARY OF IPC OPERATING MODES

Common to all Modes: System Bus Interface Reserved Register Area 6 Semaphore Registers I/O Port 2 Programmable Timer Serial Communications Interface 128 bytes of Dual Ported RAM	Expanded Multiplexed Modes Four Memory Space Options (64K Address Space): (1) MDOS Compatible (2) No ROM (3) External Vector Space (4) ROM with Partial Address Bus* External Memory Space Accessed Through: Port 3 as a Multiplexed Address/Data Bus Port 4 as an Address Bus (High) SC1 is Address Strobe Bus (AS) Input SC2 is Read/Write (R/$\overline{W}$)
Single Chip Mode* 2048 Bytes of ROM (Internal) Port 3 is a Parallel I/O Port with Two Control Lines Port 4 is a Parallel I/O Port SC1 is Input Strobe 3 ($\overline{IS3}$) SC2 is Output Strobe 3 ($\overline{OS3}$)	Test Modes Expanded Multiplexed Test Mode May be Used to Test RAM and ROM* Single Chip and Non-Multiplexed Test Mode* May be Used to Test Ports 3 and 4 as I/O Ports
Expanded Non-Multiplexed Mode* 2048 Bytes of ROM (Internal) 256 Bytes of External Memory Space Port 3 is an 8-Bit Data Bus Port 4 is an Address Bus SC1 is Input/Output Select ($\overline{IOS}$) SC2 is Read/Write (R/$\overline{W}$)	*MC68120 only

FIGURE 24 — IPC FUNDAMENTAL OPERATING MODES



Expanded Non-Multiplexed Mode (5) — A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bi-directional data bus and Port 4 is configured as an input data port. Any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Internal pullup resistors are provided to pull Port 4 lines high until it is configured.

Figure 26 illustrates the external resources available in the Expanded Non-Multiplexed Mode. The IPC interfaces directly with M6800 Family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and may be used as an address or chip select line.

Expanded-Multiplexed Modes (0, 1, 2, 3, 6) — In the Expanded Multiplexed Modes, the IPC has the ability to access a 64K-byte memory space. Port 3 functions as a time-multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS) and the data bus valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8-A15. However, in Mode 6, Port 4 can provide any subset of A8 to A15 while retaining the remainder as input lines. Writing 1's to the desired bits in the Data Direction Register (DDR) will output the corresponding address lines while the remaining bits will remain inputs (as configured from reset or from 0's written to the DDR). Internal pullup resistors are provided to pull Port 4 lines high until software configures the port. Initialization of Port 4 in Mode six must be done to obtain any upper address lines externally.

FIGURE 25 — SINGLE CHIP MODE

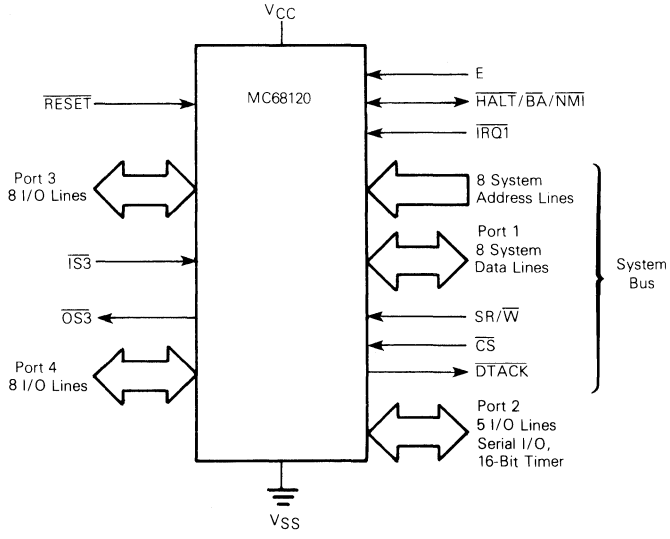


FIGURE 26 — EXPANDED NON-MULTIPLEXED MODE

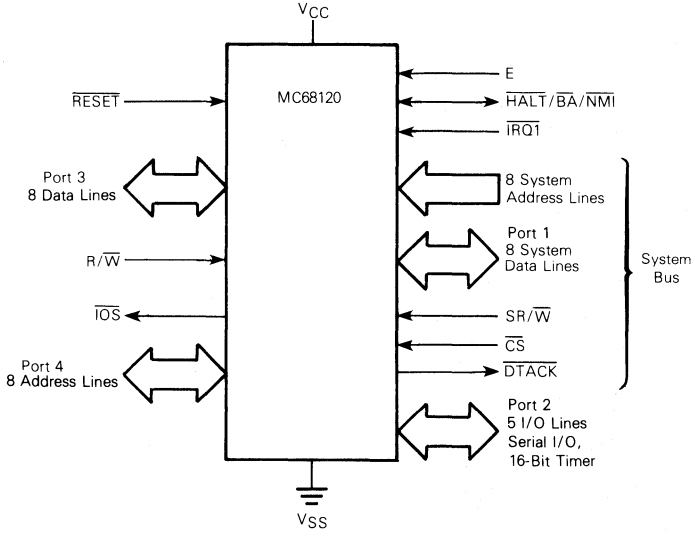
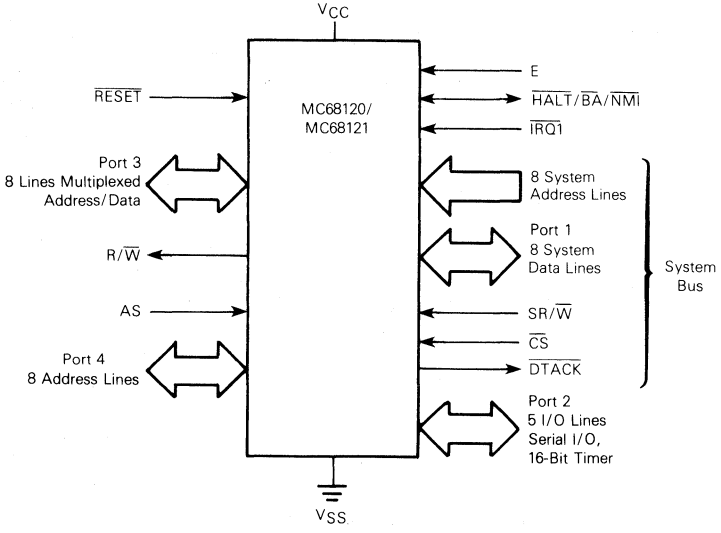


Figure 27 depicts the external resources available in the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 23. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the reset vector is external at $BFFE and $BFFF after the positive edge of $\overline{RESET}$. In addition, the internal and external data buses are connected together so there must be no memory map overlap (to avoid potential bus conflicts). Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with automated test equipment.

FIGURE 27 — EXPANDED MULTIPLEXED MODE



MODE PROGRAMMING

The operating mode is programmed by the levels asserted on P22, P21, and P20 during the positive edge of $\overline{RESET}$. These are latched into PC2, PC1, and PC0 of the program control register. The operating mode may be read from the Port 2 Data Register and programming levels and timing must be met as shown in Figure 28 and Table 7. Any mode may be entered from either Mode 0 or Mode 4 without going through reset by writing the appropriate bits to the port 2 data register. A brief outline of the operating modes is shown in Table 8.

Circuitry to provide the programming levels is primarily dependent on the normal system use of the three pins. If configured as outputs, the circuit shown in Figure 29 may be used; otherwise, the three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The IPC provides up to 64K bytes of address space depending upon the operating mode. A memory map for each operating mode is shown in Figure 30. In Modes 1R and 6R, the "R" means the ROM has been relocated by a mask option. The first 32 locations of each map are reserved for the IPC internal register area, as shown in Table 9, with exceptions as indicated.

FIGURE 28 — MODE PROGRAMMING TIMING

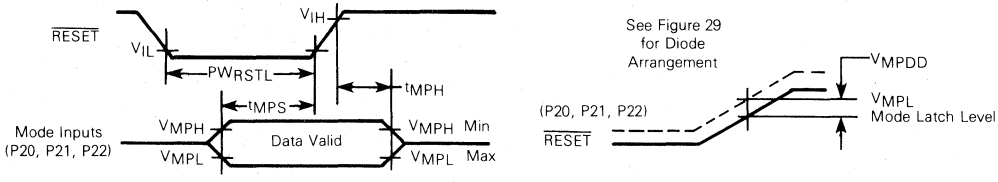


TABLE 7 — MODE PROGRAMMING SPECIFICATIONS (See Figure 30)

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

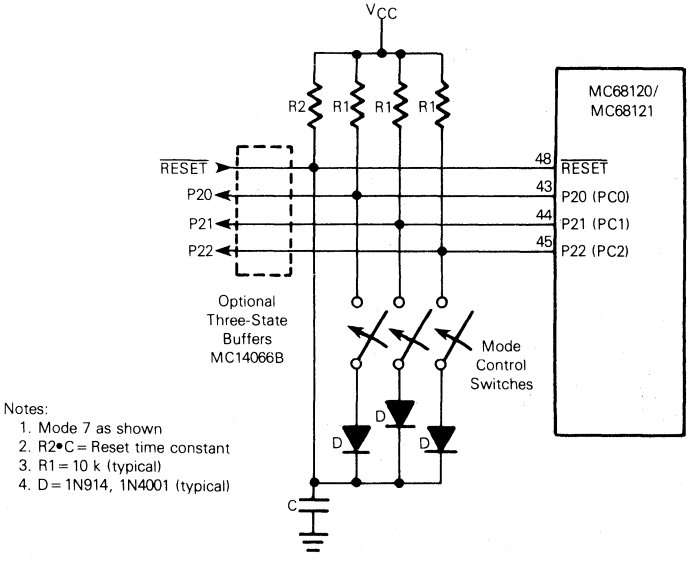
TABLE 8 — MODE SELECTION SUMMARY

Mode	Pin 45 P22 PC2	Pin 44 P21 PC1	Pin 43 P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode(5)
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode(5)
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	I(7)	E	MUX(4)	Multiplexed/RAM(4)
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM(4)
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and ROM(4)
0	L	L	L	I	I	E(3)	MUX(4)	Multiplexed Test(4)

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

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) Interrupt vectors externally located at $BFF0-$BFFF
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register
(7) Internal RAM and registers located at $C0XX (for use with MDOS)

FIGURE 29 — TYPICAL MODE PROGRAMMING CIRCUIT

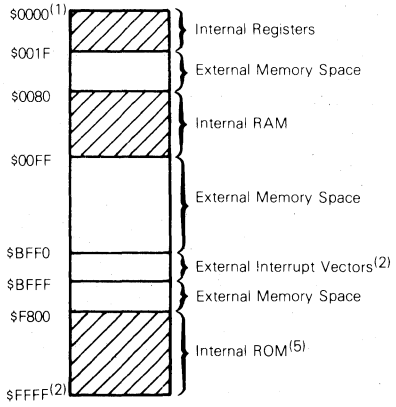


Notes:
1. Mode 7 as shown
2. R2•C = Reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)

FIGURE 30 — IPC MEMORY MAPS

Multiplexed Test Mode

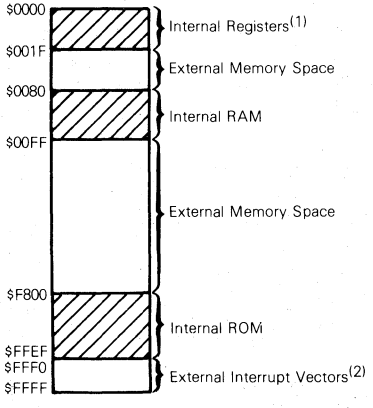
MC68120 Mode **0**



Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) The interrupt vectors are externally located at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.
5) MC68120 only.

MC68120 Mode **1**

Multiplexed/RAM and ROM

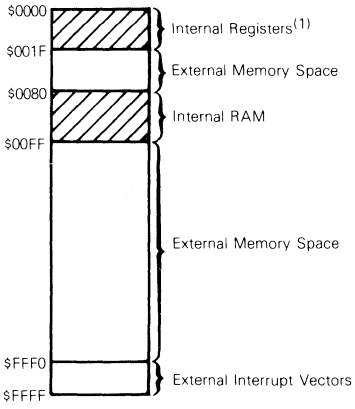


Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

FIGURE 30 — IPC MEMORY MAPS (CONTINUED)

MC68120/ MC68121 Mode **2**

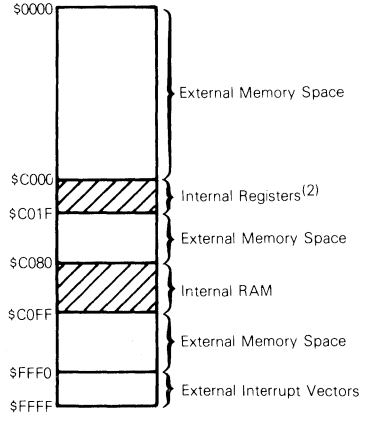
Multiplexed/RAM



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

MC68120/ MC68121 Mode **3**

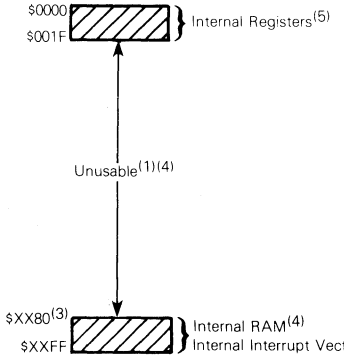
Multiplexed/RAM, MDOS Compatible (1)



Notes:
1) Relocating the internal registers and the internal RAM to high memory allows processor to run MDOS.
2) Excludes the following addresses which may be used externally: $C004, $C005, $C006, $C007, and $C00F.

MC68120 Mode **4**

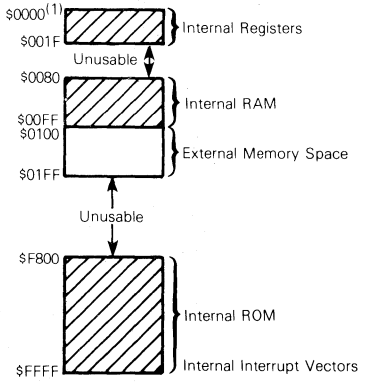
Single Chip Test(2)



Notes:
1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a "1" into bit 5 (PCO) of Port 2 Data Register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.
5) MPU Read of Port 3 Data Direction Register will access Port 3 Data Register instead.

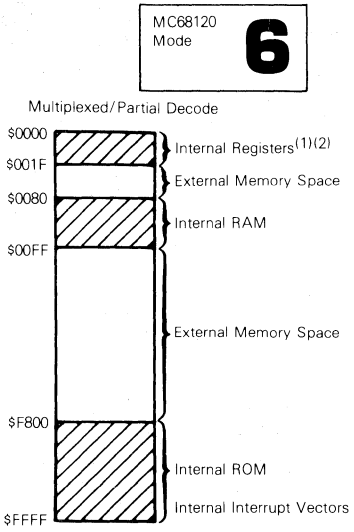
MC68120 Mode **5**

Non-Multiplexed/Partial Decode (2) (3)

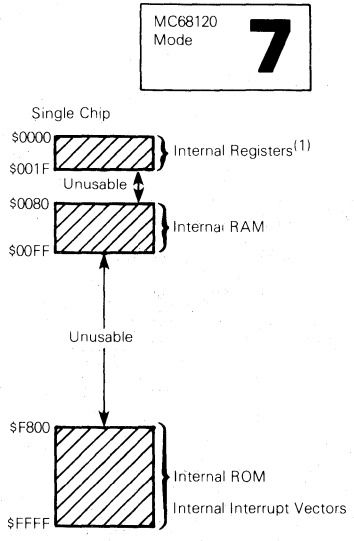


Notes:
1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{\text{IOS}}$).
2) This mode may be entered without going through Reset by using Mode 4 and subsequently writing a "1" into bit 5 (PCO) of Port 2 Data Register.
3) Address lines A0 to A7 will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.

FIGURE 30 — IPC MEMORY MAPS (CONCLUDED)



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



Notes:
1) MPU reads of Port 3's Data Direction Register will access Port 3's Data Register instead.

TABLE 9 — INTERNAL REGISTER AREA

Register	Address**** (Hexadecimal)	Register	Address**** (Hexadecimal)
Reserved	00	SCI Rate and Mode Control Register	10
Port 2 Data Direction Register***	01	Transmit/Receive Control and Status Register	11
Reserved	02	SCI Receive Data Register	12
Port 2 Data Register	03	SCI Transmit Data Register	13
Port 3 Data Direction Register***	04*		
Port 4 Data Direction Register***	05**	Function Control Register	14
Port 3 Data Register	06*	Counter Alternate Address (High Byte)	15
Port 4 Data Register	07**	Counter Alternate Address (Low Byte)	16
Timer Control and Status Register	08	Semaphore 1	17
Counter (High Byte)	09	Semaphore 2	18
Counter (Low Byte)	0A	Semaphore 3	19
Output Compare Register (High Byte)	0B	Semaphore 4	1A
Output Compare Register (Low Byte)	0C	Semaphore 5	1B
Input Capture Register (High Byte)	0D	Semaphore 6	1C
Input Capture Register (Low Byte)	0E	Reserved	1D-1F
Port 3 Control and Status Register	0F*		

*These external addresses in Modes 0, 1, 2, 3, 5, 6 cannot be accessed in Mode 5 (no IOS).

**These are external addresses in Modes 0, 1, 2, 3.

***1 = Output, 0 = Input

****These addresses relocated at $C000-$C01F in Mode 3.

INTERRUPTS

The IPC supports two types of interrupt requests: Maskable and Non-Maskable. A Non-Maskable Interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ1}$ and $\overline{IRQ2}$. The Programmable Timer and Serial Communications Interface use an internal $\overline{IRQ2}$ interrupt line, as shown in the block diagram of the IPC. External devices (and $\overline{IS3}$) use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before an $\overline{IRQ2}$ interrupt if both are pending.

All $\overline{IRQ2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order where each is vectored to a separate location. All IPC vector locations are shown in Table 10, from highest (top) to lowest (bottom) priority.

The interrupt flowchart is depicted in Figure 31. The Program Counter, Index Register, Accumulator A, Accumulator B, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. The general interrupt timing sequence is shown in Figure 32. The Interrupt $\overline{HALT}/\overline{BA}$ timing is illustrated in Figure 21 and 22.

TABLE 10 — MCU VECTOR LOCATIONS *

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

*These locations are relocated at $BFF0-$BFFF in Mode 0.

**Highest priority.

FIGURE 31 — INTERRUPT FLOWCHART

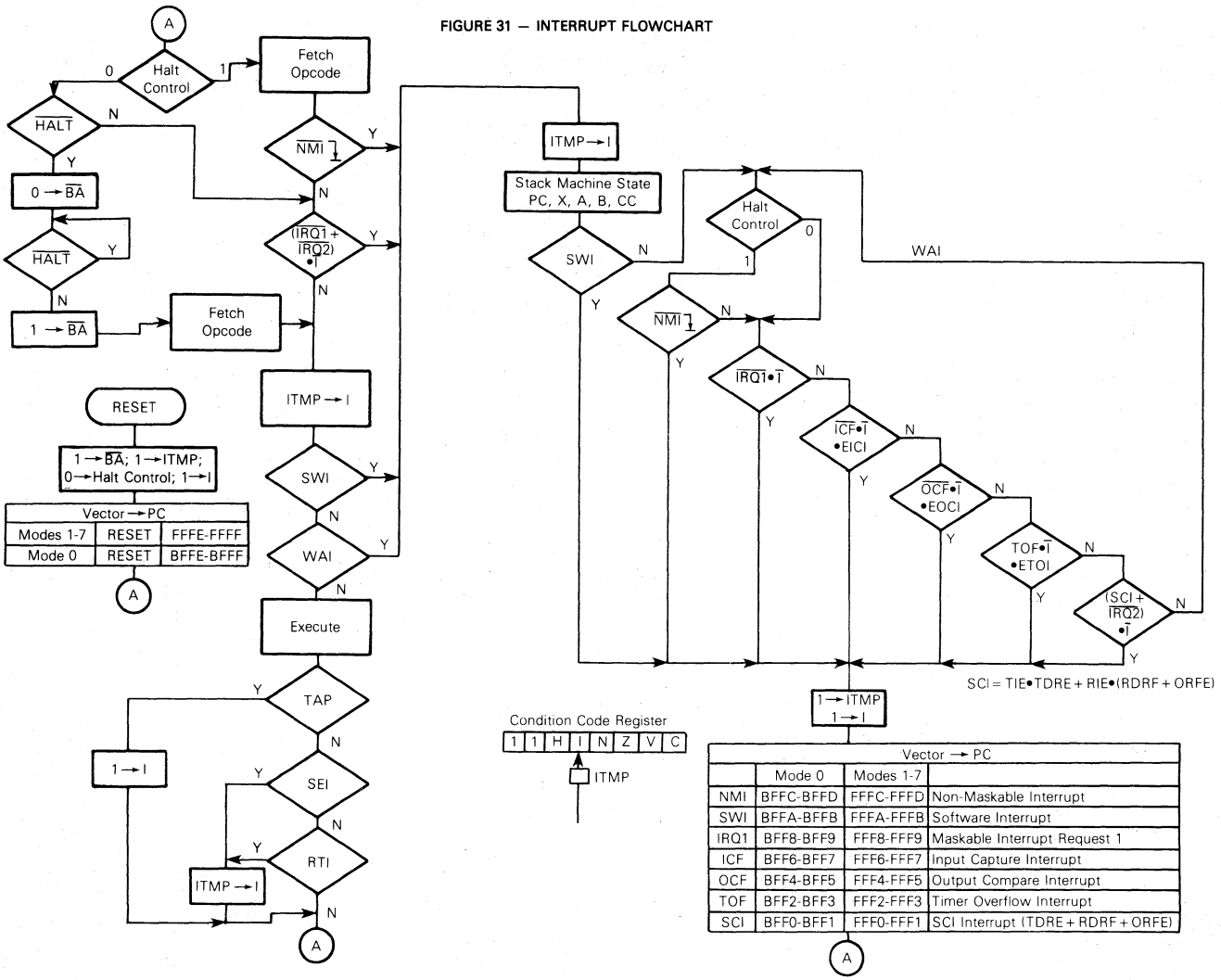
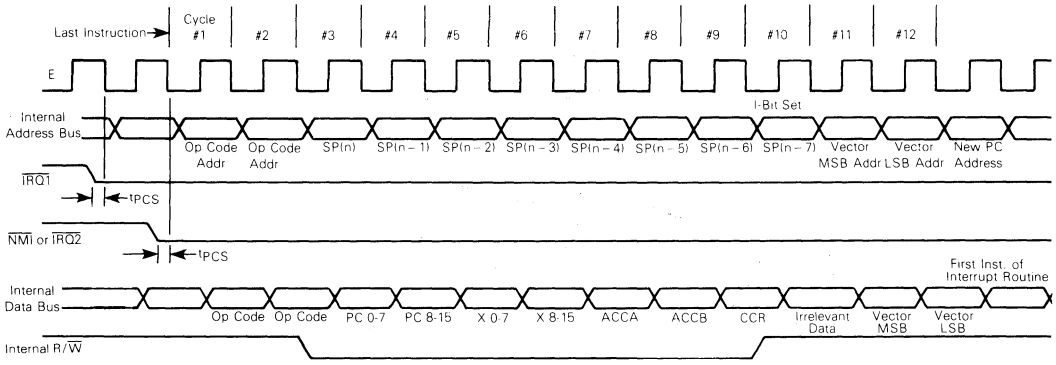


FIGURE 32 — INTERRUPT SEQUENCE



3

PROGRAMMABLE TIMER

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 33.

TIMER CONTROL AND STATUS REGISTER ($08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable while bits 0-4 can be written. The three most significant bits provide the timer status and they indicate:

- a proper level transition has been detected, or
- a match has been found between the free-running counter and the output compare register, or
- the free-running counter has overflowed.

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

TIMER CONTROL AND STATUS REGISTER (TSCR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$08

Bit 0 OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared by reset.

Bit 1 IEDG Input Edge. IEDG is cleared by reset and controls which level transition will trigger a counter transfer to the Input Capture Register:

IEDG = 0 Transfer on a negative edge
IEDG = 1 Transfer on a positive edge

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

Bit 3 EOCI Enable Output Compare Interrupt. When set, an $\overline{IRQ2}$ interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 4 EICI Enable Input Capture Interrupt. When set, an $\overline{IRQ2}$ interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 5 TOF Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) followed by reading the highest byte of the counter ($09), or by reset. Reading the counter at $15 will not clear TOF.

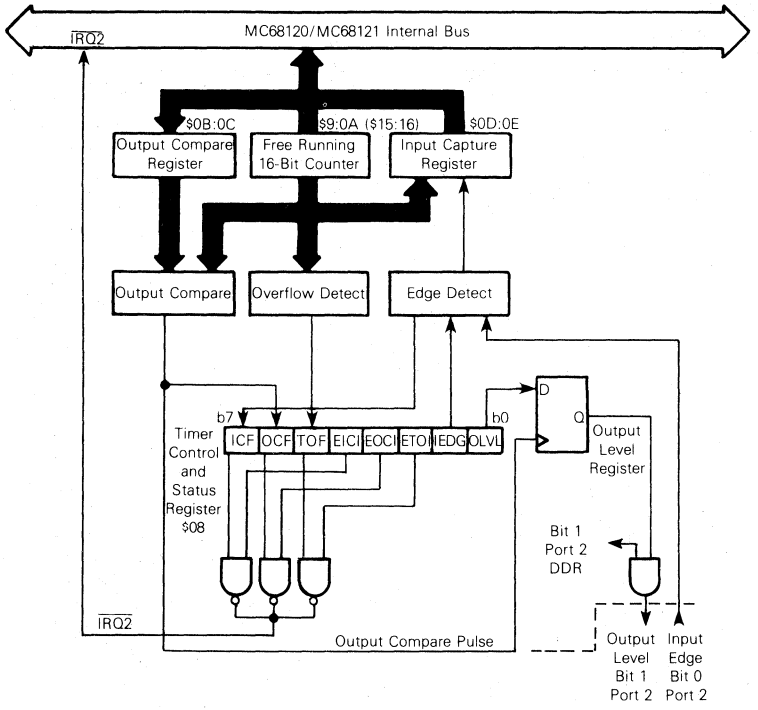
Bit 6 OCF Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register ($0B or $0C), or by reset.

Bit 7 ICF Input Capture Flag. ICF is set to indicate a proper level transition. It is cleared by reading the TCSR (with ICF set) and then reading the Input Capture Register High Byte ($0D), or by reset.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during reset and is a read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all 1's. The counter may also be read at location $15 and $16 to avoid the clearing of the TOF.

FIGURE 33 — PROGRAMMABLE TIMER — BLOCK DIAGRAM



OUTPUT COMPARE REGISTER ($0B:0C)

The Output Compare Register is a 16-bit Read/Write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E-cycle. When a match is found, OCF is set and OLVL is clocked to an output level register. If Port 2, bit 1 is configured as an output, OLVL will appear at P21. The Output Compare Register and OLVL can then be changed for the next compare. The compare function is inhibited for one cycle after a write to the high byte of the counter ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF by reset.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20, even when configured as an output. An input capture can occur independently of ICF: the input capture register always contains the most current value regardless of whether ICF was previously set or not. Counter transfer is inhibited, however, between accesses of a double byte IPC read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with two data formats and a choice of Baud rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Bi-phase. Both formats provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to allow uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until the data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or by reset. Software must provide the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

- format: standard mark/space (NRZ) or Bi-phase

- clock: external or internal clock source
- Baud rate: one of four per E-clock frequency, or one-eighth of the external clock input to P22
- wake-up features: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in Figure 34. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and read-only Receive Register. The shift registers are not accessible by software.

Rate and Mode Control Register ($10) — The Rate and Mode Control Register (RMCR) controls the SCI Baud rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared by reset. The two least significant bits control the Baud rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

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

Bit 1: Bit 0 SS1:SS0 Speed Select. These two bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the IPC input frequency (E). Table 11 lists bit times and rates for three selected IPC frequencies.

Bit 3: Bit 2 CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the Data Direction Register (DDR) value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 12 defines the format, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired Baud rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0 = 10, the internal Baud rate clock is provided at P22 regardless of the values for TE or RE.

NOTE: The source of SCI internal baud rate clock is the free-running counter of the timer. An IPC write to the counter can disturb serial operations.

FIGURE 34 — SCI REGISTERS

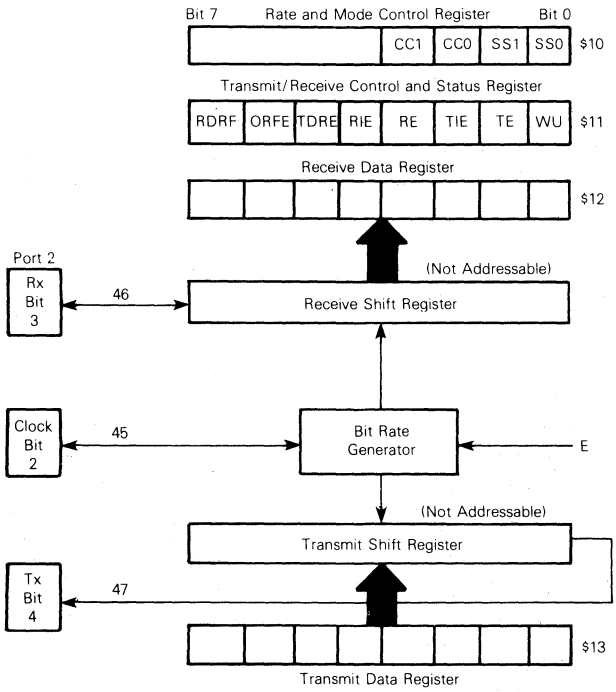


TABLE 11 — SCI BIT TIMES AND RATES

SS1:SS0	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	÷16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	÷128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	÷1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	÷4096	6.67ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

TABLE 12 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
0 0	Bi-Phase	Internal	Not Used
0 1	NRZ	Internal	Not Used
1 0	NRZ	Internal	Output
1 1	NRZ	External	Input

Transmit/Receive Control and Status Register ($11) — The Transmit/Receive Control and Status Register (TRCSR) controls the transmitter, receiver, wake-up features, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while only bits 0 to 4 are writable. The register is initialized to $20 by reset.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

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

Bit 0 WU — "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by reset. WU will not set if the line is idle.

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

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

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

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

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

Bit 6 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun occurs when a new byte is ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. ORFE is cleared by reading the TRCSR (with ORFE set) then reading the Receive Data Register, or by reset.

Bit 7 RDRF — Receive Data Register Full. RDRF is set when the contents of the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then reading the Receive Data Register, or by reset.

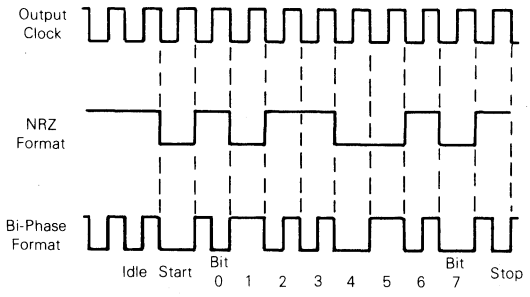
SERIAL OPERATIONS

The SCI is initialized by writing the control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the Transmit Shift Register is connected to P24 and serial output is initiated by the transmission of a 9-bit preamble of 1's.

At this point one of two situations exist: 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit Data Register (TDRE = 0), the byte will be transferred to the Transmit Shift Register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. In Bi-phase format, the output toggles at the start of each bit and at half time when a "1" is sent. SCI data formats are illustrated in Figure 35. In receiving Bi-phase, a "1" is input when two transitions occur in less than 3/4 bit-time, and a "0" is input when more than 3/4 bit-time passes after a transition on P23.

FIGURE 35 — SCI DATA FORMATS



Data: 01001101 ($4D)

INSTRUCTION SET

The MC68120/MC68121 is upward source and object code compatible with the MC6800 processor and directly compatible with the M6801 Family processors.

PROGRAMMING MODEL

A programming model for the MC68120/MC68121 is shown in Figure 14. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

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

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

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

Accumulators — The IPC contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

Condition Code Register — The Condition Code Register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits b6 and b7, are read as ones.

ADDRESSING MODES

The MC68120/MC68121 provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 13, 14, 15 and 16 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 17. With an input frequency (E) of 1 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 18 and a description of selected instructions is shown in Figure 38.

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

Direct Addressing — The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access (refer to Table 1). In most applications, this 256-byte area is reserved for frequently referenced data. Note that no direct addressing of internal control registers is possible in Mode 3.

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

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

Inherent Addressing — The operand(s) are registers and no memory reference is required. These are single byte instructions.

Relative Addressing — Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of −126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

TABLE 13 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

TABLE 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	•
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A - 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B - 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•

TABLE 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

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

The Condition Code Register notes are listed after table 16.

TABLE 15 — JUMP AND BRANCH INSTRUCTIONS

Operations	Mnemonic	Direct OP	Direct ~	Direct #	Relative OP	Relative ~	Relative #	Index OP	Index ~	Index #	Extnd OP	Extnd ~	Extnd #	Inherent OP	Inherent ~	Inherent #	Branch Test	Cond. Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	●	●	●	●	●	●
Branch Never	BRN				21	3	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC				24	3	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS				25	3	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ				27	3	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE				2C	3	2										N⊕V = 0	●	●	●	●	●	●
Branch If > Zero	BGT				2E	3	2										Z + (N⊕V) = 0	●	●	●	●	●	●
Branch If Higher	BHI				22	3	2										C + Z = 0	●	●	●	●	●	●
Branch If Higher or Same	BHS				24	3	2										C = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE				2F	3	2										Z + (N⊕V) = 1	●	●	●	●	●	●
Branch If Carry Set	BLO				25	3	2										C = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT				2D	3	2										N⊕V = 1	●	●	●	●	●	●
Branch If Minus	BMI				2B	3	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC				28	3	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS				29	3	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL				2A	3	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR				8D	6	2										See Special Operations — Figure 36	●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3					●	●	●	●	●	●
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1		●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1	See Special Operations — Figure 36	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 16 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

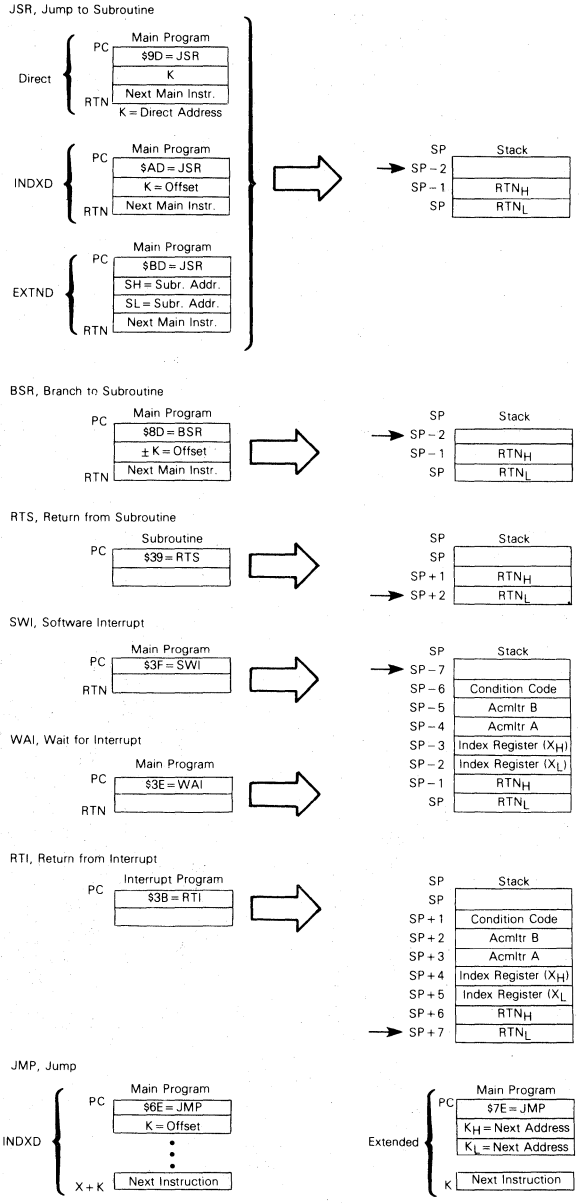
H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

TABLE 17 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

FIGURE 36 — SPECIAL OPERATIONS



Legend:

RTN = Address of next instruction in Main Program to be executed upon return from subroutine

RTN_H = Most significant byte of Return Address

RTN_L = Least significant byte of Return Address

→ = Stack pointer after execution

K = 8-bit unsigned value

CYCLE-BY-CYCLE OPERATION SUMMARY

Table 18 provides a detailed description of the information present on the Address Bus, Data Bus, and the R/$\overline{W}$ line during cycle of each instructions.

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

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

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 19. There are 220 valid machine codes, 34 unassigned codes and 2 reserved for test purposes.

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 1 of 5)

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

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 2 of 5)

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

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 3 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS LDX LDD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS STX STD	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR ASR NEG CLR ROL COM ROR DEC TST (1) INC	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Current Operand Data
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Index Register Plus Offset	0	New Operand Data
CPX SUBD ADDD	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 4 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address +1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Irrelevent Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address +1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address +1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address +1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer -1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer -1 Stack Pointer -2 Stack Pointer -3 Stack Pointer -4 Stack Pointer -5 Stack Pointer -6	1 1 0 0 0 0 0 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Cond. Code Register

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 5 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer +2	1	Contents of Accumulator B from Stack
		6	Stack Pointer +3	1	Contents of Accumulator A from Stack
		7	Stack Pointer +4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer +5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer +6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer +7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer -1	0	Return Address (High Order Byte)
		5	Stack Pointer -2	0	Index Register (Low Order Byte)
		6	Stack Pointer -3	0	Index Register (High Order Byte)
		7	Stack Pointer -4	0	Contents of Accumulator A
		8	Stack Pointer -5	0	Contents of Accumulator B
		9	Stack Pointer -6	0	Contents of Cond. Code Register
		10	Stack Pointer -7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer -1	0	Return Address (High Order Byte)

TABLE 19 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	*				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	*				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	*				37	PSHB		3	1	6B	*				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	*				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	*				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	*				75	*				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	*				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	*				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	*				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	*				47	ASRA		2	1	7B	*				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	*				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	*				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	*				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	*				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	*				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	*				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	*				51	*				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	*				52	*				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	*				53	COMB		2	1	87	*				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	*				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	*				8F	*				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	*				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	*				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	*				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	*				99	ADCA		3	2	CD	*								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB	INHER	4	1	67	ASR	INDXD	6	2	9B	ADDA	DIR	3	2	CF	*								

*UNDEFINED OP CODE

NOTES:

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

APPENDIX A
MC68120 CUSTOM ORDERING INFORMATION

A.0

Address $FFEF is Reserved for the Checksum value for the ROM, to be generated at the factory.

A.1 CUSTOM MC68120 ORDERING INFORMATION

The custom MC68120 specifications may be transmitted to Motorola in any of the following media:

A) EPROM(s)

B) MDOS diskette

The specification should be formatted and packaged, as indicated in the appropriate paragraph below, and mailed prepaid and insured with a cover letter (see Figure A-1) to:

Motorola Inc.
MPU Marketing
3501 Ed Bluestein Blvd.
Austin, Texas 78721

A copy of the cover letter should also be mailed separately.

A.2 EPROMs

MCM2708 and MCM2716 type EPROMs, programmed with the custom program (positive logic notation for address and data), may be submitted for pattern generation. The MC2708s must be clearly marked to indicate which PROM corresponds to which address space ($F800-$FBFF; $FC00-$FFFF). See Figure A-2 for recommended marking procedure.

FIGURE A-2

XXX = Customer ID

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

A.3 MDOS DISKETTE

The file name and start/end location should be written on the label.

FIGURE A-1

CUSTOMER NAME _____

ADDRESS _____

STATE _____ CITY _____ ZIP _____

PHONE _____ EXTENSION _____

CONTACT MS/MR _____

CUSTOMER PART # _____

PATTERN MEDIA
☐ 2708 EPROM
☐ 2716 EPROM
☐ Diskette (MDOS)

TEMPERATURE RANGE
☐ 0° to 70°C

PACKAGE TYPE
☐ Ceramic

MARKING
☐ Standard
☐ Special

(Note 1) _____

NOTE: (1) Other Media Require Prior Factory Approval

SIGNATURE _____

TITLE _____

MOTOROLA

MC68701

Advance Information

MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

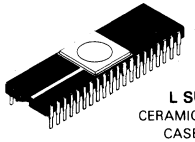
MICROCOMPUTER WITH EPROM

3

MC68701 MICROCOMPUTER UNIT (MCU)

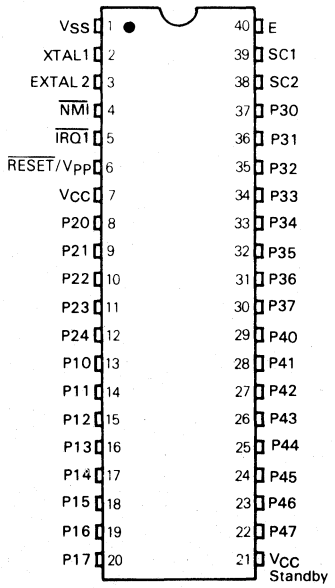
The MC68701 is an 8-bit single chip microcomputer unit (MCU) which significantly enhances the capabilities of the M6800 family of parts. It can be used in production systems to allow for easy firmware changes with minimum delay or it can be used to emulate the MC6801/03 for software development. It includes an upgraded M6800 microprocessor unit (MPU) with upward source and object code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned multiply. The MCU can function as a monolithic microcomputer or can be expanded to a 64K byte address space. It is TTL compatible and requires one +5 volt power supply for nonprogramming operation. An additional V_{PP} power supply is needed for EPROM programming. On-chip resources include 2048 bytes of EPROM, 128 bytes of RAM, Serial Communications Interface (SCI), parallel I/O, and a three function Programmable Timer. A summary of MCU features includes:

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



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT

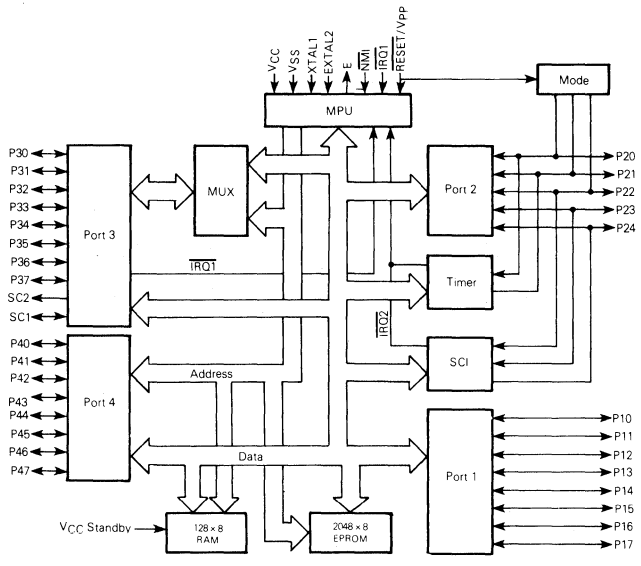


GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic	1.0	0°C to 70°C	MC68701L
L Suffix	1.0	−40°C to 85°C	MC68701CL
	1.25	0°C to 70°C	MC68701L-1
	1.25	−40°C to 85°C	MC68701CL-1
	1.5	0°C to 70°C	MC68A701L
	2.0	0°C to 70°C	MC68B701L

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

MC68701 MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68701 MC68701C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	0 to 85	°C

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

THERMAL CHARACTERISTICS

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

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic	Symbol	MC68701		MC68701-1		MC68A701		MC68B701		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.5	1.0	0.5	1.25	0.5	1.5	0.5	2.0	MHz
Crystal Frequency	f_{XTAL}	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
External Oscillator Frequency	$4f_o$	2.0	4.0	2.0	5.0	2.0	6.0	2.0	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	–	100	–	100	–	100	–	100	ms
Processor Control Setup Time	t_{PCS}	200	–	170	–	140	–	110	–	ns

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

Characteristic		Symbol	MC68701			MC68701C			Unit
			Min	Typ	Max	Min	Typ	Max	
Input High Voltage	$\overline{RESET}$	V_{IH}	$V_{SS}+4.0$	–	V_{CC}	$V_{SS}+4.0$	–	V_{CC}	V
	Other Inputs*		$V_{SS}+2.0$	–	V_{CC}	$V_{SS}+2.2$	–	V_{CC}	
Input Low Voltage	$\overline{RESET}$	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.4$	$V_{SS}-0.3$	–	$V_{SS}+0.4$	V
	Other Inputs*		$V_{SS}-0.3$	–	$V_{SS}+0.8$	$V_{SS}-0.3$	–	$V_{SS}+0.8$	
Input Current, See Note (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	–	0.6	–	–	1.0	mA
	SCI		–	–	1.0	–	–	1.6	
Input Current (V_{in} = 0 to 5.25 V)	$\overline{NMI}$, $\overline{IRQ1}$	I_{in}	–	1.5	2.5	–	1.5	5	μA
Input Current (V_{in} = 0 to 0.4 V)	$\overline{RESET}/V_{PP}$	I_{in}	–	−2.0	–	–	−2.0	–	mA
(V_{in} = 4.0 V to V_{CC})			–	–	8.0	–	–	8.0	
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	Ports 1, 2, and 3	I_{TSI}	–	2	10	–	2	20	μA
Output High Voltage (I_{Load} = −65 μA, V_{CC} = Min)	Port 4, SC1, SC2	V_{OH}	$V_{SS}+2.4$	–	–	$V_{SS}+2.4$	–	–	V
(I_{Load} = −100 μA, V_{CC} = Min)	Other Outputs		$V_{SS}+2.4$	–	–	$V_{SS}+2.4$	–	–	
Output Low Voltage I_{Load} = 2.0 mA, V_{CC} = Min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	–	–	$V_{SS}+0.6$	V
Darlington Drive Current (V_O = 1.5 V)	Port 1	I_{OH}	1.0	2.5	10.0	1.0	2.5	10.0	mA
Internal Power Dissipation (Measured at $T_A = T_L$ in Steady-State Operation)		P_{INT}	–	–	1500	–	–	1500	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1 MHz)	Port 3, Port 4, SCI	C_{in}	–	–	12.5	–	–	12.5	pF
	Other Inputs		–	–	10.0	–	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}	4.0	–	5.25	4.0	–	5.25	V
	Powerup	V_{SB}	4.75	–	5.25	4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	–	–	6.0	–	–	8.0	mA
Programming Time Per Byte (T_A = 25°C)		t_{PP}	25	–	50	25	–	50	ms
Programming Voltage (T_A = 25°C)		V_{PP}	20.0	21.0	22.0	20.0	21.0	22.0	V
Programming Current ($V_{\overline{RESET}} = V_{PP}$, T_A = 25°C)		I_{PP}	–	30	50	–	30	50	mA

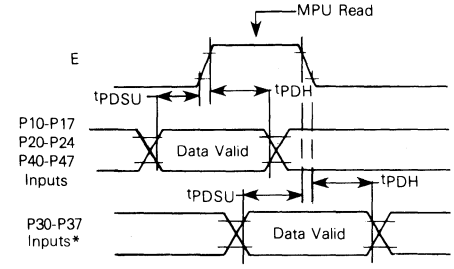
*Except mode programming levels; see Figure 15.

NOTE: $\overline{RESET}/V_{PP}$ I_{in} differs from MC6801 and MC6803 values.

PERIPHERAL PORT TIMING (Refer to Figures 3-6)

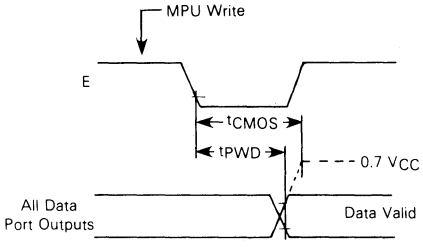
Characteristics	Symbol	MC68701		MC68701-1		MC68A701		MC68B701		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
Peripheral Data Setup Time	t_{PDSU}	200	–	200	–	150	–	100	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	200	–	150	–	100	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	t_{PWD}	–	350	–	350	–	300	–	250	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	–	2.0	–	2.0	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	200	–	150	–	100	–	ns
Input Data Hold Time	t_{IH}	50	–	50	–	40	–	30	–	ns
Input Data Setup Time	t_{IS}	20	–	20	–	20	–	20	–	ns

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



* Port 3 Non-Latched Operation (LATCHE ENABLE = 0)

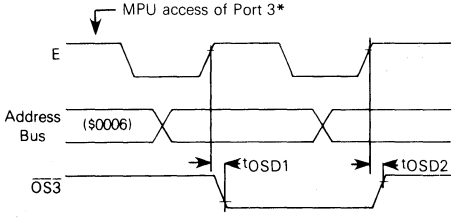
FIGURE 2 — DATA SETUP AND HOLD TIMES
(MPU WRITE)



NOTES:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

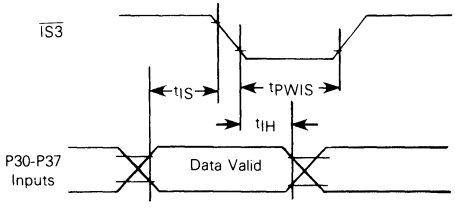
3

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



* Access matches Output Strobe Select (OSS = 0, a read;
OSS = 1, a write)

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



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

FIGURE 5 — CMOS LOAD

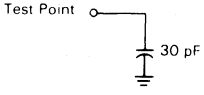
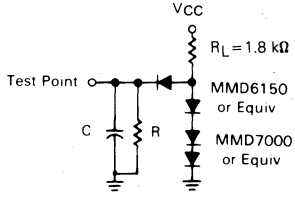


FIGURE 6 — TIMING TEST LOAD PORTS 1, 2, 3, 4



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

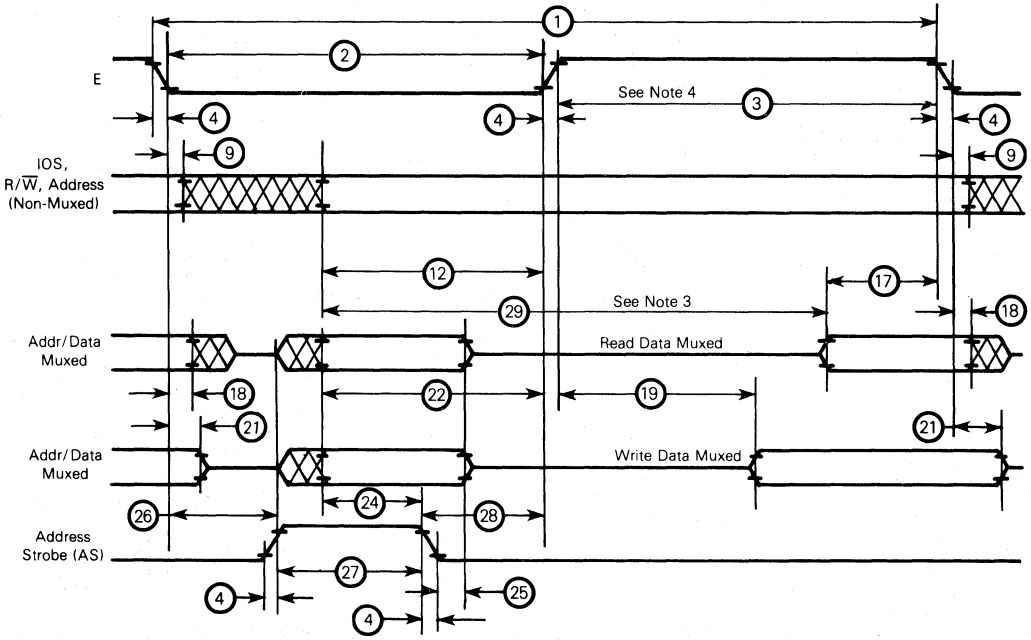
BUS TIMING (See Notes 2 and 3)

Ident Number	Characteristic	Symbol	MC68701		MC68701-1		MC68A701		MC68B701		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	–	2.0	0.5	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	300	1000	210	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	300	1000	220	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	10	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	115	–	70	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	60	–	40	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	–	170	–	120	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	20	–	10	–	ns
22	Multiplexed Address Valid Time to E Rise*	t_{AVM}	200	–	150	–	115	–	80	–	ns
24	Multiplexed Address Valid Time to AS Fall*	t_{ASL}	60	–	50	–	40	–	20	–	ns
25	Multiplexed Address Hold time	t_{AHL}	20	–	20	–	20	–	10	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	90**	–	70**	–	60**	–	45**	–	ns
27	Pulse Width, AS High*	PW_{ASH}	220	–	170	–	140	–	110	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	60	–	45	–	ns
29	Usable Access Time*	t_{ACC}	595	–	465	–	380	–	270	–	ns

*At specified cycle time.

**t_{ASD} parameters listed assume external TTL clock drive with 50% ±5% duty cycle. Devices driven by an external TTL clock with 50% ±1% duty cycle or which use a crystal have the following t_{ASD} specification: 100 nanoseconds minimum (1.0 MHz devices), 80 nanoseconds minimum (1.25 MHz devices), 65 nanoseconds minimum (1.5 MHz devices), 50 nanoseconds minimum (2.0 MHz devices).

FIGURE 7 — BUS TIMING



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

INTRODUCTION

The MC68701 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port," by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port Data Direction Register and the programmer has direct access to the port pins using the port Data Register. Port pins are labled as Pij where i identifies one of four ports and j indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced MC6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800. The programming model is depicted in Figure 8 where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the M6800 instruction set are shown in Table 1.

The basic difference between the MC6801 and the MC68701 is that the MC6801 has an onboard ROM while the MC68701 has an onboard EPROM. The MC68701 is pin and code compatible with the MC6801 and can be used to emulate the MC6801, allowing easy software development using the onboard EPROM. Software developed using the MC68701 can then be masked into the MC6801 ROM.

In order to support the onboard EPROM, the MC68701 differs from the MC6801 as follows:

(1) Mode 0 in the MC6801 is a test mode only, while in the MC68701 Mode 0 is also used to program the onboard EPROM and has interrupt vectors at $BFF0-$BFFF rather than $FFF0-$FFFF.

(2) The MC68701 RAM/EPROM Control Register has two bits used to control the EPROM in Mode 0 that are not defined in the MC6801 RAM Control Register.

(3) The $\overline{\text{RESET}}$/Vpp pin in the MC68701 is dual purpose, used to supply EPROM power as well as to reset the device; while in the MC6801 the pin is called $\overline{\text{RESET}}$ and is used only to reset the device.

3

FIGURE 8 — MC68701/6801/6803 PROGRAMMING MODEL

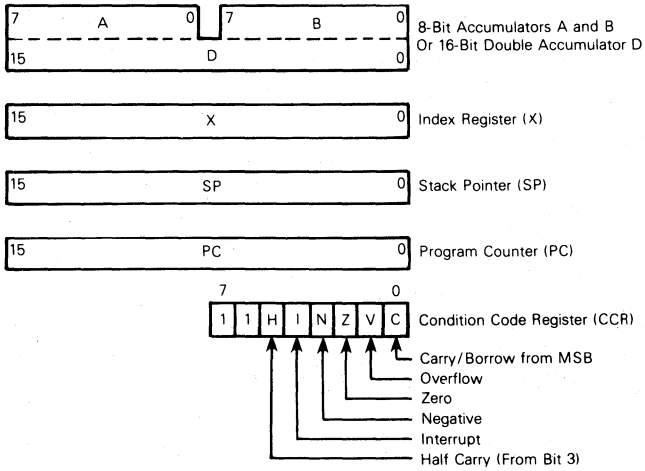


TABLE 1 — NEW INSTRUCTIONS

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

3

OPERATING MODES

The MCU provides eight different operating modes which are selectable by hardware programming and referred to as Mode 0 through Mode 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC1, SC2, and the physical location of interrupt vectors.

FUNDAMENTAL MODES

The eight MCU modes can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Modes 4 and 7 are single chip modes. Mode 5 is the expanded non-multiplexed mode, and the remaining modes are expanded multiplexed modes. Table 2 summarizes the characteristics of the operating modes.

Single-Chip Modes (4, 7)

In the Single-Chip Mode, the four MCU ports are configured as parallel input/output data ports, as shown in Figure 9. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. Peripherals or another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in Figure 10.

In Single-Chip Test Mode (4), the RAM responds to $XX80 through $XXFF and the EPROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is reset and then programmed into Mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from Mode 4 without asserting $\overline{\text{RESET}}$ by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

TABLE 2 — SUMMARY OF MC68701 OPERATING MODES

Common to all Modes:
- Reserved Register Area
- Port 1
- Port 2
- Programmable Timer
- Serial Communications Interface

Single Chip Mode 7
- 128 bytes of RAM; 2048 bytes of EPROM
- Port 3 is a parallel I/O port with two control lines
- Port 4 is a parallel I/O port
- SC1 is Input Strobe 3 (IS3)
- SC2 is Output Strobe 3 (OS3)

Expanded Non-Multiplexed Mode 5
- 128 bytes of RAM; 2048 bytes of EPROM
- 256 bytes of external memory space
- Port 3 is an 8-bit data bus
- Port 4 is an input port/address bus
- SC1 is Input/Output Select (IOS)
- SC2 is Read/Write (R/W)

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

Test Mode 4
- (1) May be changed to Mode 5 without going through Reset
- (2) May be used to test Ports 3 and 4 as I/O ports

Expanded Multiplexed Mode 0
- (1) Internal RAM and EPROM
- (2) External interrupt vectors located at $BFF0-$BFFF
- (3) Used to program EPROM

FIGURE 9 — SINGLE-CHIP MODE

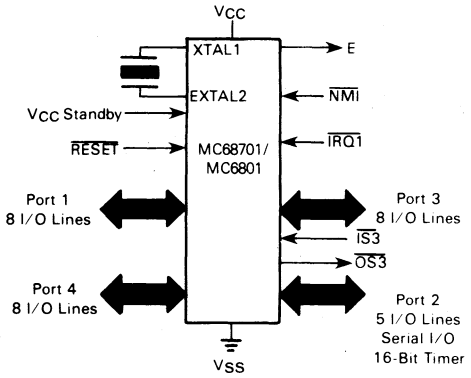


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION

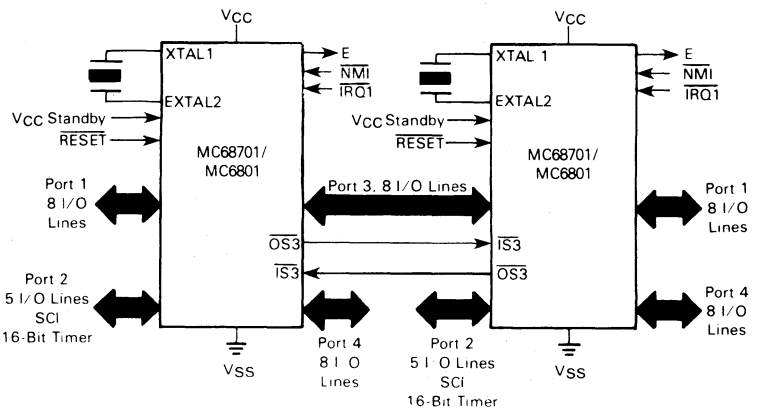
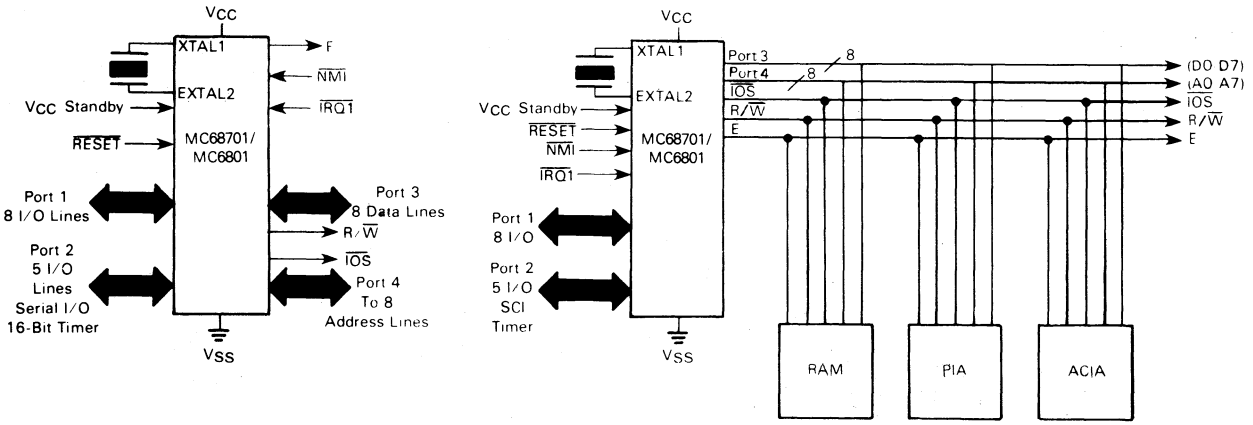


FIGURE 11 — EXPANDED NON-MULTIPLEXED CONFIGURATION



Expanded Non-Multiplexed Mode (5)

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

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

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

In the Expanded-Multiplexed Modes, the MCU has the ability to access a 64K byte memory space. Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS), and data valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8 to A15. In Mode 6, however, Port 4 is initially configured at $\overline{RESET}$ as an input data port. The Port 4 Data Direction Register can then be changed to provide any combination of address lines, A8 to A15. Stated alternatively, any subset of A8 to A15 can be provided while retaining the remaining Port 4 lines as input data lines. Internal pullup resistors pull the Port 4 lines high until software configures the port.

Figure 12 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0 to A7, as shown in Figure 13. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the internal and external data buses are connected; there must therefore be no memory map overlap in order to avoid potential bus conflicts. Mode 0 is used to program the onboard EPROM. All interrupt vectors are external in this mode and are located at $BFF0-$BFFF.

PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

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

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

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

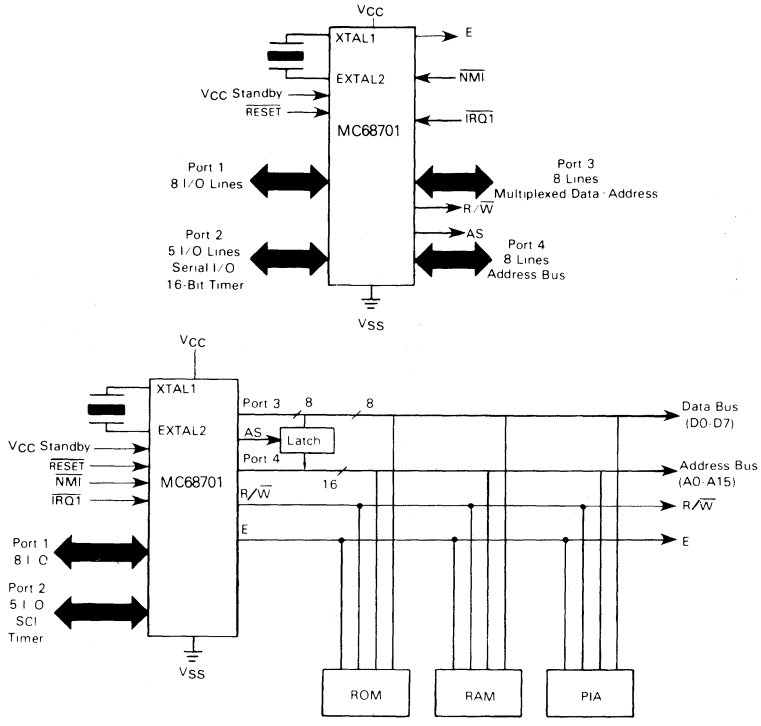
TABLE 3 — MODE SELECTION SUMMARY

Mode	P22 PC2	P21 PC1	P20 PC0	EPROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed/No RAM or EPROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and EPROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed/Programming

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

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal EPROM is disabled
(3) Interrupt vectors located at $BFF0-$BFFF
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register

FIGURE 12 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 13 — TYPICAL LATCH ARRANGEMENT

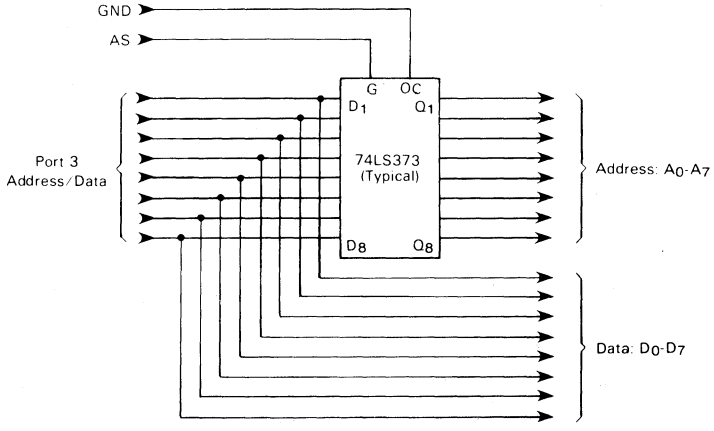
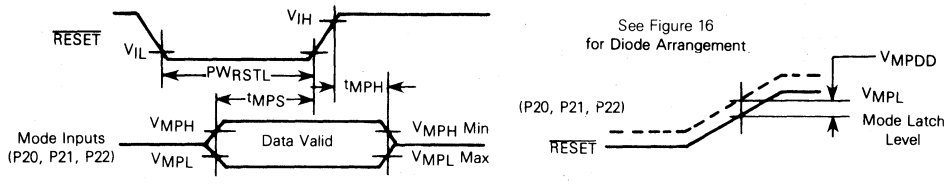


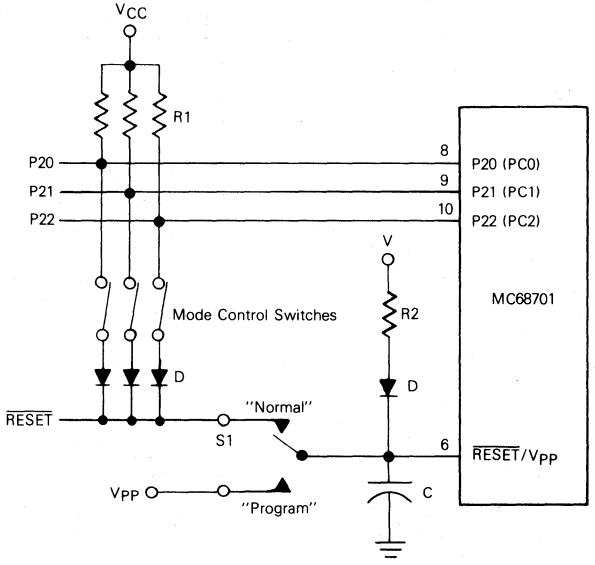
FIGURE 14 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 14)

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

FIGURE 15 — TYPICAL MODE PROGRAMMING CIRCUIT

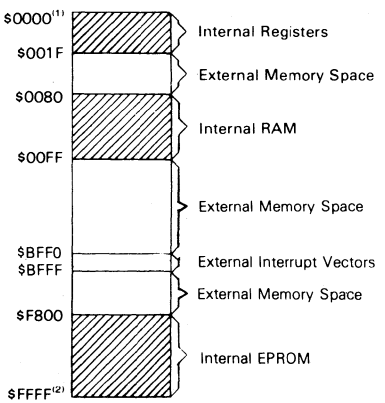


Notes:
1. Mode 0 as shown (switches closed).
2. R1 = 10k ohms (typical).
3. The $\overline{RESET}$ time constant is equal to RC where R is the equivalent parallel resistance of R2 and the number of resistors (R1) placed in the circuit by closed mode control switches.
4. D = 1N914, 1N4001 (typical).
5. If V = V_{CC}, then R2 = 50 ohms (typical) to meet V_{IH} for the $\overline{RESET}$/V_{PP} pin. V = V_{CC} is also compatible with MC6801. The $\overline{RESET}$ time constant in this case is approximately R2*C.
6. Switch S1 allows selection of normal ($\overline{RESET}$) or programming (V_{PP}) as the input to the $\overline{RESET}$/V_{PP} pin. During switching, the input level is held at a value determined by a diode (D), resistor (R2) and input voltage (V).
7. While S1 is in the "Program" position, $\overline{RESET}$ should not be asserted.
8. From powerup, $\overline{RESET}$ must be held low for at least t_{RC}. The capacitor, C, is shown for conceptual purposes only and is on the order of 1000 μF for the circuit shown. Typically, a buffer with an RC input will be used to drive $\overline{RESET}$, eliminating the need for the larger capacitor.
9. Diode V_f should not exceed V_{MPDD} min.

FIGURE 16 — MC68701 MEMORY MAPS

MC68701 Mode 0

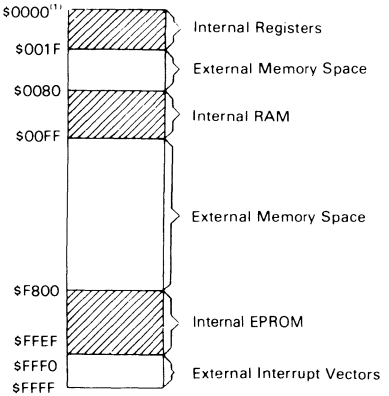
Multiplexed Test mode



Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
3) This mode is used to program the onboard EPROM.

MC68701 Mode 1

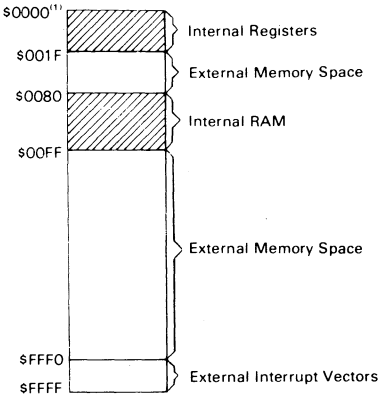
Multiplexed/RAM & EPROM



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

MC68701 Mode 2

Multiplexed/RAM

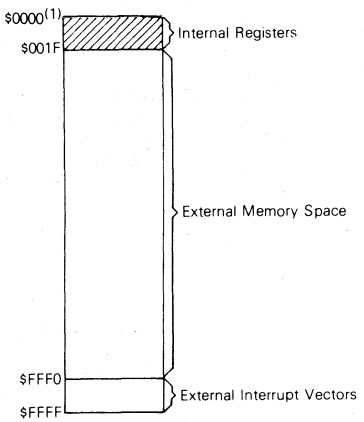


Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

FIGURE 16 — MC68701 MEMORY MAPS (CONTINUED)

MC68701 Mode 3

Multiplexed/No RAM or EPROM

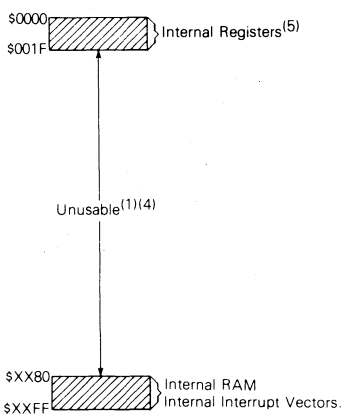


Notes:

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

MC68701 Mode 4

Single Chip Test

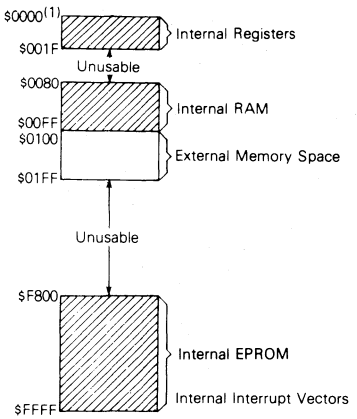


Notes:

1) The internal EPROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a "1" into the PCO bit of Port 2 Data Register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.
5) MCU read of the Port 3 Data Direction Register will access the Port 3 Data Register.

MC68701 Mode 5

Non-Multiplexed/Partial Decode



Notes:

1) Excludes the following addresses which may **NOT** be used externally: $04, $06, and $0F (No IOS).
2) This mode may be entered without going through $\overline{\text{RESET}}$ by using Mode 4 and subsequently writing a "1" into the PCO bit of Port 2 Data Register.
3) Address lines A0 to A7 will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.

FIGURE 16 — MC68701 MEMORY MAPS (CONCLUDED)

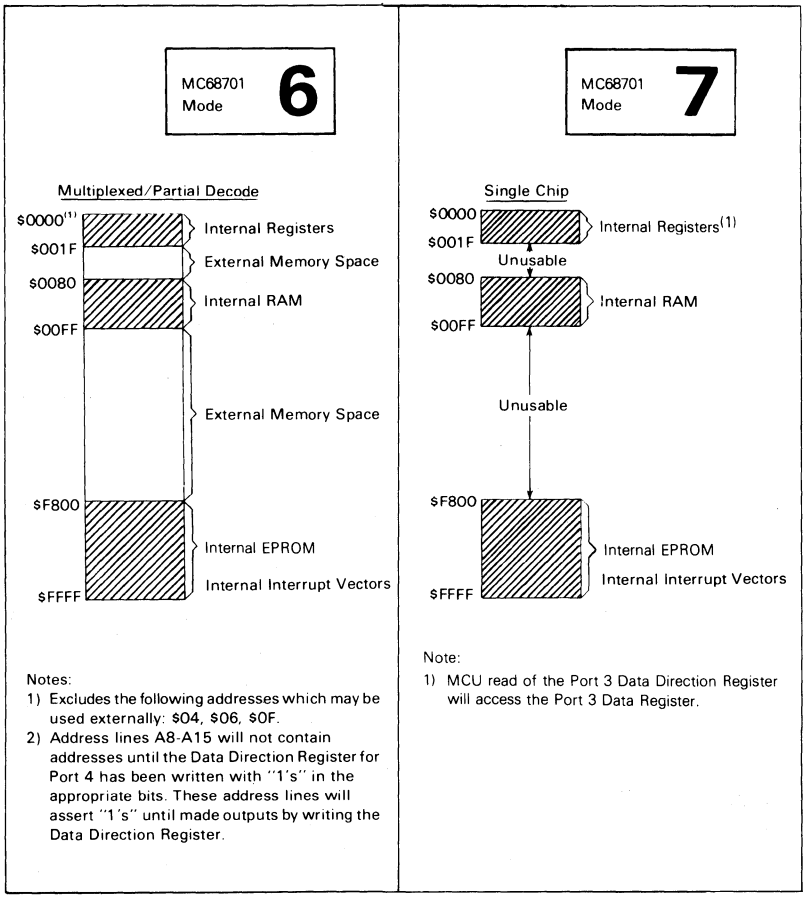


TABLE 4 — INTERNAL REGISTER AREA

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

*External addresses in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No $\overline{IOS}$)
**External addresses in Modes 0, 1, 2, 3
***1 = output, 0 = Input
***1 = Output, 0 = Input

MC68701 INTERRUPTS

The MCU supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register's I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ1}$ and $\overline{IRQ2}$. The Programmable Timer and Serial Communications Interface use an internal $\overline{IRQ2}$ interrupt line. External devices (and $\overline{IS3}$) use $\overline{IRQ1}$. An $\overline{IRQ1}$ interrupt is serviced before $\overline{IRQ2}$ if both are pending.

All $\overline{IRQ2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All MCU interrupt vector locations are shown in Table 5.

3

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

Mode 0		Modes 1-7		
MSB	LSB	MSB	LSB	Interrupt
BFFE	BFFF	FFFE	FFFF	$\overline{RESET}$
BFFC	BFFD	FFFC	FFFD	$\overline{NMI}$
BFFA	BFFB	FFFA	FFFB	Software Interrupt (SWI)
BFF8	BFF9	FFF8	FFF9	$\overline{IRQ1}$ (or $\overline{IS3}$)
BFF6	BFF7	FFF6	FFF7	ICF (Input Capture)*
BFF4	BFF5	FFF4	FFF5	OCF (Output Compare)*
BFF2	BFF3	FFF2	FFF3	TOF (Timer Overflow)*
BFF0	BFF1	FFF0	FFF1	SCI(RDRF+ORFE+TDRE)*

*IRQ2 Interrupt

The Interrupt flowchart is depicted in Figure 17 and is common to every MCU interrupt excluding reset. During interrupt servicing the Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 18 and 19.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

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

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

XTAL1 AND EXTAL2

These two input pins interface either a crystal or TTL compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz Color Burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL compatible clock at $4f_o$ with a duty cycle of 50% (±5%) with XTAL1 connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.** The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 20.

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

This input is used to reset the MCU internal state and provide an orderly startup procedure. During powerup, $\overline{RESET}$ must be held below 0.4 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} Standby reaches V_{SB} volts. $\overline{RESET}$ must be held low at least three E-cycles if asserted during powerup operation.

This pin is also used to supply V_{PP} in Mode 0 for programming the EPROM, and supplies operating power to the EPROM during powerup operation.

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the MCU input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

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

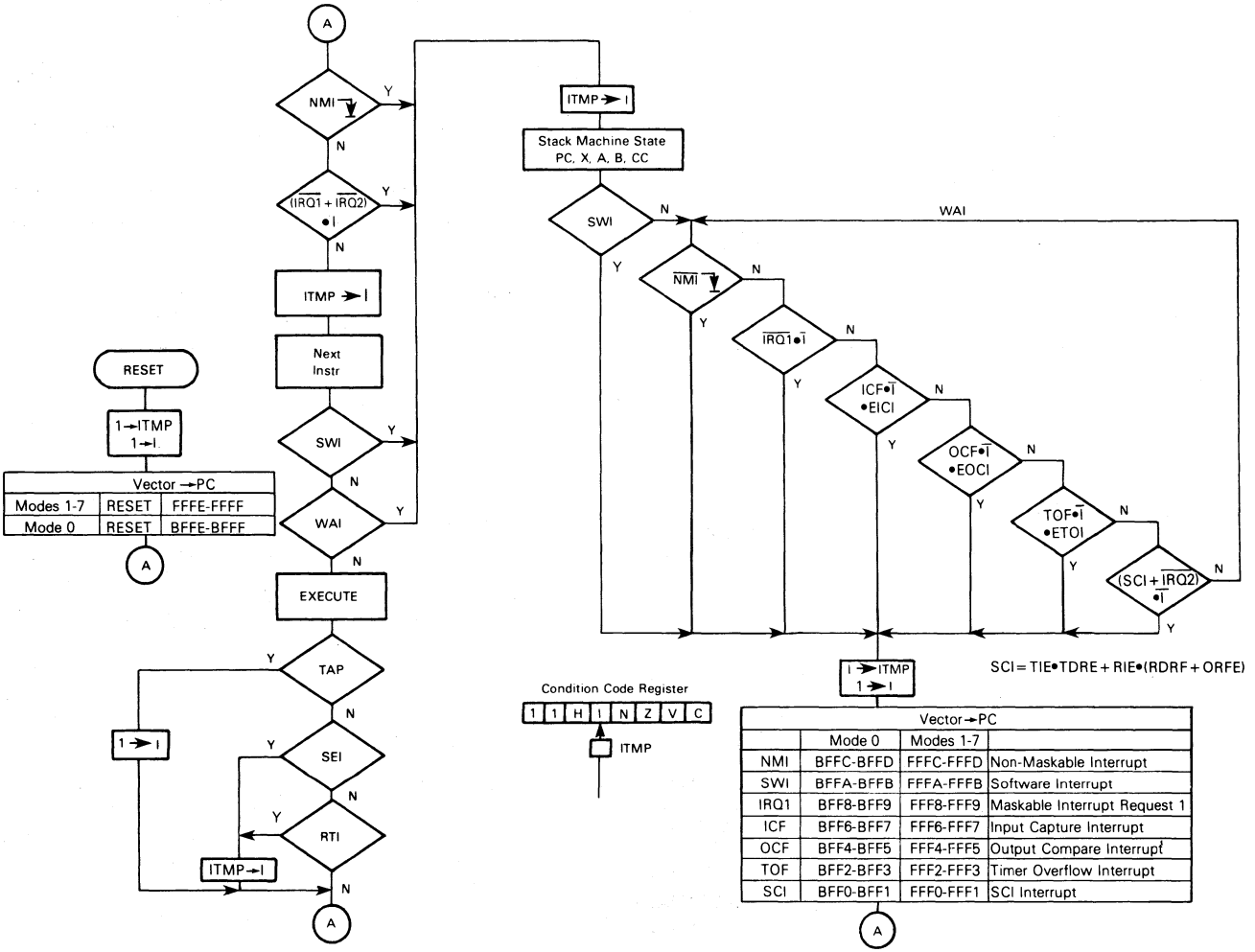
An $\overline{NMI}$ negative edge requests an MCU interrupt sequence, but the current instruction will be completed before it responds to the request. The MCU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD (or $BFFC and $BFFD in Mode 0), transferred to the Program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

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

$\overline{IRQ1}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the inter-

**Devices made with masks subsequent to T7A and CB4 incorporate an advanced clock with improved startup characteritics.

FIGURE 17 — INTERRUPT FLOWCHART



Vector →PC			
	Mode 0	Modes 1-7	
NMI	BFFC-BFFD	FFFC-FFFD	Non-Maskable Interrupt
SWI	BFFA-BFFB	FFFA-FFFB	Software Interrupt
IRQ1	BFF8-BFF9	FFF8-FFF9	Maskable Interrupt Request 1
ICF	BFF6-BFF7	FFF6-FFF7	Input Capture Interrupt
OCF	BFF4-BFF5	FFF4-FFF5	Output Compare Interrupt
TOF	BFF2-BFF3	FFF2-FFF3	Timer Overflow Interrupt
SCI	BFF0-BFF1	FFF0-FFF1	SCI Interrupt

FIGURE 18 — INTERRUPT SEQUENCE

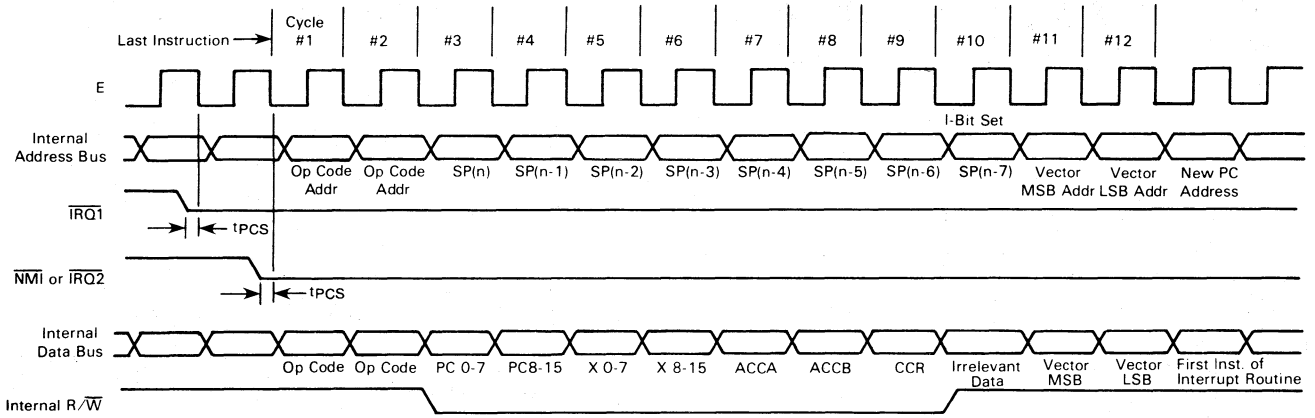
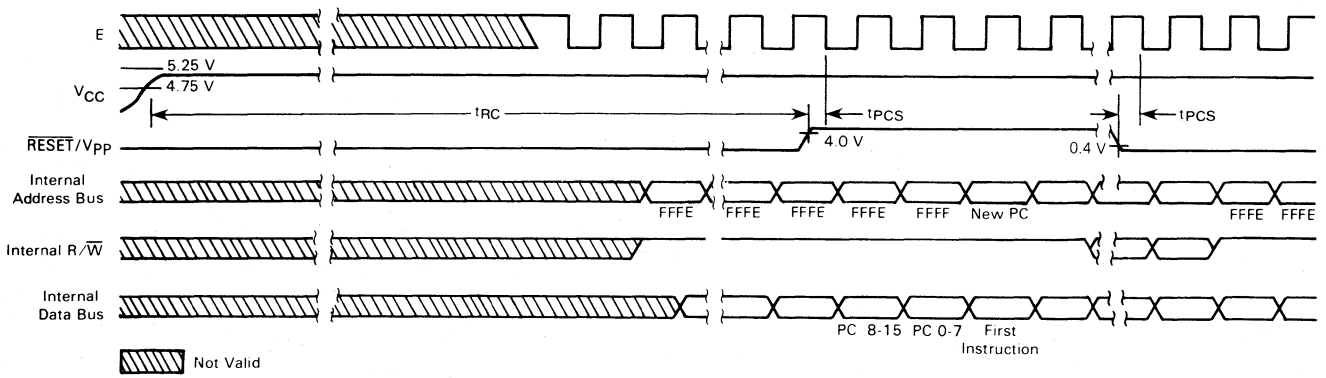


FIGURE 19 — RESET TIMING



rupt mask bit (I-bit) in the Condition Code Register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9 (or $BFF8 and $BFF9 in Mode 0), transferred to the Program Counter, and instruction execution is resumed.

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

SC1 AND SC2 (STROBE CONTROL 1 AND 2)

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

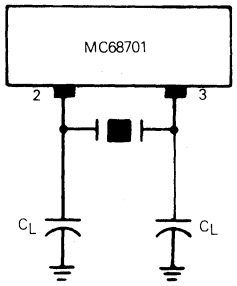
SC1 and SC2 In Single Chip Mode

In Single Chip Mode, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as $\overline{IS3}$ and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{IS3}$ are controlled by the Port 3 Control and Status Register and are discussed in the Port 3 description. If unused, $\overline{IS3}$ can remain unconnected.

SC2 is configured as $\overline{OS3}$ and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{OS3}$ timing is shown in Figure 5.

FIGURE 20 — MC68701 OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters



C_L = 20 pF (typical)

NOTE

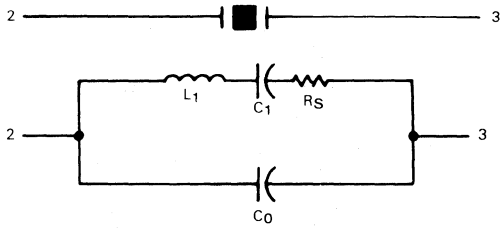
TTL-compatible oscillators may be obtained from:

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

MC68701 Nominal Crystal Parameters

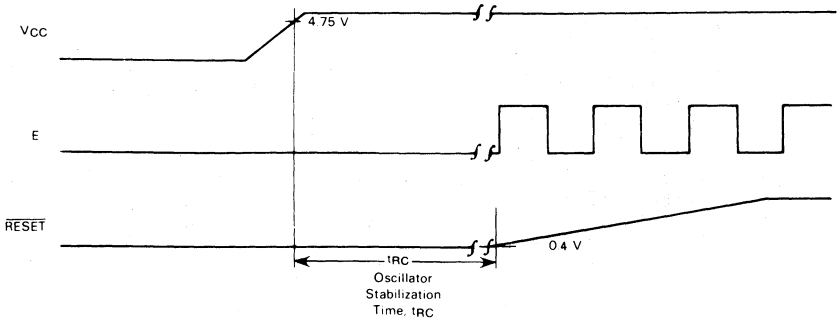
	3.58 MHz	4.00 MHz	5.0 MHz	6.0 MHz	8.0 MHz
RS	60 Ω	50 Ω	30-50 Ω	30-50 Ω	20-40 Ω
C_0	3.5 pF	6.5 pF	4.6 pF	4-6 pF	4.6 pF
C_1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40 k	>30 k	>20 k	>20 k	>20 k

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



Equivalent Circuit

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



SC1 And SC2 In Expanded Non-Multiplexed Mode

In the Expanded Non-Multiplexed Mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select ($\overline{IOS}$) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

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

SC1 And SC2 In Expanded Multiplexed Mode

In the Expanded Multiplexed Modes, both SC1 and SC2 are configured as outputs. SC1 functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in Figure 15.

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

P10-P17 (PORT 1)

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

P20-P24 (PORT 2)

Port 2 is a mode-independent, 5-bit, multipurpose I/O port. The voltage levels present on P20, P21, and P22 on the rising edge of $\overline{RESET}$ determine the operating mode of the MCU. The entire port is then configured as a data input port. The Port 2 lines can be selectively configured as data output lines by setting the appropriate bits in the Port 2 Data Direction Register. The Port 2 Data Register is used to move data through the port. However, if P21 is configured as an output, it will be tied to the timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

Port 2 can also be used to provide an interface for the Serial Communications Interface and the timer Input Edge function. These configurations are described in the appropriate SCI and Timer sections of this publication.

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

PORT 2 DATA REGISTER

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

P30-P37 (PORT 3)

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

Port 3 In Single-Chip Mode

Port 3 is an 8-bit I/O port in the Single-Chip Mode, with each line configured by the Port 3 Data Direction Register. There are also two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and are available only in Single-Chip Mode: (1) Port 3 input data can be latched using IS3 as a control signal, (2) $\overline{OS3}$ can be generated by either an MPU read or write to the Port 3 Data Register, and (3) an $\overline{IRQ1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 4.

PORT 3 CONTROL AND STATUS REGISTER

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

Bit 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of Port 3 Data Register. LATCH ENABLE is cleared during reset.

Bit 4 OSS (Output Strobe Select). This bit determines whether $\overline{OS3}$ will be generated by a read or write of the Port 3 Data Register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared during reset.

Bit 5 Not used.

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

Bit 7 IS3 FLAG. This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the Port 3 Control and Status Register (with IS3 FLAG set) followed by a read or write to the Port 3 Data Register or during reset.

Port 3 In Expanded Non-Multiplexed Mode

Port 3 is configured as a bidirectional data bus (D7-D0) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write (SC2). Data is clocked by E (Enable).

Port 3 In Expanded Multiplexed Mode

Port 3 is configured as a time multiplexed address (A0-A7) and data bus (D7-D0) in the Expanded Multiplexed Modes where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent potentional bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

Port 4 In Single Chip Mode

In Single Chip Mode, Port 4 functions as an 8-bit I/O port with each line configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Port 4 In Expanded Non-Multiplexed Mode

Port 4 is configured during reset as an 8-bit input port, where the Port 4 Data Direction Register can be written to provide any or all of eight address lines A0 to A7. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured.

Port 4 In Expanded Multiplexed Mode

In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured during reset as an 8-bit parallel input port, where the Port 4 Data Direction Register can be written to provide any or all of upper address lines A8 to A15. Internal pullup resistors pull the lines high until the Port 4 Data Direction Register is configured, where bit 0 controls A8.

RESIDENT MEMORY

The MC68701 has 128 bytes of onboard RAM and 2048 bytes of onboard UV erasable EPROM. This memory is controlled by four bits in the RAM/EPROM Control Register.

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

Power must be supplied to V_{CC} standby if the internal RAM is to be used, regardless of whether standby power operation is anticipated. In Mode 3, V_{CC} standby should be tied to ground.

The RAM is controlled by the RAM/EPROM Control Register.

RAM/EPROM CONTROL REGISTER ($14)

The RAM/EPROM Control Register includes four bits: STBY PWR, RAME, PPC, and PLC. Two of these bits, STBY PWR and RAME, are used to control RAM access and determine the adequacy of the standby power source during power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down procedure. RAME and STBY PWR are Read/Write bits.

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

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

A description of the RAM/EPROM Control Register follows.

MC68701 RAM/EPROM CONTROL REGISTER

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

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

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

PLC = 1 EPROM address latch is transparent.

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

PPC = 0 EPROM programming power (V_{PP}) applied.

PPC = 1 EPROM programming power (V_{PP}) is not applied.

Bit 2-5 — Unused.

Bit 6 RAME — RAM Enable. This Read/Write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of reset. If RAME is clear, any access to a RAM address is external. If RAME is set and not in Mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR — Standby Power. This bit is a read/write status bit which, when once set, remains set as long as V_{CC} standby remains above V_{SBB} (minimum). As long as this bit is set following a period of standby operation, the standby power supply has adequately preserved the data in the standby RAM. If this bit is cleared during a period of standby operation, it indicates that V_{CC} standby had fallen to a level sufficiently below V_{SBB} (minimum) to suspect that data in the standby RAM is not valid. This bit can be set only by software and is not affected during reset.

Note that if PPC and PLC are set, they cannot be simultaneously cleared with a single MPU write. The PLC bit must be cleared prior to attempting to clear PPC. If both PPC and PLC are clear, setting PLC will also set PPC. In addition,

it is assumed that V_{PP} is applied to the RESET/V_{PP} pin whenever PPC is clear. If this is not the case, the result is undefined.

ERASING THE MC68701 EPROM

Ultraviolet erasure will clear all bits of the EPROM to the "0" state. Note that this erased state differs from that of some other widely used EPROMs (such as the MCM68708) where the erased state is a "1". The MC68701 EPROM is programmed by erasing it to "0's" and entering "1's" into the desired bit locations.

The MC68701 EPROM can be erased by exposure to high intensity ultraviolet light with a wave length of 2537A for a minimum of 30 minutes. The recommended integrated dose (UV intensity X exposure time) is 15 Ws/cm. The lamps should be used without shortwave filters and the MC68701 should be positioned about one inch away from the UV tubes.

The MC68701 transparent lid should always be covered after erasing. This protects both the EPROM and light-sensitive nodes from accidental exposure to ultraviolet light.

PROGRAMMING THE MC68701 EPROM

When the MC68701 is released from Reset in Mode 0, a vector is fetched from location $BFFE:BFFF. This provides a method for an external program to obtain control of the microcomputer with access to every location in the EPROM.

To program the EPROM, it is necessary to operate the MC68701 in Mode 0 under the control of a program resident in external memory which can facilitate loading and programming of the EPROM. After the pattern has been loaded into external memory, the EPROM can be programmed as follows:

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

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

A routine which can be used to program the MC68701 EPROM is provided at the end of this publication. This non-reentrant routine requires four double byte variables named IMBEQ, IMEND, PNTR, and WAIT to be initialized prior to entry to the routine. These variables indicate (a) the first and last memory locations which bound the data to be programmed into the EPROM, (b) the first EPROM location to be programmed, and (c) a number which is used to generate the programming time delay. The last variable, WAIT, takes into account the MCU input crystal (or TTL-compatible clock) frequency to insure the programming time, t_{pp}, is met. WAIT is defined as the number of MPU E-cycles that will occur in the real-time EPROM programming interval, t_{pp}. For example, if t_{pp} = 50 milliseconds and the MC68701 is being driven with a 4.00 MHz TTL-compatible clock:

$$\text{WAIT (MPU E-cycles)} = .t_{pp} * (\text{MCU INPUT FREQ/})4 * 10^6$$
$$= 50000(4 * 10^6)/4 * 10^6$$
$$= 50000$$

NOTE

A monitor program called PRObug® is available from Motorola Microsystems. PRObug contains a user option for programming the on-board MC68701 EPROM.

PROGRAMMABLE TIMER

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 21.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during reset and is read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all 1's.

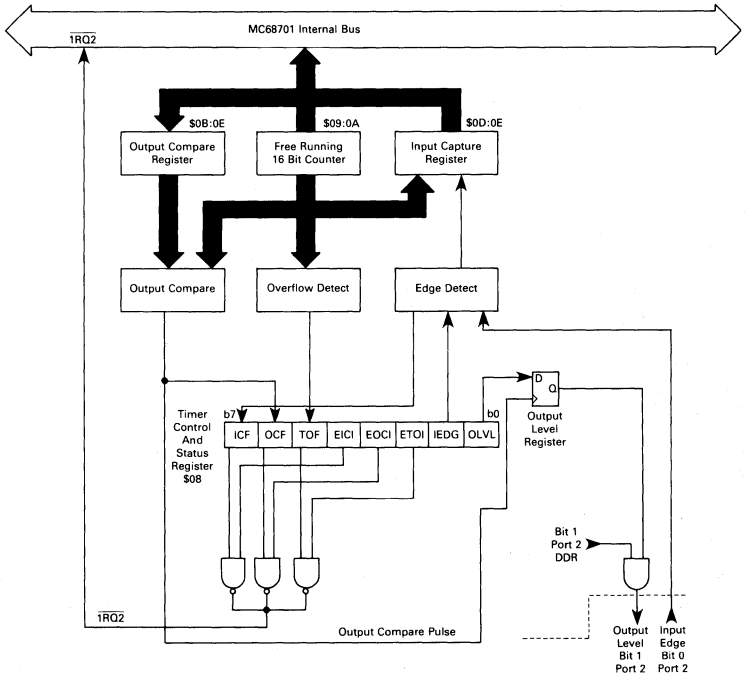
OUTPUT COMPARE REGISTER ($0B:0C)

The Output Compare Register is a 16-bit Read/Write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E-cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P21 and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to the high byte of the Compare Register ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF during reset.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always

FIGURE 21 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



senses P20 even when configured as an output. An input capture can occur independently of ICF: the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTER ($08)

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

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

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

TIMER CONTROL AND STATUS REGISTER (TCSR)

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

Bit 0 OLVL — Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared during reset.

Bit 1 EIDG — Input Edge. IEDG is cleared during reset and controls which level transition will trigger a counter transfer to the Input Capture Register:

IEDG = 0 Transfer on a negative-edge

IEDG = 1 Transfer on a positive-edge.

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

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

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

Bit 5 TOF — Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) then reading the counter high byte ($09), or by RESET.

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

Bit 7 ICF — Input Capture Flag. ICF is set to indicate a proper level transition; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte ($0D), or by RESET.

SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and Bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until the data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

- format: standard mark/space (NRZ) or Bi-phase
- clock: external or internal bit rate clock
- Baud : one of 4 per E-clock frequency, or external clock (X8 desired baud)
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in Figure 22. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

Rate and Mode Control Register (RMCR) ($10)

The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared during reset. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

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

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

Bit 3:Bit 2 — CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (±10%). If CC1:CC0=10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE: The source of SCI internal bit rate clock is the timer free running counter. An MPU write to the counter can disturb serial operations.

Transmit/Receive Control And Status Register (TRCSR) ($11)

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

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

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

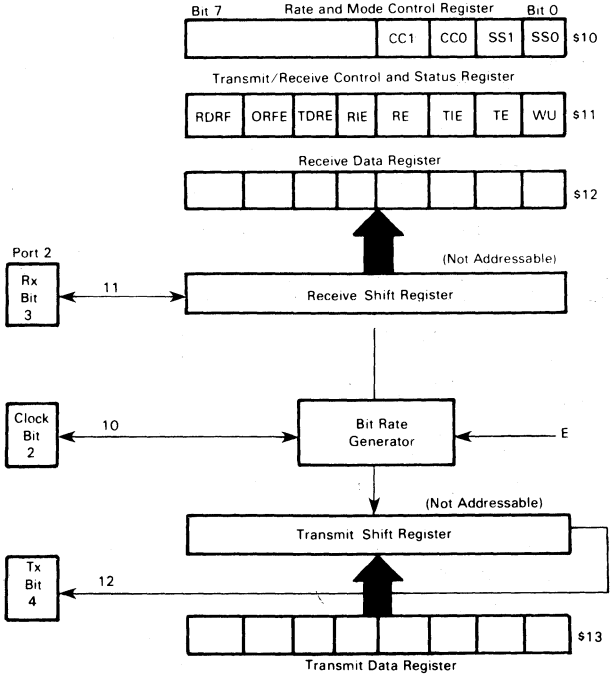
TABLE 6 — SCI BIT TIMES AND RATES

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

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2, Bit 2
0 0	Bi-Phase	Internal	Not Used
0 1	NRZ	Internal	Not Used
1 0	NRZ	Internal	Output
1 1	NRZ	External	Input

FIGURE 22 — SCI REGISTERS



Bit 0 WU — "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or during reset. WU will not set if the line is idle.

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

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

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

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

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

Bit 6 ORFE — Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. Unframed data causing a framed error is transferred to the Receive Data Register. However, subsequent data transfer is blocked until the framing error flag is cleared.* ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or during reset.

Bit 7 RDRF — Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or during reset.

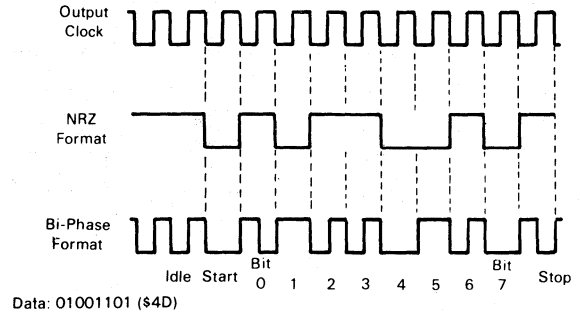
SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting to 9-bit preamble of 1's.

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

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

FIGURE 23 — SCI DATA FORMATS



Data: 01001101 ($4D)

*Devices made with mask numbers T7A and CB4 do not transfer unframed data to the Receive Data Register.

INSTRUCTION SET

The MC68701 is upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply. A list of new operations added to the MC6800 instruction set is shown in Table 1. In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the program counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

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

PROGRAMMING MODEL

A programming model for the MC68701 is shown in Figure 9. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

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

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

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

Accumulators — The MCU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

Condition Code Registers — The condition code register indicates the results of an instruction and includes the Overflow (V), Carry/Borrow from MSB (C), and Half Carry following five condition bits: Negative (N), Zero (Z), from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7 are read as ones.

ADDRESSING MODES

The MC68701 provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 9, 10, 11, and 12 where execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, E cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and a description of selected instructions is shown in Figure 24.

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

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

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

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

Inherent Addressing — The operand(s) are registers and no memory reference is required. These are single byte instructions.

Relative Addressing — Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of −126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	•				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	•				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	•				75	•				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	•				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	•				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	•				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	•				47	ASRA		2	1	7B	•				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	•				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	•				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	•				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	•				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	•				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	•				51	•				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	•				52	•				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	•				8F	•				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	•				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	•				99	ADCA		3	2	CD	•								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

* UNDEFINED OP CODE

NOTES:

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

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

3

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

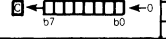
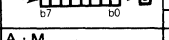
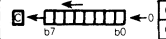
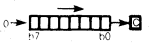
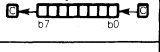
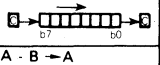
Accumulator and Memory Operations	MNE	Immed Op	Immed ~	Immed #	Direct Op	Direct ~	Direct #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inher Op	Inher ~	Inher #	Boolean Expression	Condition Codes H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	1		•	•	↕	↕	↕	↕

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

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct Op	Direct ~	Direct #	Relative Op	Relative ~	Relative #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inherent Op	Inherent ~	Inherent #	Branch Test	Condition Code Reg. 5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3				See Special Operations-Figure 24	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 24	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write (R/W) line during each cycle of each instruction.

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

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

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

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

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/W Line	Data Bus
EXTENDED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
STD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Opcode Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

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

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

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

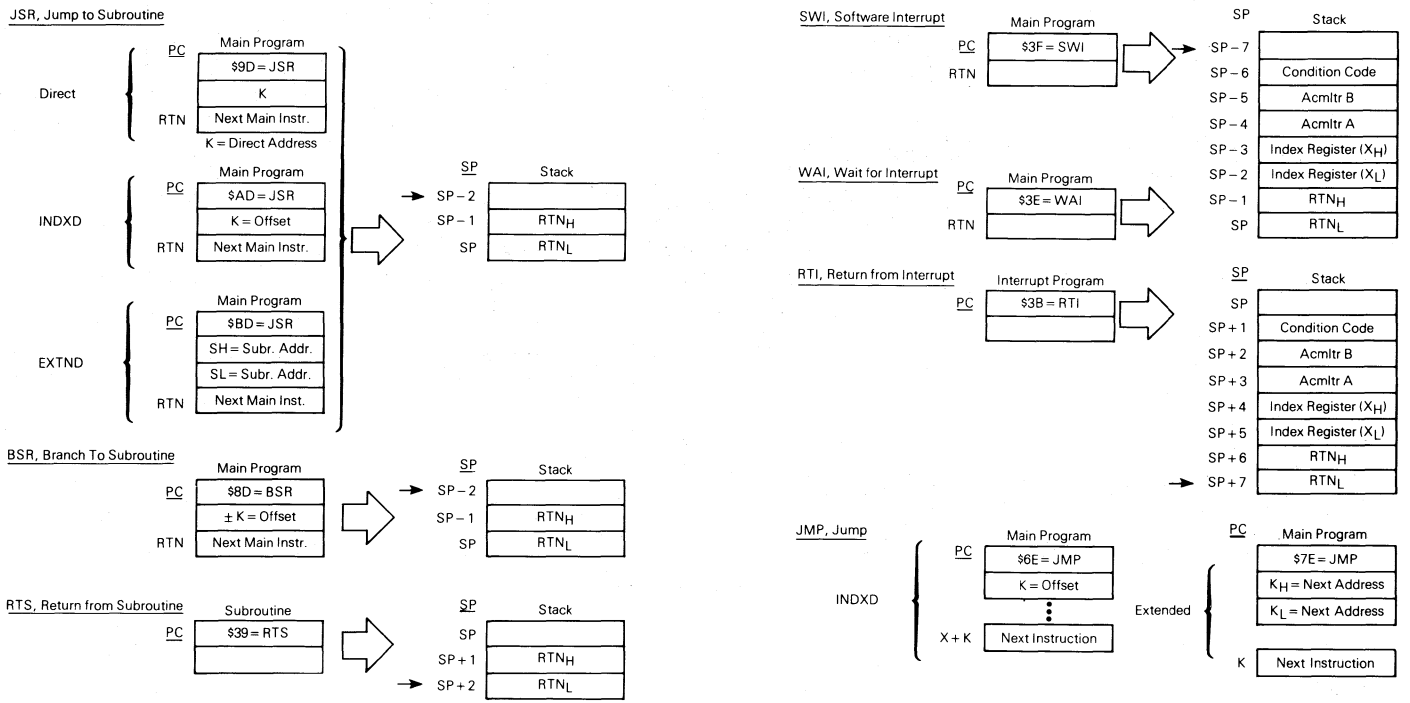
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Stack Pointer − 5 Stack Pointer − 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

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

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer -1	0	Return Address (High Order Byte)

FIGURE 24 — SPECIAL OPERATIONS



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

EPROM PROGRAMMING ROUTINE

```
PAGE  001  EPROM    .SA:1  EPROM  *** ROUTINE TO PROGRAM THE MC68701 EPROM ***

00001                        NAM     EPROM
00002                        OPT     Z01,LLEN=80
00003                        TTL     *** ROUTINE TO PROGRAM THE MC68701 EPROM **
00004
00005                  ************************************************************
00006                  *
00007                  *  E P R O M -- A  NON-REENTRANT  ROUTINE  TO  PROGRAM
00008                  *                THE MC68701 EPROM.
00009                  *
00010                  *                THE ROUTINE PROGRAMS THE MC68701 EPROM
00011                  *                STARTING  AT  ADDRESS  "PNTR"  FROM  A
00012                  *                BLOCK OF MEMORY  STARTING  AT  "IMBEG"
00013                  *                AND ENDING AT "IMEND".
00014                  *
00015                  *  CALLING CONVENTION:
00016                  *
00017                  *     JSR  EPROM
00018                  *
00019                  *  NOTES:
00020                  *
00021                  *     1.  THE ROUTINE EXPECTS FOUR DOUBLE BYTE  VALUES
00022                  *         TO  BE INITIALIZED PRIOR  TO  BEING  CALLED.
00023                  *         THESE VALUES ARE:
00024                  *
00025                  *         IMBEG = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00026                  *                 TO THE FIRST BYTE TO  BE  PROGRAMMED
00027                  *                 INTO THE EPROM.
00028                  *
00029                  *         IMEND = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00030                  *                 TO THE LAST BYTE TO BE PROGRAMED IN-
00031                  *                 INTO THE EPROM.
00032                  *
00033                  *         PNTR  = A DOUBLE BYTE ADDRESS  WHICH  POINTS
00034                  *                 TO THE FIRST BYTE IN THE EPROM TO BE
00035                  *                 PROGRAMMED.
00036                  *
00037                  *         WAIT  = A DOUBLE BYTE COUNTER VALUE WHICH IS
00038                  *                 A FUNCTION OF THE MCU INPUT FREQUEN-
00039                  *                 CY AND IS USED WITH THE OUTPUT  COM-
00040                  *                 PARE FUNCTION TO  GENERATE A 50 MSEC
00041                  *                 TIMEOUT.  IT IS EQUIVALENT TO
00042                  *
00043                  *                 50000 * (MCU INPUT FREQ) / 4 * 10**6
00044                  *
00045                  *                 VALUES FOR TYPICAL INPUT FREQS  ARE:
00046                  *
00047                  *                   WAIT                MCU INPUT FREQ
00048                  *                 -------------         --------------
00049                  *                 30615 ($7797)            2.45 MHZ
00050                  *                 50000 ($C350)            4.00 MHZ
00051                  *                 61375 ($EFBF)            4.91 MHZ
00052                  *
00053                  *     2.  IT IS ASSUMED THAT POWER (VPP) IS  AVAILABLE
00054                  *         TO THE RESET PIN FOR PROGRAMMING.
00055                  *
00056                  *     3.  THIS ROUTINE  PERFORMS  NO  ERROR  CHECKING.
00057                  *
00058                  *  Routine parameter initialization, such as stack pointer, etc., must be done prior to entry.
                        (Use of PROBug will ensure all needed initialization.)
```

EPROM PROGRAMMING ROUTINE

```
PAGE  002  EPROM   .SA:1  EPROM  *** ROUTINE TO PROGRAM THE MC68701 EPROM ***

00060
00061                    *  E Q U A T E S
00062
00063           0008  A TCSR   EQU    $08       TIMER CONTROL/STAT REGISTER
00064           0009  A TIMER  EQU    $09       COUNTER REGISTER
00065           000B  A OUTCMP EQU    $0B       OUTPUT COMPARE REGISTER
00066           0014  A EPMCNT EQU    $14       RAM/EPROM CONTROL REGISTER
00067
00068                    *  L O C A L     V A R I A B L E S
00069
00070A 0080                      ORG    $80
00071A 0080     0002  A IMBEG  RMB    2         START OF MEMORY BLOCK
00072A 0082     0002  A IMEND  RMB    2         LAST BYTE OF MEMORY BLOCK
00073A 0084     0002  A PNTR   RMB    2         FIRST BYTE OF EPROM TO BE PGM'D
00074A 0086     0002  A WAIT   RMB    2         COUNTER VALUE
00075
00076                    *  E P R O M     S T A R T S     H E R E
00077
00078A 3000                      ORG    $3000
00079A 3000 DE 84     A EPROM  LDX    PNTR      SAVE CALLING ARGUMENT
00080A 3002 3C                   PSHX           RESTORE WHEN DONE
00081A 3003 DE 80     A         LDX    IMBEG     USE STACK
00082
00083A 3005 3C          EPR002 PSHX             SAVE POINTER ON STACK
00084A 3006 86 FE     A         LDAA   #$FE      REMOVE VPP, SET LATCH
00085A 3008 97 14     A         STAA   EPMCNT    PPC=1, PLC=0
00086A 300A A6 00     A         LDAA   X         MOVE DATA MEMORY-TO-LATCH
00087A 300C DE 84     A         LDX    PNTR      GET WHERE TO PUT IT
00088A 300E A7 00     A         STAA   X         STASH AND LATCH
00089A 3010 08                   INX            NEXT ADDR
00090A 3011 DF 84     A         STX    PNTR      ALL SET FOR NEXT
00091A 3013 86 FC     A         LDAA   #$FC      ENABLE EPROM POWER (VPP)
00092A 3015 97 14     A         STAA   EPMCNT    PPC=0, PLC=0
00093
00094                    * NOW WAIT FOR 50 MSEC TIMEOUT USING OUTPUT COMPARE.
00095
00096A 3017 DC 86     A         LDD    WAIT      GET CYCLE COUNTER
00097A 3019 D3 09     A         ADDD   TIMER     BUMP CURRENT VALUE
00098A 301B 7F 0008   A         CLR    TCSR      CLEAR OCF
00099A 301E DD 0B     A         STD    OUTCMP    SET OUTPUT COMPARE
00100A 3020 86 40     A         LDAA   #$40      NOW WAIT FOR OCF
00101
00102A 3022 95 08     A EPR004 BITA   TCSR
00103A 3024 27 FC 3022          BEQ    EPR004    NOT YET
00104A 3026 38                  PULX             SETUP FOR NEXT ONE
00105A 3027 08                  INX              NEXT
00106A 3028 9C 82     A         CPX    IMEND     MAYBE DONE
00107A 302A 23 D9 3005          BLS    EPR002    NOT YET
00108A 302C 86 FF     A         LDAA   #$FF      REMOVE VPP, INHIBIT LATCH
00109A 302E 97 14     A         STAA   EPMCNT    EPROM CAN NOW BE READ
00110A 3030 38                  PULX             RESTORE PNTR
00111A 3031 DF 84     A         STX    PNTR
00112A 3033 39                  RTS              THAT'S ALL
00113                           END
TOTAL ERRORS 00000--00000
```

IMPORTANT NOTICE

Devices made with mask numbers T7A and CB4 may generate multiple framing error flags in response to unframed data. These devices will eventually synchronize correctly after a framing error; but valid, framed data following an unframed byte may generate false framing error flags.

MOTOROLA

MC68701U4

Advance Information

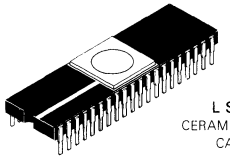
HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

8-BIT EPROM MICROCOMPUTER/ MICROPROCESSOR

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

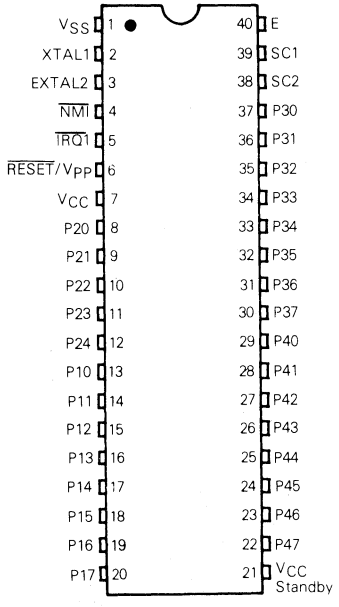
The MC68701U4 is an 8-bit single-chip EPROM microcomputer unit (MCU) which enhances the capabilities of the MC6801 and significantly enhances the capabilities of the M6800 Family of parts. It includes an MC6801 microprocessor unit (MPU) with direct object-code compatibility and upward object-code compatibility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

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



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT

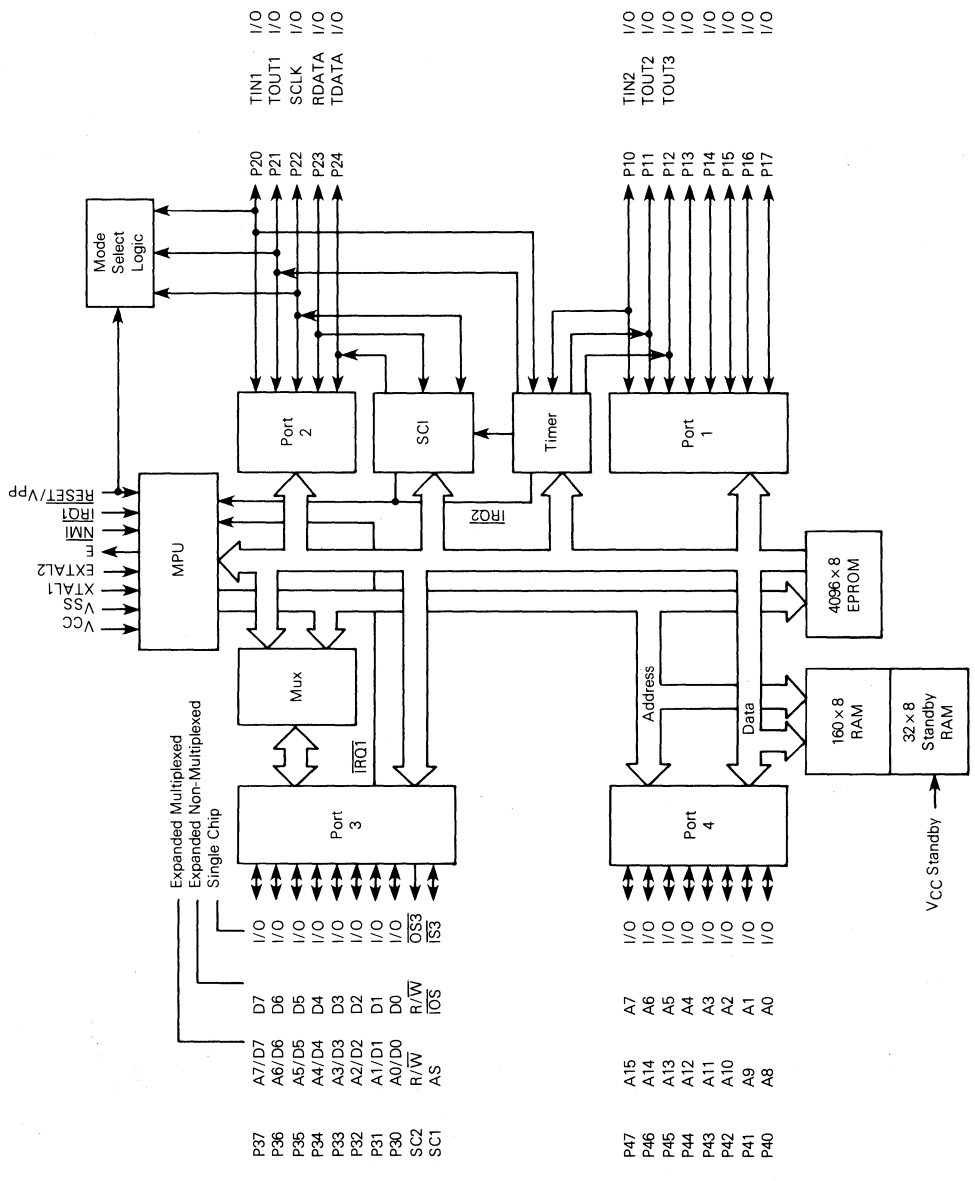


GENERIC INFORMATION
(T_A = 0° to 70°C)

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

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

BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range Programmed Unprogrammed	T_{stg}	 −55 to +100 −55 to +150	°C

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

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

THERMAL CHARACTERISTICS

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

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

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

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

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{RESET}$	V_{IH}	V_{SS} + 4.0	–	V_{CC}	V
	Other Inputs*		V_{SS} + 2.0	–	V_{CC}	
Input Low Voltage	$\overline{RESET}$	V_{IL}	V_{SS} – 0.3	–	V_{SS} + 0.4	V
	Other Inputs*		V_{SS} – 0.3	–	V_{SS} + 0.8	
Input Current (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	–	0.5	mA
See Note	SC1		–	–	0.8	
Input Current (V_{in} = 0 to 5.25 V)	$\overline{NMI}$, $\overline{IRQ1}$	I_{in}	–	–	2.5	μA
Input Current (V_{in} = 0 to 0.4 V)	$\overline{RESET}/V_{PP}$	I_{in}	–	–2.0	–	mA
See Note (V_{in} = 4.0 V to V_{CC})			–	–	8.0	
Hi-Z (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	P10-P17, P20-P24, P30-P37	I_{TSI}	–	–	10	μA
Output High Voltage (I_{load} = – 65 μA, V_{CC} = min)	P40-P47, SC1, SC2	V_{OH}	V_{SS} + 2.4	–	–	V
(I_{load} = – 100 μA, V_{CC} = min)	Other Outputs		V_{SS} + 2.4	–	–	
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)	All Outputs	V_{OL}	–	–	V_{SS} + 0.5	V
Darlington Drive Current (V_O = 1.5 V)	P10-P17	I_{OH}	1.0	–	5.0	mA
Internal Power Dissipation (measured at T_A = 0°C in Steady-State Operation)		P_{INT}	–	–	1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	P30-P37, P40-P47, SC1	C_{in}	–	–	12.5	pF
	Other Inputs		–	–	10.0	
V_{CC} Standby	Power Down	V_{SBB}	4.0	–	5.25	V
	Power Up	V_{SB}	4.75	–	5.25	
Standby Current	Power Down	I_{SBB}	–	–	3.0	mA
Programming Time (Per Byte) (T_A = 25°C)		t_{PP}	25	–	50	ms
Programming Voltage (T_A = 25°C)		V_{PP}	20.0	21.0	22.0	V
Programming Current ($V_{\overline{RESET}} = V_{PP}$) ($T_A$ = 25°C)		I_{PP}	–	30.0	50.0	mA

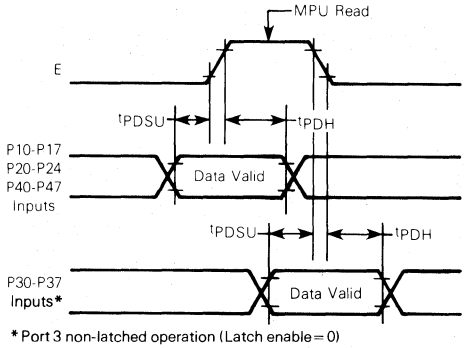
*Except Mode Programming Levels; See Figure 16.

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

PERIPHERAL PORT TIMING (Refer to Figures 1-4)

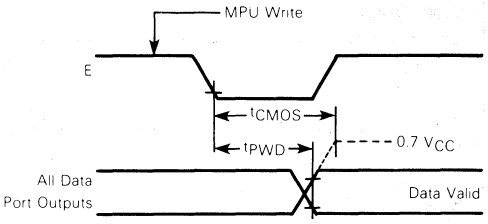
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to OS3 Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to OS3 Positive Transition	t_{OSD2}	–	–	350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

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



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

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



NOTES:
1. 10 k pullup resistor required for port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

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

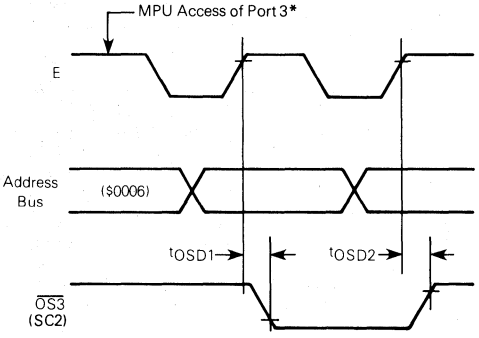
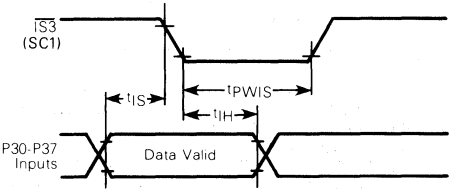


FIGURE 4 — PORT 3 LATCH TIMING
(SINGLE-CHIP MODE)



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

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

FIGURE 5 — CMOS LOAD

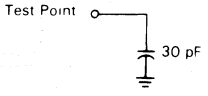
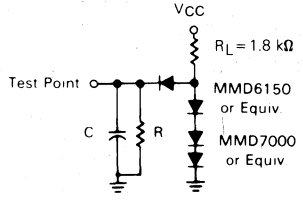


FIGURE 6 — TIMING TEST
LOAD PORTS 1, 2, 3, AND 4



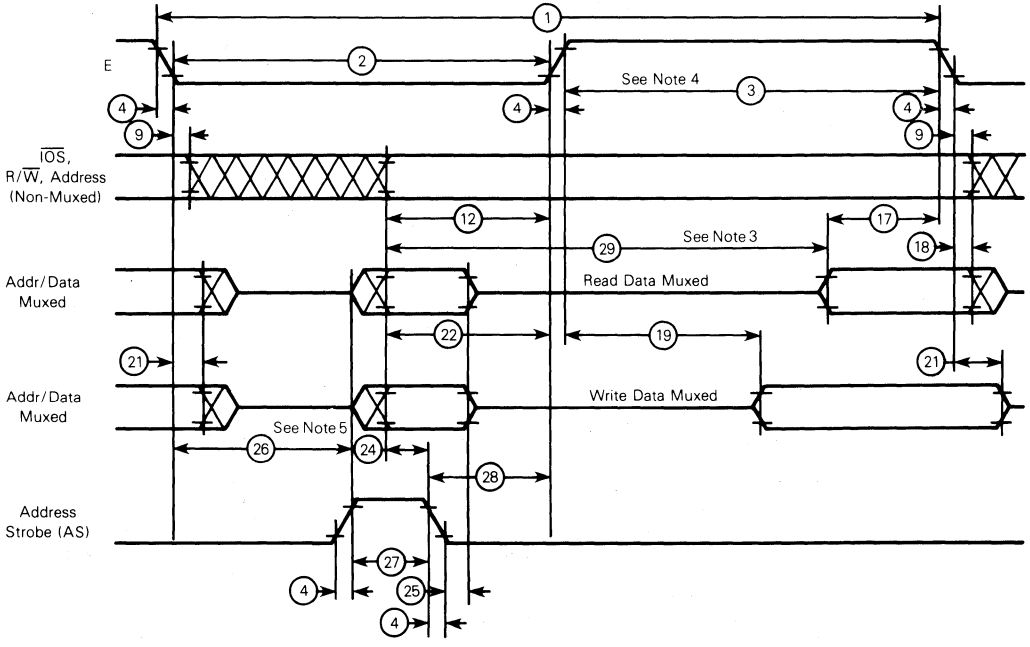
C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P10-P17, P20-P24, P30-P37, E

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

Ident. Number	Characteristics	Symbol	MC68701U4		MC68701U4-1		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	2.0	0.8	2.0	μs
2	Pulse Width, E Low	PW_{EL}	430	1000	360	1000	ns
3	Pulse Width, E High	PW_{EH}	450	1000	360	1000	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Address Hold Time	t_{AH}	20	–	20	–	ns
12	Non-Muxed Address Valid Time to E*	t_{AV}	200	–	150	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	70	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	200	ns
21	Write Data Hold Time	t_{DHW}	20	–	20	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	160	–	120	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	40	–	30	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	20	–	ns
26	Delay Time, E to AS Rise*	t_{ASD}	200	–	170	–	ns
27	Pulse Width, AS High*	PW_{ASH}	100	–	80	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	90	–	70	–	ns
29	Usable Access Time* (See Note 3)	t_{ACC}	530	–	410	–	ns

* At specified cycle time.

FIGURE 7 — BUS TIMING



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

INTRODUCTION

The MC68701U4 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into seven different operating modes. The operating mode controls the configuration of 18 of the 40 MCU pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a data register and a write-only data direction register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port" by itself refers to all of the hardware associated with the port. When the port is used as a "data port" or "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port pins are labeled as Pij where i identifies one of four ports and j indicates the particular bit.

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

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

3

FIGURE 8 — PROGRAMMING MODEL

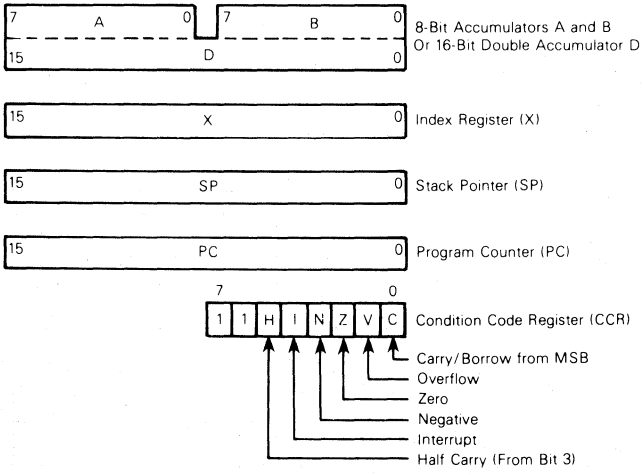


TABLE 1 — NEW INSTRUCTIONS

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

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

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

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

OPERATING MODES

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

FUNDAMENTAL MODES

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

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

TABLE 2 — SUMMARY OF OPERATING MODES

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

FIGURE 9 — SINGLE-CHIP MODE

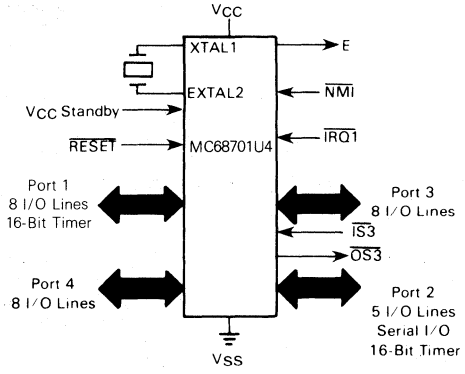
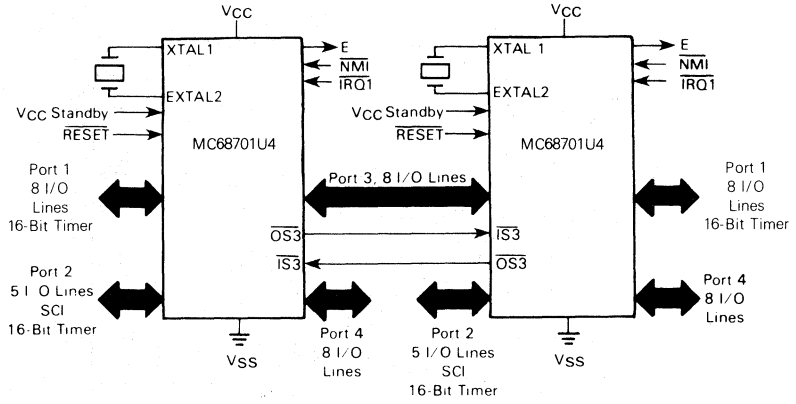


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION



EXPANDED NON-MULTIPLEXED MODE (5) — A modest amount of external memory space is provided in the expanded non-multiplexed mode while significant on-chip resources are retained. Port 3 functions as an 8-bit bidirectional data bus and port 4 is configured initially as an input data port. Any combination of the eight least significant address lines may be obtained by writing to the port 4 data direction register. Stated alternatively, any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Internal pullup resistors pull the port 4 lines high until the port is configured.

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

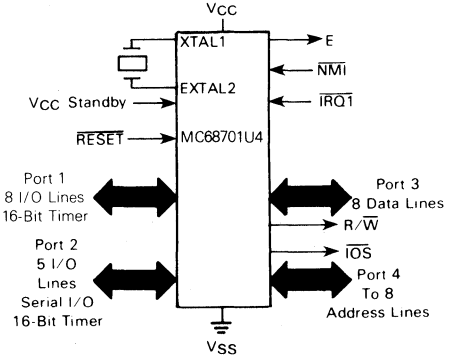
EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) — A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E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

In mode 0, the reset and interrupt vectors are located at $BFF0-$BFFF. In addition, the internal and external data

buses are connected, so there must be no memory map overlap in order to avoid potential bus conflicts. By writing the PC0-PC2 bits in the port 2 data register, modes 5, 6, and 7 can be irreversibly entered from mode 0. Mode 0 is used to program the on-chip EPROM.

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

FIGURE 11 — EXPANDED NON-MULTIPLEXED CONFIGURATION



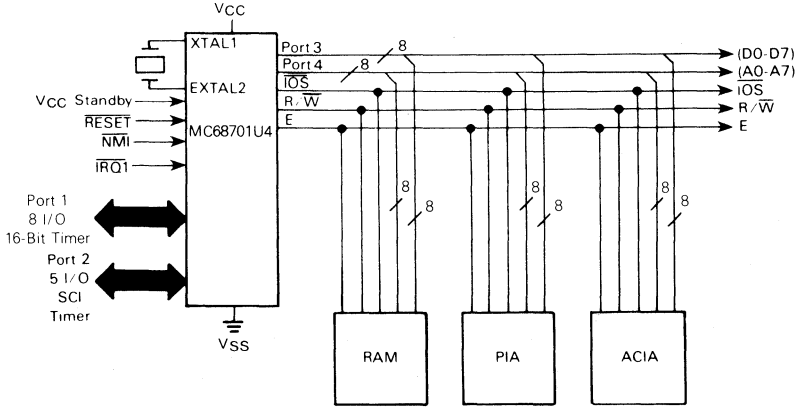
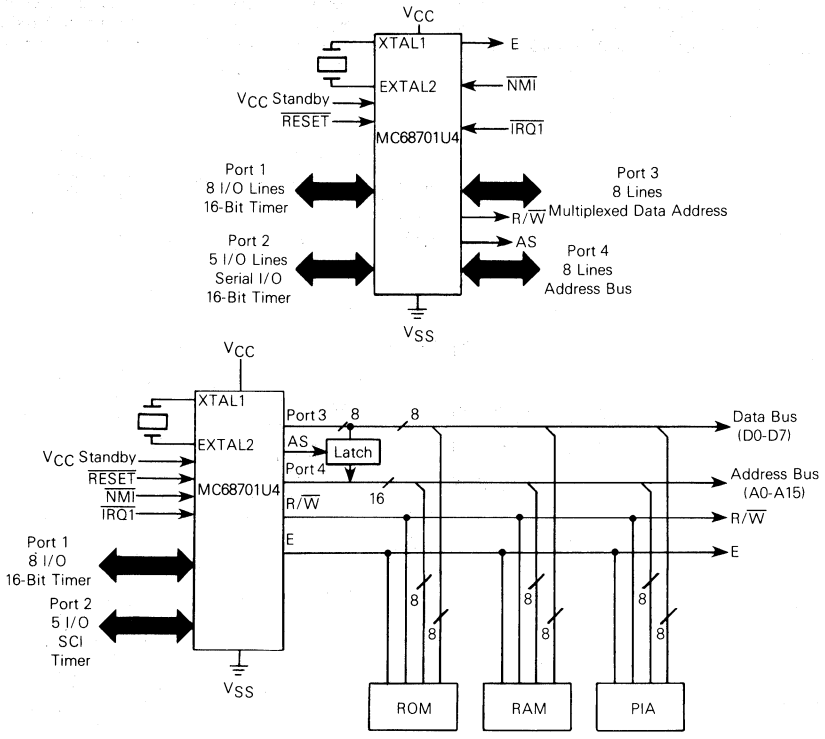
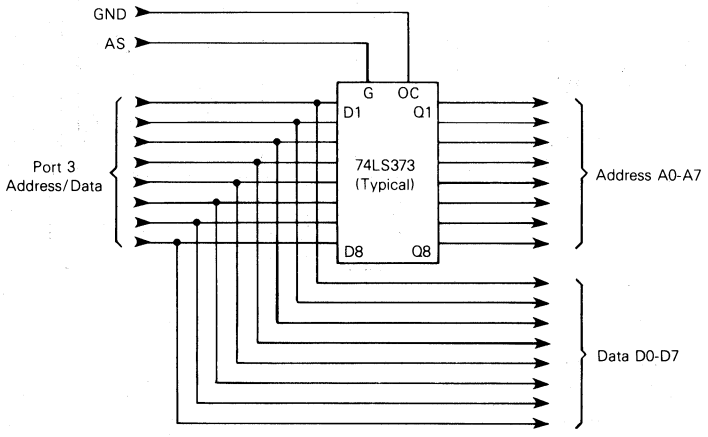


FIGURE 12 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 13 — TYPICAL LATCH ARRANGEMENT



PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

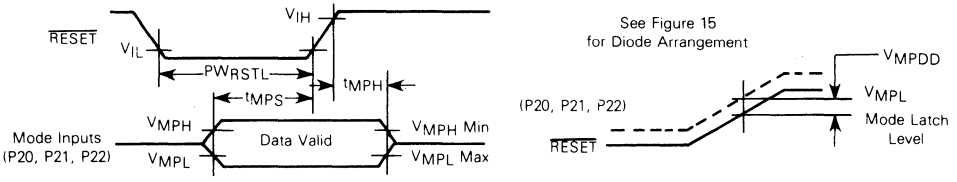
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 15 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

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

FIGURE 14 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 14)

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

TABLE 3 — MODE SELECTION SUMMARY

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

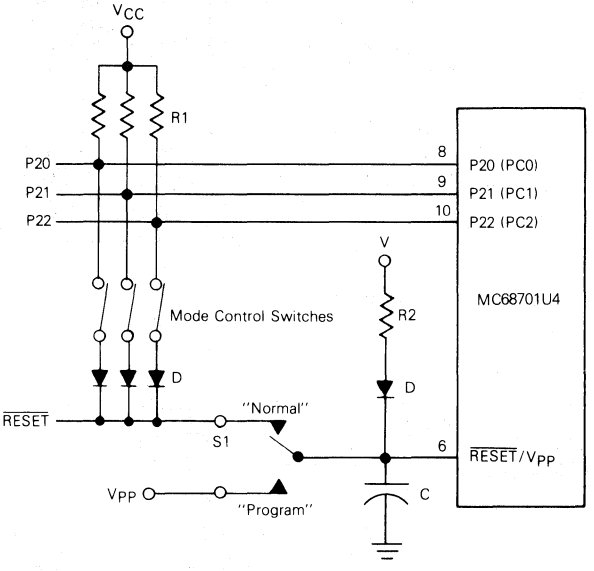
LEGEND

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

NOTES:

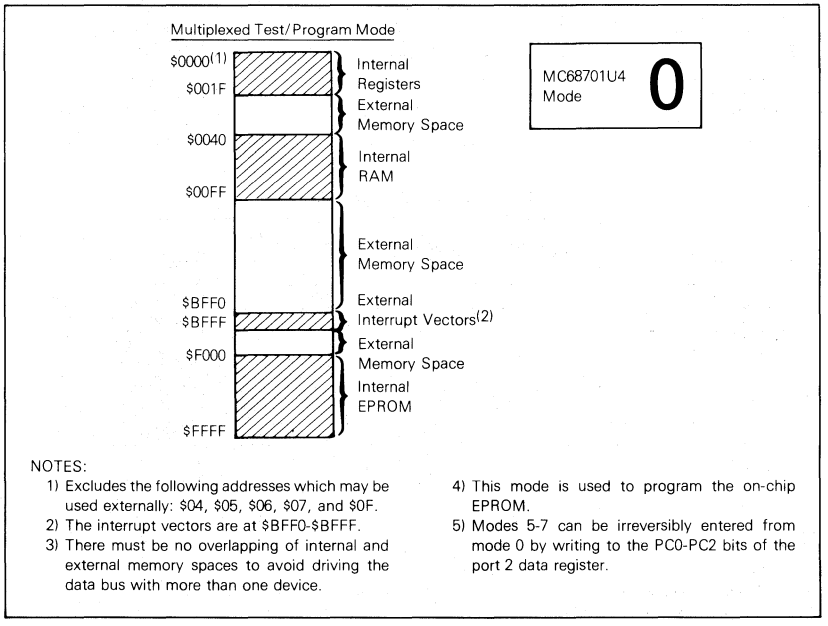
1. Addresses associated with ports 3 and 4 are considered external in modes 0, 2, and 3.
2. Addresses associated with port 3 are considered external in modes 1, 5, and 6.
3. Port 4 default is user data input; address output is optional by writing to port 4 data direction register.
4. Mode 4 is a non-user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

FIGURE 15 — TYPICAL MODE PROGRAMMING CIRCUIT



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

FIGURE 16 — MEMORY MAPS (Sheet 1 of 3)

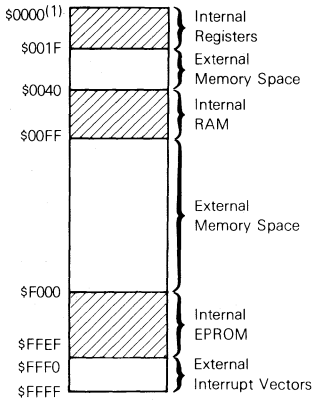


NOTES:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) The interrupt vectors are at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is used to program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

FIGURE 16 — MEMORY MAPS (Sheet 2 of 3)

MC68701U4 Mode 1

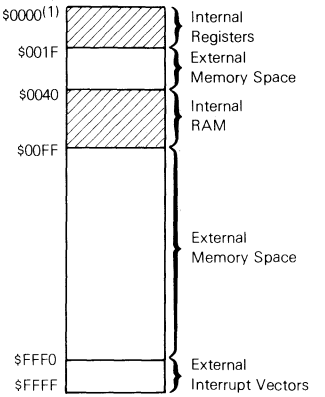
Multiplexed/RAM and EPROM



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

MC68701U4 Mode 2

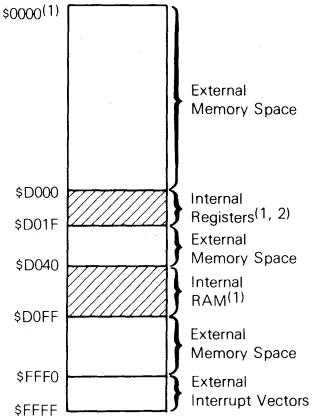
Multiplexed/RAM



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

MC68701U4 Mode 3

Multiplexed/RAM

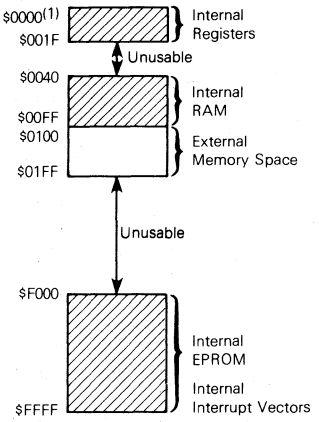


NOTES:
1) Relocating the internal registers and the internal RAM to high memory allows the processor to make use of direct addressing.
2) Excludes the following addresses which may be used externally: $D004, $D005, $D006, $D007 and $D00F.

FIGURE 16 — MEMORY MAPS (Sheet 3 of 3)

MC68701U4 Mode 5

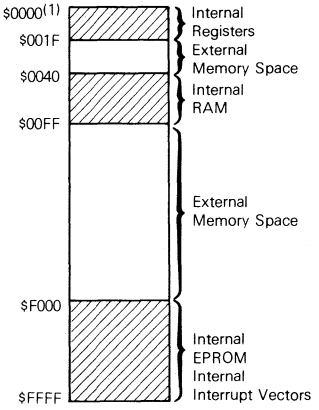
Non-Multiplexed/Partial Decode



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

MC68701U4 Mode 6

Multiplexed/Partial Decode



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

MC68701U4 Mode 7

Single Chip

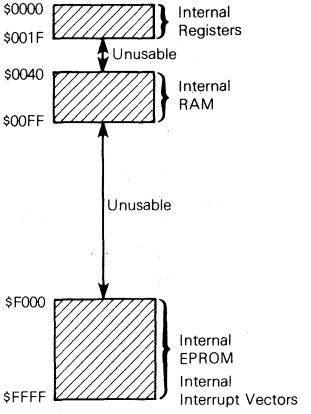


TABLE 4 — INTERNAL REGISTER AREA

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Counter Alternate Address (High Byte)	15
Counter Alternate Address (Low Byte)	16
Timer Control Register 1	17
Timer Control Register 2	18
Timer Status Register	19
Output Compare Register 2 (High Byte)	1A
Output Compare Register 2 (Low Byte)	1B
Output Compare Register 3 (High Byte)	1C
Output Compare Register 3 (Low Byte)	1D
Input Capture Register 2 (High Byte)	1E
Input Capture Register 2 (Low Byte)	1F

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

MC68701U4 INTERRUPTS

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

NOTE

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

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

The interrupt flowchart is depicted in Figure 17 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, index register, A accumulator, B accumulator, and condition code register are pushed to the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{RESET}$ timing are illustrated in Figures 18 and 19.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

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

FIGURE 17 – INTERRUPT FLOWCHART

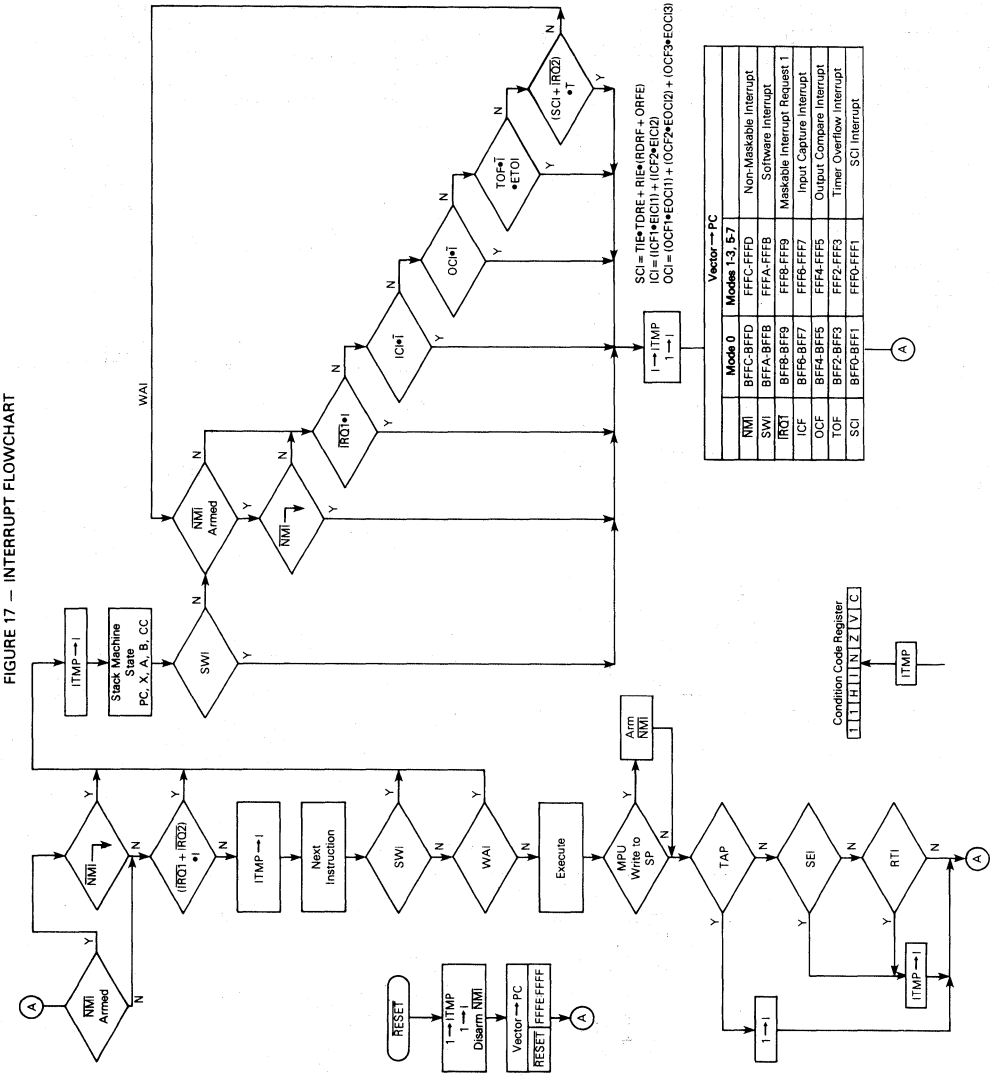


FIGURE 18 — INTERRUPT SEQUENCE

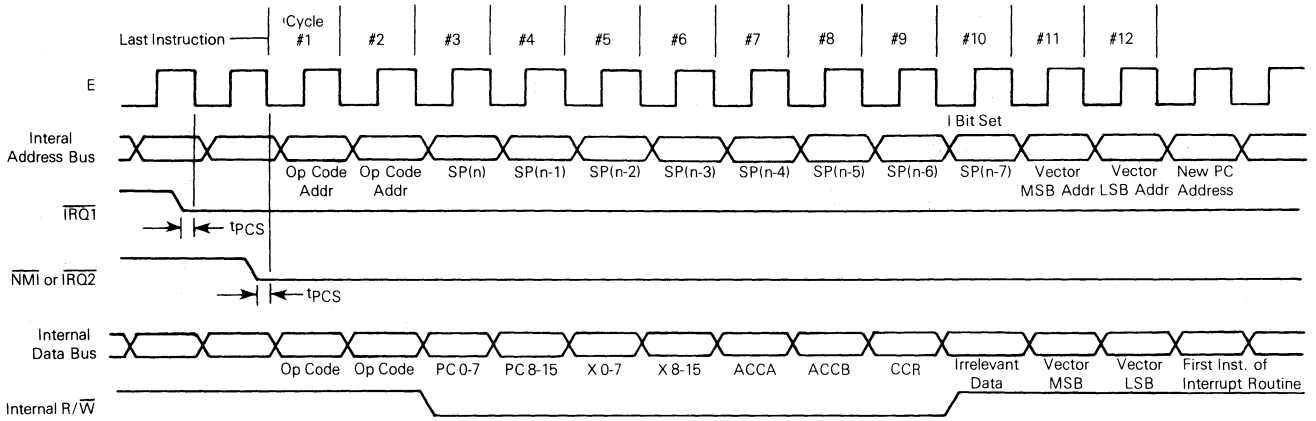
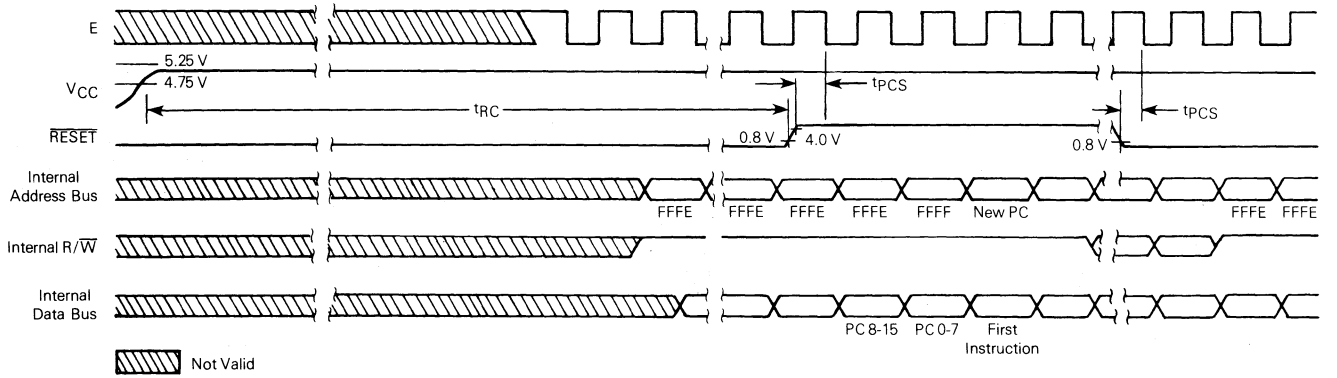


FIGURE 19 — RESET TIMING



FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

V_{CC} standby provides power to the standby portion ($40 through $5F) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide +5 volts (±5%) and must reach V_{SB} volts before $\overline{RESET}$ reaches 4.0 volts. During power down, V_{CC} standby must remain above V_{SBB} (minimum) to sustain the standby RAM and STBY PWR bit. While in power-down operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during power-down operation.

XTAL1 AND EXTAL 2

These two input pins interface either a crystal or TTL-compatible clock to the MCU internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz or 4.4336 MHz color burst TV crystals. A 20 pF capacitor should be tied from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL2 may be driven by an external TTL-compatible clock at 4 f_o with a duty cycle of 50% (±5%) with XTAL1 connected ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator operated in parallel resonance mode in the frequency range specified for f_{XTAL}. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals. Nominal crystal parameters are shown in Figure 20.

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

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{RESET}$ must be held below 0.8 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} standby reaches 4.75 volts. $\overline{RESET}$ must be held low at least three E cycles if asserted during power-up operation.

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

E (ENABLE)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the device input clock frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

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

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

NOTE

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

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

$\overline{IRQ1}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin an interrupt sequence. A vector is fetched from $FFF8 and $FFF9 ($BFF8 and $BFF9 in mode 0), transferred to the program counter, and instruction execution is resumed.

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

SC1 and SC2 (STROBE CONTROL 1 AND 2)

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

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

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

SC1 AND SC2 IN EXPANDED NON-MULTIPLEXED MODE – In the expanded non-multiplexed mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

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

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

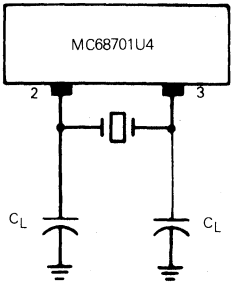
FIGURE 20 — OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

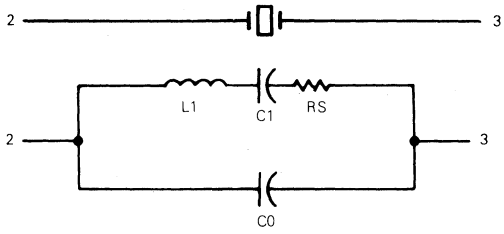
Nominal Crystal Parameters*

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

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



C_L = 20 pF (typical)

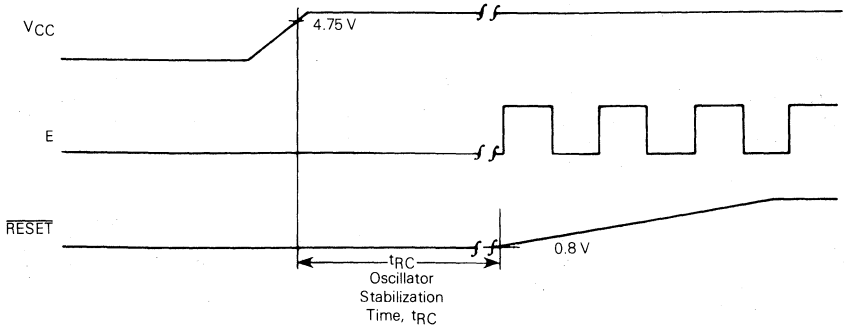


Equivalent Circuit

NOTE
TTL-compatible oscillators may be obtained from:

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

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



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

P10-P17 (PORT 1)

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

P20-P24 (PORT 2)

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

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

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

PORT 2 DATA REGISTER

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

P30-P37 (PORT 3)

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

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

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

PORT 3 CONTROL AND STATUS REGISTER

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

Bits 0-2 Not used.

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

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

Bit 5 Not used.

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

Bit 7 **IS3 Flag** – This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 control and status register (with IS3 flag set) followed by a read or write to the port 3 data register or during reset.

PORT 3 IN EXPANDED NON-MULTIPLEXED MODE – Port 3 is configured as a bidirectional data bus (D7-D0) in the expanded non-multiplexed mode. The direction of data transfers is controlled by read/write (SC2). Data is clocked by E (enable).

PORT 3 IN EXPANDED MULTIPLEXED MODE – Port 3 is configured as a time multiplexed address (A7-A0) and data bus (D7-D0) in the expanded multiplexed mode where address strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent bus conflicts.

P40-P47 (PORT 4)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF, and is the only port with internal pullup resistors. Unused lines can remain unconnected.

PORT 4 IN SINGLE-CHIP MODE – In single-chip mode, port 4 functions as an 8-bit I/O port with each line configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5-volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

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

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

RESIDENT MEMORY

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

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

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

The RAM is controlled by the RAM/EPROM control register.

RAM/EPROM CONTROL REGISTER ($14)

The RAM/EPROM control register includes four bits: STBY PWR, RAME, PLC, and PPC. Two of these bits, STBY PWR and RAME, are used to control RAM access and determine the adequacy of the standby power source during power-down operation. It is intended that RAME be cleared and STBY PWR be set as part of a power-down procedure. RAME and STBY PWR are read/write bits.

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

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

A description of the RAM/EPROM control register follows.

RAM/EPROM CONTROL REGISTER

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

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

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

PLC = 1 — EPROM address latch is transparent.

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

PPC = 0 — EPROM programming power (V_{PP}) applied.

PPC = 1 — EPROM programming power (V_{PP}) is not applied.

Bit 2-5 Unused.

Bit 6 **RAM Enable (RAME).** This read/write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during reset provided standby power is available on the positive edge of reset. If RAME is clear, any access to a RAM address is external. If RAME is set, the RAM is included in the internal map.

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

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

ERASING THE MC68701U4 EPROM

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

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

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

PROGRAMMING THE MC68701U4 EPROM

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

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

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

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

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

PROGRAMMABLE TIMER

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

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

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

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

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

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

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

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

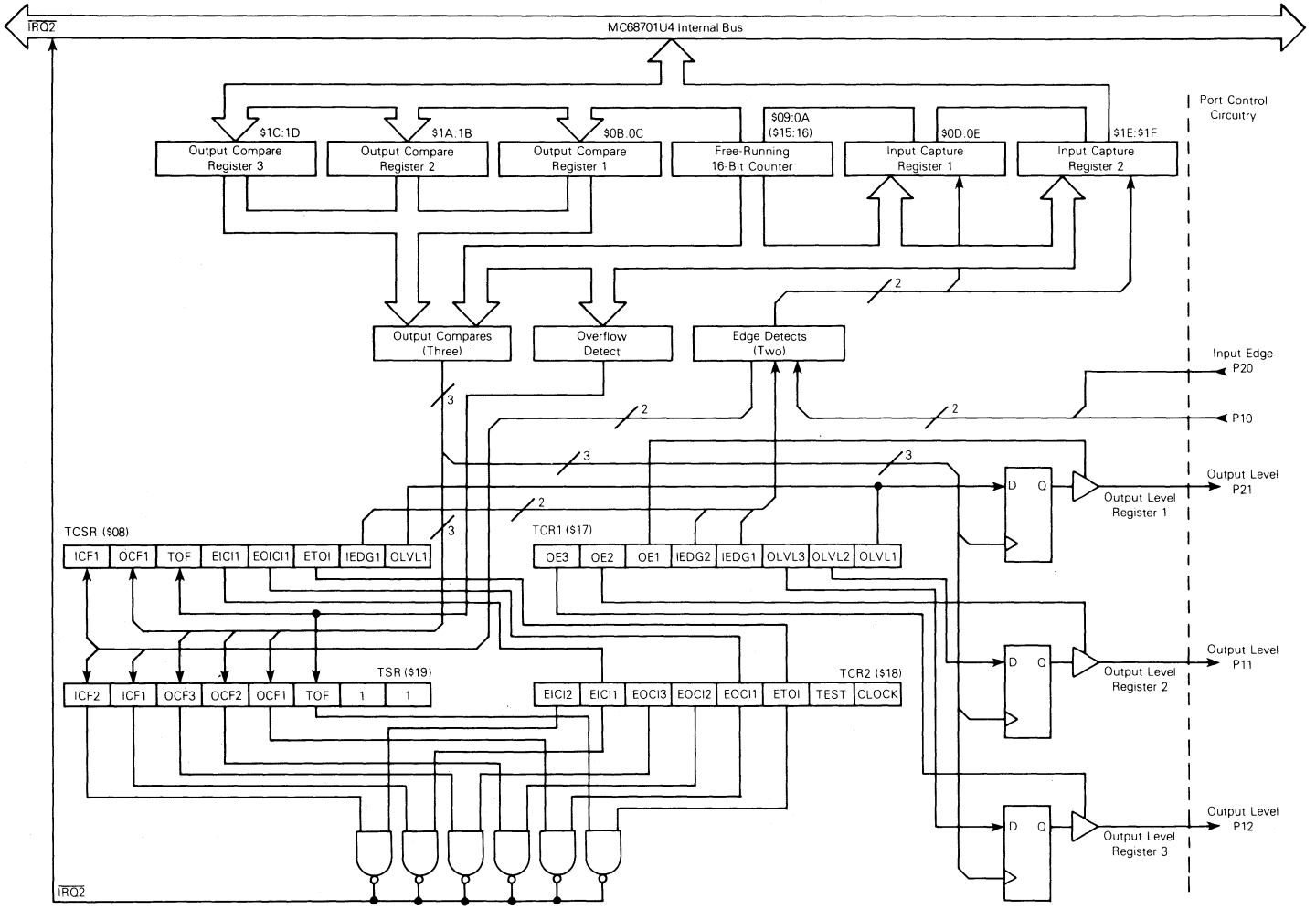
An input capture can occur independently of ICF; the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTERS

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

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

FIGURE 21 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



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

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

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

TIMER CONTROL AND STATUS REGISTER

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

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

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

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

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

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

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

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

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

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

TIMER CONTROL REGISTER 1

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

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

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

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

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

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

Bit 5 **Output Enable 1** — OE1 is set during reset and enables the contents of output level register 1 to be connected to P21 when bit 1 of port 2 data direction register is set.
OE1 = 0 port 2 bit 1 data register output
OE1 = 1 output level register 1

Bit 6 **Output Enable 2** — OE2 is cleared during reset and enables the contents of output level register 2 to be connected to P11 when bit 1 of port 1 data direction register is set.
OE2 = 0 port 1 bit 1 data register output
OE2 = 1 output level register 2

Bit 7 **Output Enable 3** – OE3 is cleared during reset and enables the contents of output level register 3 to be connected to P12 when bit 2 of port 1 data direction register is set
OE3 = 0 port 1 bit 2 data register output
OE3 = 1 output level register 3

TIMER CONTROL REGISTER 2 (TCR2) ($18) – Timer control register 2 is an 8-bit read/write register (except bits 0 and 1) which enable the interrupts associated with the free-running counter, the output compare registers, and the input capture registers. In test mode 0, two more bits (clock and test) are available for checking the timer.

TIMER CONTROL REGISTER 2
(Non-Test Modes)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	1	1	$18

Bits 0-1 **Read-Only Bits** – When read, these bits return a value of 1. Refer to **TIMER CONTROL REGISTER 2 (Test Mode)**.

Bit 2 **Enable Timer Overflow Interrupt** – When set, an $\overline{IRQ2}$ interrupt will be generated when the timer overflow flag is set; when clear, the interrupt is inhibited. ETOI is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Enable Output Compare Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 1 is set; when clear, the interrupt is inhibited. EOCI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Enable Output Compare Interrupt 2** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 2 is set; when clear, the interrupt is inhibited. EOCI2 is cleared during reset.

Bit 5 **Enable Output Compare Interrupt 3** – When set, an $\overline{IRQ2}$ interrupt will be generated when the output compare flag 3 is set; when clear, the interrupt is inhibited. EOCI3 is cleared during reset.

Bit 6 **Enable Input Capture Interrupt 1** – When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 1 is set; when clear, the interrupt is inhibited. EICI1 is cleared during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Enable Input Capture Interrupt 2** – When set, an $\overline{IRQ2}$ interrupt will be generated when the input capture flag 2 is set; when clear, the interrupt is inhibited. EICI2 is cleared during reset.

The timer test bits (test and clock) allow the free-running counter to be tested as two separate 8-bit counters to speed testing.

TIMER CONTROL REGISTER 2
(Test Mode)

7	6	5	4	3	2	1	0	
EICI2	EICI1	EOCI3	EOCI2	EOCI1	ETOI	TEST	CLOCK	$18

Bit 0 **CLOCK** – The CLOCK control bit selects which half of the 16-bit free-running counter (MSB or LSB) should be clocked with E. The CLOCK bit is a read/write bit only in mode 0 and is set during reset.
CLOCK = 0 – Only the eight most significant bits of the free-running counter run with TEST = 0.
CLOCK = 1 – Only the eight least significant bits of the free-running counter run when TEST = 0.

Bit 1 **TEST** – the TEST control bit enables the timer test mode. TEST is a read/write bit in mode 0 and is set during reset.
TEST = 0 – Timer test mode enabled:
a) The timer LSB latch is transparent which allows the LSB to be read independently of the MSB.
b) Either the MSB or the LSB of the timer is clocked by E, as defined by the CLOCK bit.
TEST = 1 – Timer test mode disabled.

Bits 2-7 See **TIMER CONTROL REGISTER 2 (Non-Test Modes)**. (These bits function the same as in the non-test modes.)

TIMER STATUS REGISTER (TSR) ($19) – The timer status register is an 8-bit read-only register which contains the flags associated with the free-running counter, the output compare registers, and the input capture registers.

TIMER STATUS REGISTER

7	6	5	4	3	2	1	0	
ICF2	ICF1	OCF3	OCF2	OCF1	TOF	1	1	$19

Bits 0-1 Not used.

Bit 2 **Timer Overflow Flag** – The TOF is set when the counter contains all ones ($FFFF). It is cleared by reading the TSR or the TCSR (with TOF set) and then the counter high byte ($09), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 3 **Output Compare Flag 1** – OCF1 is set when output compare register 1 matches the free-running counter. OCF1 is cleared by reading the TSR or the TCSR (with OCF1 set) and then writing to output compare register 1 ($0B or $0C), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 4 **Output Compare Flag 2** – OCF2 is set when output compare register 2 matches the free-running counter. OCF2 is cleared by reading the TSR (with OCF2 set) and then writing to output compare register 2 ($1A or $1B), or during reset.

Bit 5 **Output Compare Flag 3** – OCF3 is set when output compare register 3 matches the free-running counter. OCF3 is cleared by reading the TSR (with OCF3 set) and then writing to output compare register 3 ($1C or $1D), or during reset.

Bit 6 **Input Capture Flag 1** – ICF1 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR or the TCSR (with ICF1 set) and the input capture register 1 high byte ($0D), or during reset. Refer to **TIMER CONTROL AND STATUS REGISTER (TCSR) ($08)**.

Bit 7 **Input Capture Flag 2** — ICF2 is set to indicate that a proper level transition has occurred; it is cleared by reading the TSR (with ICF2 set) and the input capture register 2 high byte ($1E), or during reset.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a variety of rates. The SCI transmitter and receiver are functionally independent but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and bi-phase and both provide one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

WAKE-UP FEATURE

In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to permit uninterested MPUs to ignore the remainder of the message, wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive ones or during reset. Software must provide for the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

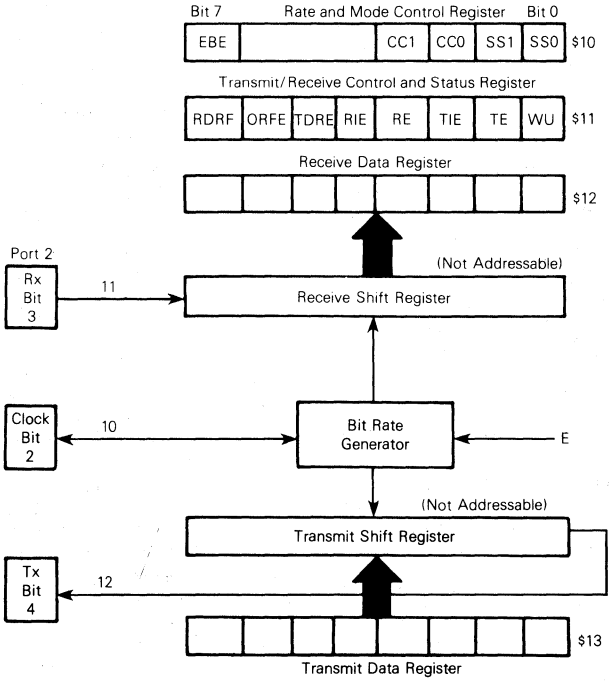
The following features of the SCI are programmable:

- Format: standard mark/space (NRZ) or bi-phase
- Clock: external or internal bit rate clock
- Baud: one of eight per E clock frequency or external clock (×8 desired baud)
- Wake-Up Feature: enabled or disabled
- Interrupt Requests: enabled individually for transmitter and receiver
- Clock Output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in Figure 22. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

FIGURE 22 — SCI REGISTERS



RATE AND MODE CONTROL REGISTER (RMCR) ($10) — The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of five write-only bits which are cleared during reset. The two least significant bits in conjunction with bit 7 control the bit rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER

7	6	5	4	3	2	1	0	
EBE	X	X	X	CC1	CC0	SS1	SS0	$10

Bit 1:Bit 0 **SS1:SS0 Speed Select** — These two bits select the baud when using the internal clock. Eight rates may be selected (in conjunction with bit 7) which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3:Bit 2 **CC1:CC0 Clock Control and Format Select** — These two bits control the format and select the serial clock source. If CC1 is set, the DDR value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the formats, clock source, and use of P22.

Bits 4-6 Not used.

Bit 7 **EBE Enhanced Baud Enable** — EBE selects the standard MC6801 baud rates when clear and the additional baud rates when set (Table 6). This bit is cleared by reset and is a write-only control bit.

EBE = 0 standard MC6801 baud rates
EBE = 1 additional baud rates

If both CC1 and CC0 are set, an external TTL-compatible clock must be connected to P22 at eight times (8×) the desired bit rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P22 regardless of the values for TE or RE.

NOTE

The source of SCI internal bit rate clock is the timer free-running counter. An MPU write to the counter in mode 0 can disturb serial operations.

TABLE 6 — SCI BIT TIMES AND RATES

EBE	SS1:SS0		4 f_o →	2.4576 MHz		4.0 MHz		4.9152 MHz	
				614.4 kHz		1.0 MHz		1.2288 MHz	
			E	Baud	Time	Baud	Time	Baud	Time
0	0	0	÷ 16	38400.0	26 μs	62500.0	16.0 μs	76800.0	13.0 μs
0	0	1	÷ 128	4800.0	208.3 μs	7812 5	128.0 μs	9600.0	104.2 μs
0	1	0	÷ 1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 μs
0	1	1	÷ 4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
1	0	0	÷ 64	9600.0	104.2 μs	15625.0	64 μs	19200.0	52.0 μs
1	0	1	÷ 256	2400.0	416.6 μs	3906.3	256 μs	4800.0	208.3 μs
1	1	0	÷ 512	1200.0	833.3 μs	1953.1	512 μs	2400.0	416.6 μs
1	1	1	÷ 2048	300.0	3.33 ms	488.3	2.05 ms	600.0	01.67 ms
External (P22)*				76800.0	13.0 μs	125000.0	8.0 μs	153600.0	6.5 μs

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
00	Bi-Phase	Internal	Not Used
01	NRZ	Internal	Not Used
10	NRZ	Internal	Output
11	NRZ	External	Input

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) — The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{\text{RESET}}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11

Bit 0 **"Wake-Up" on Idle Line** — When set, WU enables the wake-up function; it is cleared by ten consecutive ones or during reset. WU will not be set if the line is idle. Refer to **WAKE-UP FEATURE.**

Bit 1 **Transmit Enable** — When set, P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive ones is transmitted. TE is cleared during reset.

Bit 2 **Transmit Interrupt Enable** — When set, an $\overline{\text{IRQ2}}$ is set; when clear, the interrupt is inhibited. TE is cleared during reset.

Bit 3 **Receive Enable** — When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared during reset.

Bit 4 **Receiver Interrupt Enable** — When set, an $\overline{\text{IRQ2}}$ interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared during reset.

Bit 5 **Transmit Data Register Empty** — TDRE is set when the transmit data register is transferred to the output serial shift register or during reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the transmit data register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 **Overrun Framing Error** — If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the stop bit (1) is not found in the tenth bit time. An overrun can be distinguished from a framing error by the state of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. Unframed data causing a framing error is transferred to the receive data register. However, subsequent data transfer is blocked until the framing error flag is cleared. ORFE is cleared by reading the TRCSR (with ORFE set) then the receive data register, or during reset.

Bit 7 **Receive Data Register Full** — RDRF is set when the input serial shift register is transferred to the receive data register, or during reset.

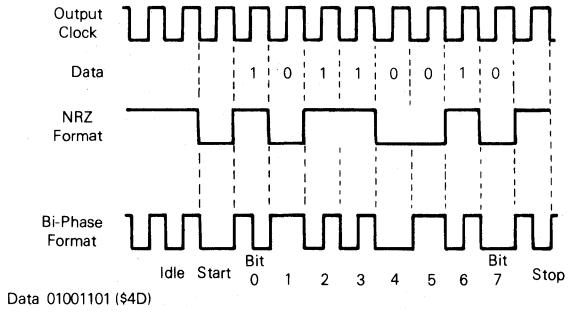
SERIAL OPERATIONS

The SCI is initialized by writing control bytes first to the rate and mode control register and then to the transmit/receive control and status register. When TE is set, the output of the transmit serial shift register is connected to P24 and serial output is initiated by transmitting a 9-bit preamble of ones.

At this point, one of two situations exists: 1) if the transmit data register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line; or 2) if a byte has been written to the transmit data register (TDRE = 0), it will be transferred to the output serial shift register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer occurs, ones will be sent until more data is provided. In bi-phase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 23.

FIGURE 23 — SCI DATA FORMATS



Data 01001101 ($4D)

INSTRUCTION SET

The MC68701U4 is directly source compatible with the MC6801 and upward source and object code-compatible with the MC6800. Execution times of key instructions have been reduced and several instructions have been added, including a hardware multiply. A list of new operations added to the MC6800 instruction set is shown in Table 1.

In addition, two special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the program counter to increment like a 16-bit counter causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 8. There are 220 valid machine codes, 34 unassigned codes, and 2 codes reserved for test purposes.

TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	A0	SUBA	INDXD	4	2	D4	ANDB		3	2
05	ASLD		3	1	39	RTS		5	1	6D	TST		6	2	A1	CMPA		4	2	D5	BITB		3	2
06	TAP		2	1	3A	ABX		3	1	6E	JMP		3	2	A2	SBCA		4	2	D6	LDAB		3	2
07	TPA		2	1	3B	RTI		10	1	6F	CLR	INDXD	6	2	A3	SUBD		6	2	D7	STAB		3	2
08	INX		3	1	3C	PSHX		4	1	70	NEG	EXTND	6	3	A4	ANDA		4	2	D8	EORB		3	2
09	DEX		3	1	3D	MUL		10	1	71	•				A5	BITA		4	2	D9	ADCB		3	2
0A	CLV		2	1	3E	WAI		9	1	72	•				A6	LDAA		4	2	DA	ORAB		3	2
0B	SEV		2	1	3F	SWI		12	1	73	COM		6	3	A7	STAA		4	2	DB	ADDB		3	2
0C	CLC		2	1	40	NEGA		2	1	74	LSR		6	3	A8	EORA		4	2	DC	LDD		4	2
0D	SEC		2	1	41	•				75	•				A9	ADCA		4	2	DD	STD		4	2
0E	CLI		2	1	42	•				76	ROR		6	3	AA	ORAA		4	2	DE	LDX		4	2
0F	SEI		2	1	43	COMA		2	1	77	ASR		6	3	AB	ADDA		4	2	DF	STX	DIR	4	2
10	SBA		2	1	44	LSRA		2	1	78	ASL		6	3	AC	CPX		6	2	E0	SUBB	INDXD	4	2
11	CBA		2	1	45	•				79	ROL		6	3	AD	JSR		6	2	E1	CMPB		4	2
12	•				46	RORA		2	1	7A	DEC		6	3	AE	LDS		5	2	E2	SBCB		4	2
13	•				47	ASRA		2	1	7B	•				AF	STS	INDXD	5	2	E3	ADDD		6	2
14	•				48	ASLA		2	1	7C	INC		6	3	B0	SUBA	EXTND	4	3	E4	ANDB		4	2
15	•				49	ROLA		2	1	7D	TST		6	3	B1	CMPA		4	3	E5	BITB		4	2
16	TAB		2	1	4A	DECA		2	1	7E	JMP		3	3	B2	SBCA		4	3	E6	LDAB		4	2
17	TBA		2	1	4B	•				7F	CLR	EXTND	6	3	B3	SUBD		6	3	E7	STAB		4	2
18	•				4C	INCA		2	1	80	SUBA	IMMED	2	2	B4	ANDA		4	3	E8	EORB		4	2
19	DAA	INHER	2	1	4D	TSTA		2	1	81	CMPA		2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•				4E	T				82	SBCA		2	2	B6	LDAA		4	3	EA	ORAB		4	2
1B	ABA	INHER	2	1	4F	CLRA		2	1	83	SUBD		4	3	B7	STAA		4	3	EB	ADDB		4	2
1C	•				50	NEGB		2	1	84	ANDA		2	2	B8	EORA		4	3	EC	LDD		5	2
1D	•				51	•				85	BITA		2	2	B9	ADCA		4	3	ED	STD		5	2
1E	•				52	•				86	LDAA		2	2	BA	ORAA		4	3	EE	LDX		5	2
1F	•				53	COMB		2	1	87	•				BB	ADDA		4	3	EF	STX	INDXD	5	2
20	BRA	REL	3	2	54	LSRB		2	1	88	EORA		2	2	BC	CPX		6	3	F0	SUBB	EXTND	4	3
21	BRN		3	2	55	•				89	ADCA		2	2	BD	JSR		6	3	F1	CMPB		4	3
22	BHI		3	2	56	RORB		2	1	8A	ORAA		2	2	BE	LDS		5	3	F2	SBCB		4	3
23	BLS		3	2	57	ASRB		2	1	8B	ADDA		2	2	BF	STS	EXTND	5	3	F3	ADDD		6	3
24	BCC		3	2	58	ASLB		2	1	8C	CPX	IMMED	4	3	C0	SUBB	IMMED	2	2	F4	ANDB		4	3
25	BCS		3	2	59	ROLB		2	1	8D	BSR	REL	6	2	C1	CMPB		2	2	F5	BITB		4	3
26	BNE		3	2	5A	DECB		2	1	8E	LDS	IMMED	3	3	C2	SBCB		2	2	F6	LDAB		4	3
27	BEQ		3	2	5B	•				8F	•				C3	ADDD		4	3	F7	STAB		4	3
28	BVC		3	2	5C	INCB		2	1	90	SUBA	DIR	3	2	C4	ANDB		2	2	F8	EORB		4	3
29	BVS		3	2	5D	TSTB		2	1	91	CMPA		3	2	C5	BITB		2	2	F9	ADCB		4	3
2A	BPL		3	2	5E	T				92	SBCA		3	2	C6	LDAB		2	2	FA	ORAB		4	3
2B	BMI		3	2	5F	CLRB	INHER	2	1	93	SUBD		5	2	C7	•				FB	ADDB		4	3
2C	BGE		3	2	60	NEG	INDXD	6	2	94	ANDA		3	2	C8	EORB		2	2	FC	LDD		5	3
2D	BLT		3	2	61	•				95	BITA		3	2	C9	ADCB		2	2	FD	STD		5	3
2E	BGT		3	2	62	•				96	LDAA		3	2	CA	ORAB		2	2	FE	LDX		5	3
2F	BLE	REL	3	2	63	COM		6	2	97	STAA		3	2	CB	ADDB		2	2	FF	STX	EXTND	5	3
30	TSX	INHER	3	1	64	LSR		6	2	98	EORA		3	2	CC	LDD		3	3					
31	INS		3	1	65	•				99	ADCA		3	2	CD	•								
32	PULA		4	1	66	ROR		6	2	9A	ORAA		3	2	CE	LDX	IMMED	3	3					
33	PULB		4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

* UNDEFINED OP CODE

NOTES:

1. Addressing Modes
   INHER ≡ Inherent INDXD ≡ Indexed IMMED ≡ Immediate
   REL ≡ Relative EXTND ≡ Extended DIR ≡ Direct
2. Unassigned opcodes are indicated by "•" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

PROGRAMMING MODEL

A programming model for the MC68701U4 is shown in Figure 8. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulators A and/or B. Other registers are defined as follows:

PROGRAM COUNTER – The program counter is a 16-bit register which always points to the next instruction.

STACK POINTER – The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random-access memory at a location defined by the programmer.

INDEX REGISTER – The index register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

ACCUMULATORS – The MPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

CONDITION CODE REGISTER – The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I bit) and inhibits all maskable interrupts when set. The two unused bits, B6 and B7, are read as ones.

ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 9, 10, 11, and 12 where execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, one E cycle is equivalent to one microsecond. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and descriptions of selected instructions are shown in Figure 24.

IMMEDIATE ADDRESSING – The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

DIRECT ADDRESSING – The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

EXTENDED ADDRESSING – The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

INDEXED ADDRESSING – The unsigned offset contained in the second byte of the instruction is added with carry to the index register and is used to reference memory without changing the index register. These are two byte instructions.

INHERENT ADDRESSING – The operand(s) is a register and no memory reference is required. These are single byte instructions.

RELATIVE ADDRESSING – Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of −126 to +129 bytes from the first byte of the instruction. These are two byte instructions.

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 14 provides a detailed description of the information present on the address bus, data bus, and the read/write ($R/\overline{W}$) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. "High order" byte refers to the most significant byte of a 16-bit value. During unused bus cycles, the address bus is forced to $FFFF and $R/\overline{W}$ is high.

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	MNEM	Immed Op	Immed ~	Immed #	Direct Op	Direct ~	Direct #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inherent Op	Inherent ~	Inherent #	Boolean/ Arithmetic Operation	Condition Codes 5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M:M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pointer	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pointer	INS													31	3	1	1 SP + 1 → SP	•	•	•	•	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H,(M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H,(M + 1) → SP_L	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M,X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M,SP_L → (M + 1)	•	•	↕	↕	R	•
Index Reg → Stack Pointer	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Register	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP},SP − 1 → SP X_H → M_{SP},SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP,M_{SP} → X_H SP + 1 → SP,M_{SP} → X_L	•	•	•	•	•	•

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

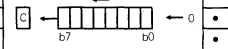
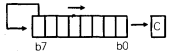
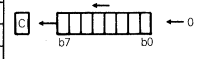
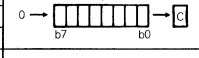
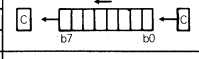
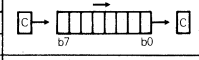
Accumulator and Memory Operations	MNEM	Immed Op	Immed ~	Immed #	Direct Op	Direct ~	Direct #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inher Op	Inher ~	Inher #	Boolean Expression	Condition Codes 5 H	4 I	3 N	2 Z	1 V	0 C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A•M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B•M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Double	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A•M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B•M	•	•	↕	↕	R	•
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				M → M	•	•	↕	↕	R	S
	COMA													43	2	1	A → A	•	•	↕	↕	R	S
	COMB													53	2	1	B → B	•	•	↕	↕	R	S

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

Accumulator and Memory Operations		Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
																		5	4	3	2	1	0
	MNEM	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Decimal Adjust, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Accumulators	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	2		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕
Multiply	MUL													3D	10	1	A × B → D	•	•	•	•	•	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 − M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	↕	↕
	ROLA													49	2	1		•	•	↕	↕	↕	↕
	ROLB													59	2	1		•	•	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	↕	↕
	RORA													46	2	1		•	•	↕	↕	↕	↕
	RORB													56	2	1		•	•	↕	↕	↕	↕
Subtract Accumulator	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Store Accumulators	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D − M:M + 1 → D	•	•	↕	↕	↕	↕
Transfer Accumulator	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct			Relative			Index			Extend			Inherent			Branch Test	Condition Code Reg.					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3				See Special Operations-Figure 24	•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations-Figure 24	•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

Operations	Inherent				Boolean Operation	Condition Code Register					
	MNEM	Op	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND

Op Operation Code (Hexadecimal)
~ Number of MPU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
M Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
• Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	●	●	●	●	2	●
ABX	●	●	●	●	3	●
ADC	2	3	4	4	●	●
ADD	2	3	4	4	●	●
ADDD	4	5	6	6	●	●
AND	2	3	4	4	●	●
ASL	●	●	6	6	2	●
ASLD	●	●	●	●	3	●
ASR	●	●	6	6	2	●
BCC	●	●	●	●	●	3
BCS	●	●	●	●	●	3
BEQ	●	●	●	●	●	3
BGE	●	●	●	●	●	3
BGT	●	●	●	●	●	3
BHI	●	●	●	●	●	3
BHS	●	●	●	●	●	3
BIT	2	3	4	4	●	●
BLE	●	●	●	●	●	3
BLO	●	●	●	●	●	3
BLS	●	●	●	●	●	3
BLT	●	●	●	●	●	3
BMI	●	●	●	●	●	3
BNE	●	●	●	●	●	3
BPL	●	●	●	●	●	3
BRA	●	●	●	●	●	3
BRN	●	●	●	●	●	3
BSR	●	●	●	●	●	6
BVC	●	●	●	●	●	3
BVS	●	●	●	●	●	3
CBA	●	●	●	●	2	●
CLC	●	●	●	●	2	●
CLI	●	●	●	●	2	●
CLR	●	●	6	6	2	●
CLV	●	●	●	●	2	●
CMP	2	3	4	4	●	●
COM	●	●	6	6	2	●
CPX	4	5	6	6	●	●
DAA	●	●	●	●	2	●
DEC	●	●	6	6	2	●
DES	●	●	●	●	3	●
DEX	●	●	●	●	3	●
EOR	2	3	4	4	●	●
INC	●	●	6	6	●	●
INS	●	●	●	●	3	●

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
JMP	●	●	3	3	●	●
JSR	●	5	6	6	●	●
LDA	2	3	4	4	●	●
LDD	3	4	5	5	●	●
LDS	3	4	5	5	●	●
LDX	3	4	5	5	●	●
LSL	●	●	6	6	2	●
LSLD	●	●	●	●	3	●
LSR	●	●	6	6	2	●
LSRD	●	●	●	●	3	●
MUL	●	●	●	●	10	●
NEG	●	●	6	6	2	●
NOP	●	●	●	●	2	●
ORA	2	3	4	4	●	●
PSH	●	●	●	●	3	●
PSHX	●	●	●	●	4	●
PUL	●	●	●	●	4	●
PULX	●	●	●	●	5	●
ROL	●	●	6	6	2	●
ROR	●	●	6	6	2	●
RTI	●	●	●	●	10	●
RTS	●	●	●	●	5	●
SBA	●	●	●	●	2	●
SBC	2	3	4	4	●	●
SEC	●	●	●	●	2	●
SEI	●	●	●	●	2	●
SEV	●	●	●	●	2	●
STA	●	3	4	4	●	●
STD	●	4	5	5	●	●
STS	●	4	5	5	●	●
STX	●	4	5	5	●	●
SUB	2	3	4	4	●	●
SUBD	4	5	6	6	●	●
SWI	●	●	●	●	12	●
TAB	●	●	●	●	2	●
TAP	●	●	●	●	2	●
TBA	●	●	●	●	2	●
TPA	●	●	●	●	2	●
TST	●	●	6	6	2	●
TSX	●	●	●	●	3	●
TXS	●	●	●	●	3	●
WAI	●	●	●	●	9	●

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 1 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Operand Data
LDS LDX LDD	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Address Bus FFFF	1 1 1 1	Opcode Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Opcode Address Opcode Address + 1 Address of Operand	1 1 1	Opcode Address of Operand Operand Data
STA	3	1 2 3	Opcode Address Opcode Address + 1 Destination Address	1 1 0	Opcode Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Opcode Address Opcode Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Opcode Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Opcode Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Subroutine Address Stack Pointer Stack Pointer − 1	1 1 1 0 0	Opcode Irrelevant Data First Subroutine Opcode Return Address (Low Order Byte) Return Address (High Order Byte)

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 2 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Jump Address (High Order Byte)
		3	Opcode Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Address of Operand
AND ORA		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Destination Address (High Order Byte)
		3	Opcode Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
STD		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Opcode Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Opcode Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Address of Subroutine (High Order Byte)
		3	Opcode Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

* TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 3 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Opcode Address	1	Opcode
ADD LDA		2	Opcode Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Opcode Address	1	Opcode
LDX		2	Opcode Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Opcode Address	1	Opcode
STX		2	Opcode Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Opcode Address	1	Opcode
ASR NEG		2	Opcode Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Opcode Address	1	Opcode
SUBD		2	Opcode Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Opcode
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

*TST does not perform the write cycle during the sixth cycle. The sixth cycle is another address bus = $FFFF.

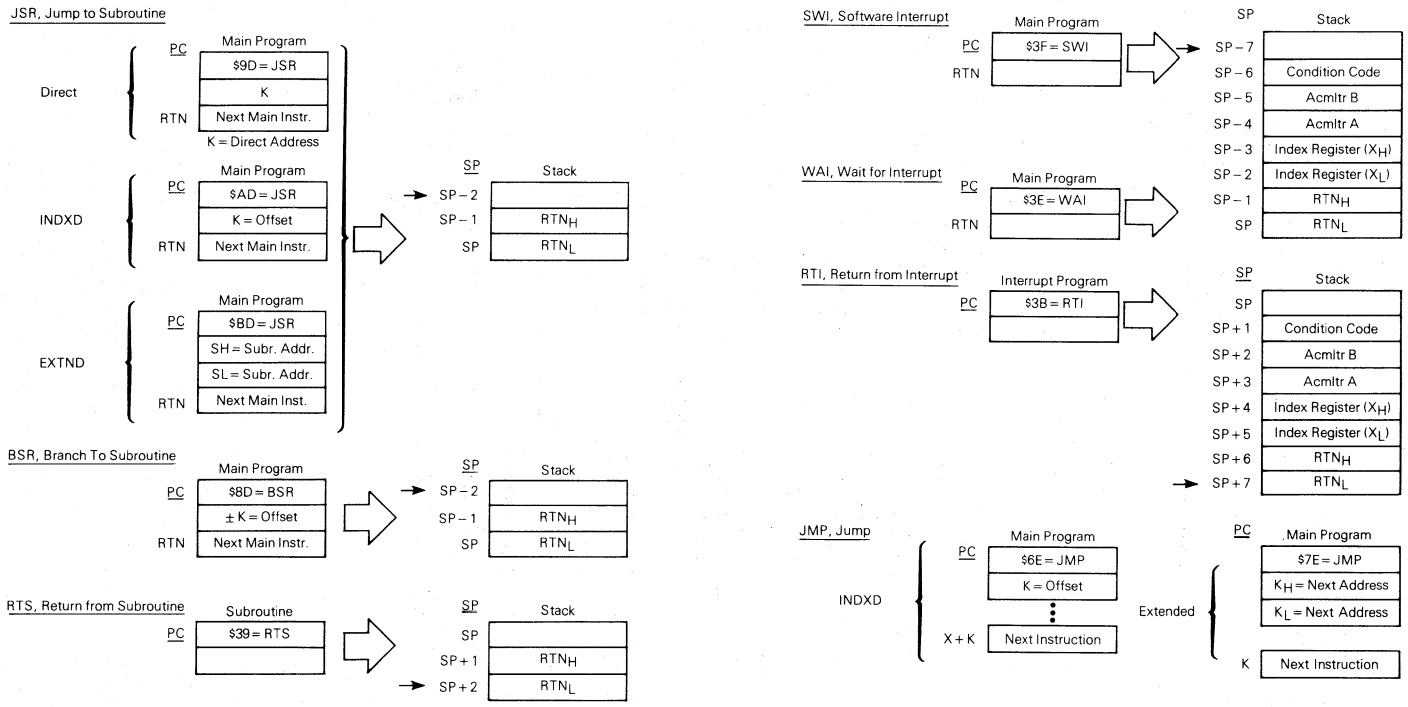
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 4 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Opcode Address Opcode Address + 1	1 1	Opcode Opcode of Next Instruction
ABX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Opcode Address Opcode Address + 1 Previous Stack Pointer Contents	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 0	Opcode Opcode of Next Instruction Accumulator Data
TSX	3	1 2 3	Opcode Address Opcode Address + 1 Stack Pointer	1 1 1	Opcode Opcode of Next Instruction Irrelevant Data
TXS	3	1 2 3	Opcode Address Opcode Address + 1 Address Bus FFFF	1 1 1	Opcode Opcode of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Opcode Opcode of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1	1 1 0 0	Opcode Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Opcode Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Opcode Address Opcode Address + 1 Stack Pointer Stack Pointer − 1 Stack Pointer − 2 Stack Pointer − 3 Stack Pointer − 4 Stack Pointer − 5 Stack Pointer − 6	1 1 0 0 0 0 0 0 0	Opcode Opcode of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Condition Code Register

TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 5)

Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT (Continued)					
MUL	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Condition Code Register from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer − 1	0	Return Address (High Order Byte)
		5	Stack Pointer − 2	0	Index Register (Low Order Byte)
		6	Stack Pointer − 3	0	Index Register (High Order Byte)
		7	Stack Pointer − 4	0	Contents of Accumulator A
		8	Stack Pointer − 5	0	Contents of Accumulator B
		9	Stack Pointer − 6	0	Contents of Condition Code Register
		10	Stack Pointer − 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMI BVS	3	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Buss FFFF	1	Low Byte of Restart Vector
BSR	6	1	Opcode Address	1	Opcode
		2	Opcode Address + 1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Opcode of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer − 1	0	Return Address (High Order Byte)

FIGURE 24 — SPECIAL OPERATIONS



Legend:
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address
→ = Stack Pointer After Execution
K = 8-bit Unsigned Value

MOTOROLA

MC68705P3

Advance Information

8-BIT EPROM MICROCOMPUTER UNIT

The MC68705P3 Microcomputer Unit (MCU) is an EPROM member of the M6805 Family of low-cost single-chip microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turn-around time for prototype evaluation of the mask ROM versions. This 8-bit microcomputer contains a CPU, on-chip CLOCK, EPROM, bootstrap ROM, RAM, I/O, and a TIMER.

Because of these features, the MC68705P3 offers the user an economical means of designing an M6805 Family MCU into his system, either as a prototype evaluation, as a low-volume production run, or a pilot production run.

HARDWARE FEATURES:

- 8-Bit Architecture
- 112 bytes of RAM
- Memory Mapped I/O
- 1804 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
  - External Timer Interrupt
- Vectored Interrupts — External, Timer, and Software
- Zero-Cross Detection on $\overline{INT}$ Input
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Generator
- Master and Power-On Reset
- Complete Development System Support on EXORciser
- Emulates the MC6805P2 and MC6805P4 (Except for V_{SB})
- Bootstrap Program in ROM Simplifies EPROM Programming

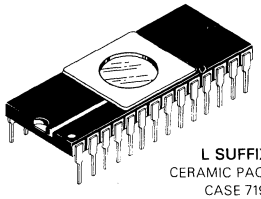
SOFTWARE FEATURES:

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to EPROM, RAM, and I/O

HMOS

(HIGH-DENSITY, N-CHANNEL DEPLETION LOAD, 5 V EPROM PROCESS)

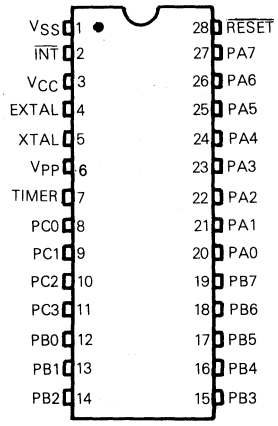
8-BIT EPROM MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719

S SUFFIX
CERDIP PACKAGE
ALSO AVAILABLE

PIN ASSIGNMENT

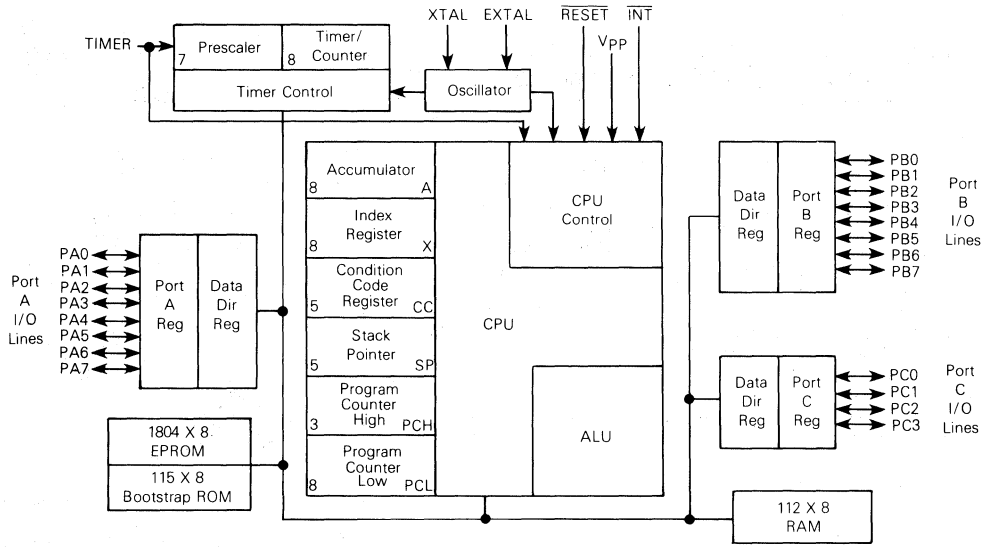


GENERIC INFORMATION
(f = 1.0 MHz, T_A = 0 to 70°C)

Package Type	Generic Number
Ceramic L Suffix	MC68705P3L
Cerdip S Suffix	MC68705P3S

This document contains information on a new product. Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage EPROM Programming Voltage (V_{PP} Pin) TIMER Pin	V_{PP}	−0.3 to +22.0	V
Normal Mode	V_{in}	−0.3 to +7.0	V
Bootstrap Programming Mode	V_{in}	−0.3 to +15.0	V
All Others	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature	T_J	+150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Package	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.25 Vdc ±0.5, V_{SS} = 0 Vdc, T_A = 20° to 30°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage (V_{PP} Pin)	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 5.25 V V_{PP} = 21.0 V	I_{PP}	– –	– –	8 30	mA
Programming Oscillator Frequency	f_{oscp}	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) I_{in} = 100 µA Max	V_{IHTP}	9.0	12.0	15.0	V

SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	f_{osc}	0.4	–	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	µs
$\overline{INT}$ or Timer Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Cap = 1.0 µF)	t_{RHL}	100	–	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Duty Cycle (EXTAL) (See Figure 12)	–	40	50	60	%

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage $\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) $\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) All Other	V_{IH}	4.0 V_{CC} – 0.5 4.0 V_{CC} – 0.5 2.0	– – ** ** –	V_{CC} V_{CC} V_{CC} V_{CC} V_{CC}	V
Input High Voltage (TIMER Pin) Timer Mode Bootstrap Programming Mode	V_{IH}	2.0 9.0	– 12.0	V_{CC} 15.0	V
Input Low Voltage $\overline{RESET}$ $\overline{INT}$ All Other	V_{IL}	–0.3 –0.3 –0.3	– ** –	0.8 1.5 0.8	V
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V, T_A = 0°C)	P_{INT}	–	450	TBD	mW
Input Capacitance XTAL All Other	C_{in}	– –	25 10	– –	pF
$\overline{INT}$ Zero-Crossing Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	V_{acp-p}
$\overline{RESET}$ Hysteresis Voltage (See Figure 11) Out of Reset Voltage Into Reset Voltage	V_{IRES+} V_{IRES-}	2.1 0.8	– –	4.0 2.0	V
Programming Voltage (V_{PP} Pin) Programming EPROM Operating Mode	V_{PP}*	20.0 4.0	21.0 V_{CC}	22.0 5.75	V
Input Current TIMER (V_{in} = 0.4 V) $\overline{INT}$ (V_{in} = 0.4 V) EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option) (V_{in} = 0.4 V Crystal Option) $\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	I_{in}	– – – – –4.0	– 20 – – –	20 50 10 –1600 –40	µA

*V_{PP} is Pin 6 on the MC68705P3 and is connected to V_{CC} in the Normal Operating Mode. In the MC6805P2, Pin 6 is NUM and is connected to V_{SS} in the Normal Operating Mode. The user must allow for this difference when emulating the MC6805P2 ROM-based MCU.

**Due to internal biasing, this input (when not used) floats to approximately 2.0 V.

PORT ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = −10 μA	V_{OH}	V_{CC}−1.0	–	–	V
Input High Voltage, I_{Load} = −300 μA (Max)	V_{IH}	2.0	–	V_{CC}+0.7	V
Input Low Voltage, I_{Load} = −500 μA (Max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	−300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	−500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = −200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	−1.0	–	−10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC}+0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA
Port C					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = −100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}+0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

FIGURE 1 — TTL EQUIVALENT TEST LOAD (PORT B)

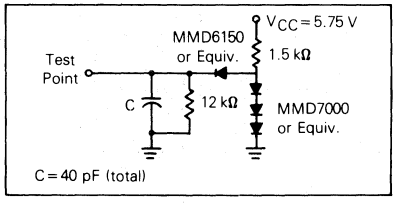


FIGURE 2 — CMOS EQUIVALENT TEST LOAD (PORT A)

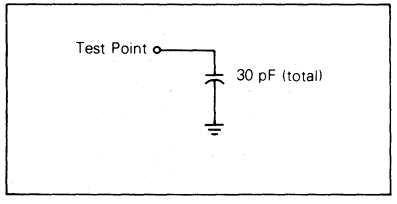
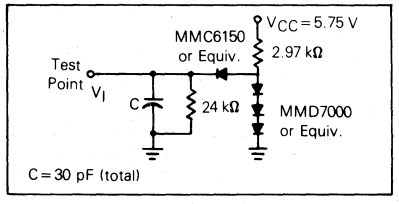


FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

V_{CC} and V_{SS} — Power is supplied to the MCU using two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$ — This pin allows an external event to asynchronously interrupt the processor. It can also be used as a polled input using the BIL and BIH instructions. Refer to INTERRUPTS for additional information.

XTAL and EXTAL — These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the CLK bit (see MASK OPTIONS), is 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 INTERNAL CLOCK GENERATOR OPTIONS for recommendations about these inputs.

TIMER — This is used as an external input to control the internal timer/circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program (see PROGRAMMING FIRMWARE). Refer to TIMER for additional information about the timer circuitry.

$\overline{RESET}$ — This pin has a Schmitt Trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to RESETS for additional information.

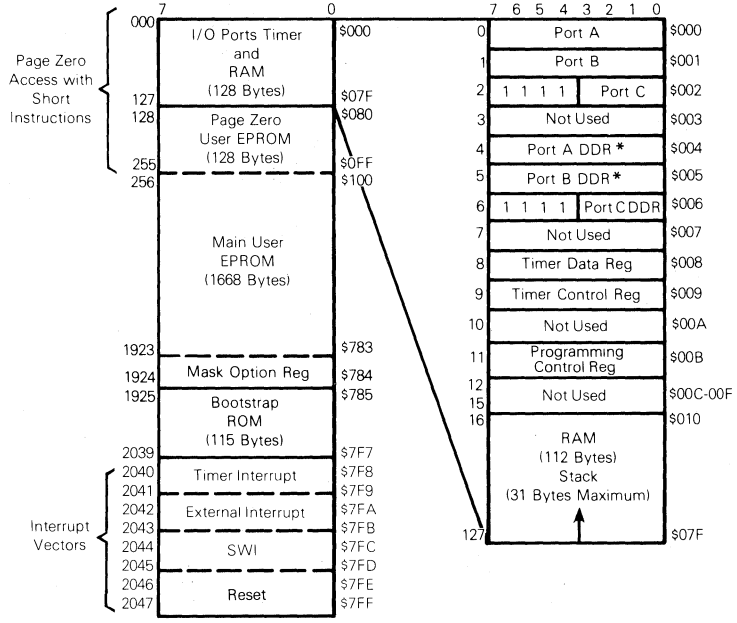
V_{PP} — This pin is used when programming the EPROM. By applying the programming voltage to this pin, one of the requirements is met for programming the EPROM. In normal operation, this pin is connected to V_{CC}. Refer to PROGRAMMING FIRMWARE and ELECTRICAL CHARACTERISTICS.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) — 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 (DDRs). Refer to INPUT/OUTPUT paragraphs for additional information, being sure to observe the Caution.

MEMORY

As shown in Figure 4, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC68705P3 MCU has implemented 2041 bytes of these locations. This consists of: 1804 bytes of user EPROM, 115 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM Mask Option Register (MOR), a Program Control Register (PCR), and eight bytes of I/O. The user EPROM is located in two areas. The main EPROM area is memory locations $080 to $783. The second area is reserved for eight interrupt/reset vector bytes at memory locations $7F8 to $7FF. The MCU uses nine of the lowest 16 memory locations for program control and I/O features such as ports, the port DDRs, and the timer. The Mask Option Register at memory location $784 completes the total. The 112 bytes of user RAM include up to 31 bytes for the stack.

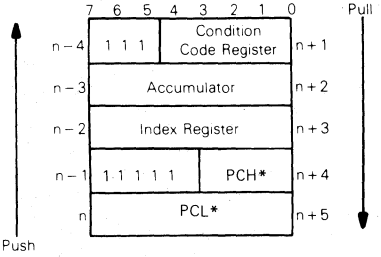
FIGURE 4 — MEMORY CONFIGURATION



Caution: Data Direction Registers (DDRs) are write-only; they read as $FF.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 5. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed.

FIGURE 5 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCH and PCL are stacked.

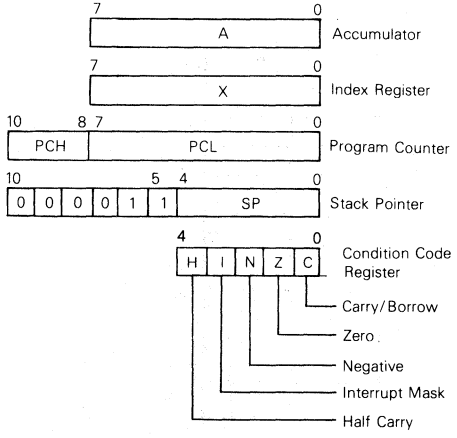
CENTRAL PROCESSING UNIT

The CPU of the M6805 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 address, data, and control buses.

REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 6 and are explained in the following paragraphs.

FIGURE 6 — PROGRAMMING MODEL



ACCUMULATOR (A) — The accumulator is a general purpose 8-bit register used to hold operands and results of the arithmetic calculations or data manipulations.

INDEX REGISTER (X) — The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC) — The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP) — The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the Reset Stack Pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC) — The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each of the five bits is explained below.

Half Carry (H) — Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I) — When this bit is set the timer and external interrupt ($\overline{INT}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C) — When set, this bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

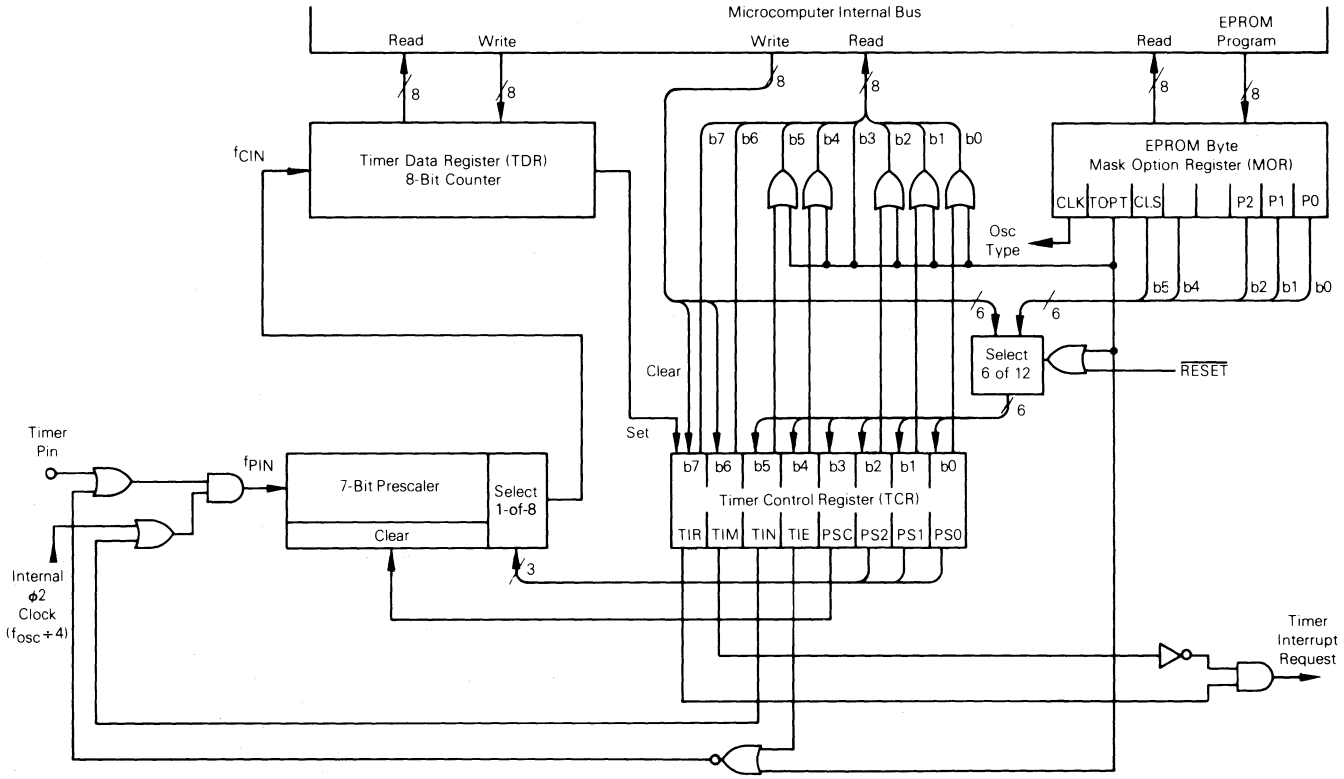
TIMER

The MC68705P3 MCU timer consists of an 8-bit software-programmable counter which is driven by a 7-bit software-programmable prescaler. Various timer clock sources may be selected ahead of the prescaler and counter. The timer selections are made via the Timer Control Register (TCR) and/or the Mask Option Register (MOR). The TCR also contains the interrupt control bits. The sections elsewhere entitled TIMER CONTROL REGISTER and MASK OPTIONS include additional details on controlling this timer.

The MCU timer circuitry is shown in Figure 7. The 8-bit counter may be loaded under program control and is decremented toward zero by the f_{CIN} counter input (output of the prescaler option selection). Once the 8-bit counter has decremented to zero, it sets the TIR (Timer Interrupt Request) bit 7 (b7 of TCR). The TIM (Timer Interrupt Mask) bit (b6) can be software set to inhibit the interrupt request, or software cleared to pass the interrupt request to the processor. When the I-bit in the Condition Code Register is cleared, the processor receives the Timer Interrupt. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9 and executing the interrupt routine. The processor is sensitive to the level of the timer interrupt request line; therefore if the interrupt is masked, the TIR bit may be cleared by software (e.g., BCLR) without generating an interrupt. When servicing a timer interrupt, the TIR bit MUST be cleared by the timer interrupt service routine software in order to clear the timer interrupt request.

The counter continues to count (decrement) after falling through to $FF from zero. Thus, the counter can be read at any time by the processor without disturbing the count. This allows a program to determine the length of time since the occurrence of a timer interrupt and does not disturb the counting process.

FIGURE 7 — TIMER FUNCTIONAL BLOCK DIAGRAM



f_{PIN} — Prescaler Input Frequency
f_{CIN} — Counter Input Frequency

Timer Control Register Bits:
- TIR — Timer Interrupt Request Status
- TIM — Timer Interrupt Mask
- TIN — Timer Input Select
- TIE — Timer External Input Enable
- PSC — Prescaler Clear
- PS2, PS1, PS0 — Prescaler Select

Mask Option Register Bits:
- CLK — Clock Oscillator Type
- TOPT — Timer Mask/Programmable Option
- CLS — Timer Clock Source
- P2, P1, P0 — Prescaler Option

NOTE: The TOPT bit in the Mask Option Register selects whether the timer is software programmable via the Timer Control Register or emulates the mask programmable parts via the MOR PROM byte.

The clock input to the timer can be from an external source (decrementing the counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER pin allowing the user to easily perform pulse-width measurements. (Note: When the MOR TOPT bit is set and the CLS bit is clear, an ungated $\phi 2$ clock input is obtained by tying the TIMER pin to V_{CC}.) The source of the clock input is selected via the TCR or the MOR as described later.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling option selects one of eight outputs on the 7-bit binary divider; one output bypasses prescaling. To avoid truncation errors, the prescaler is cleared when bit b3 of the TCR is written to a logic "1", when in the software controlled mode (TOPT = 0, more on these modes in later paragraphs); however, TCR bit b3 reads as a logic "0" when TOPT = 0 and as a "1" when TOPT = 1 to ensure proper operation with read-modify-write instructions (bit set and clear for example).

At Reset, the prescaler and counter are initialized to an all "1s" condition; the Timer Interrupt Request bit (TCR, b7) is cleared and the Timer Interrupt Request mask (TCR, b6) is set. TCR bits b0, b1, b2, b4, and b5 are initialized by the corresponding Mask Option Register (MOR) bits at Reset. They are then software selectable after Reset if TOPT = 0.

Note that the timer block diagram in Figure 7 reflects two separate timer control configurations: a) software controlled mode via the Timer Control Register (TCR), and b) MOR controlled mode to emulate a mask ROM version with the Mask Option Register. In the software controlled mode, all TCR bits are read/write, except bit b3 which is write-only (always reads as a logic "0"). In the MOR controlled mode, TCR bits b7 and b6 are read/write, the other six have no effect on a write and read as logic "1s". The two configurations provide the user with the capability to freely select timer options as well as accurately emulate the MC6805P2 and MC6805P4 mask ROM version. In the following paragraphs refer to Figure 9 as well as the TIMER CONTROL REGISTER and MASK OPTIONS sections.

The TOPT (Timer Option) bit (b6) in the Mask Option Register is EPROM programmed to a logical "0" to select the software controlled mode, which is described first. TCR bits b5, b4, b3, b2, b1, and b0 give the program direct control of the prescaler and input selection options.

The Timer Prescaler input (f_{PIN}) can be configured for three different operating modes, plus a disable mode, depending upon the value written to TCR control bits b4 and b5 (TIE and TIN).

When the TIE and TIN bits are programmed to "0", the timer input is from the internal clock ($\phi 2$) and TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

When TIE = 1 and TIN = 0, the internal clock and the TIMER input pin signals are ANDed to form the timer input f_{PIN}. This mode can be used to measure external pulse widths. The external pulse simply gates in the internal clock for the duration of the pulse. The accuracy of the count in this mode is $\pm$ one count.

When TIE = 0 and TIN = 1, no f_{PIN} input is applied to the prescaler and the timer is disabled.

When TIE and TIN are both programmed to a "1", the timer is from the external clock. The external clock can be used to count external events as well as provide an external frequency for generating periodic interrupts.

Bits b0, b1, and b2 in the TCR are program controlled to choose the appropriate prescaler output. The prescaling divides the f_{PIN} frequency by 1, 2, 4, etc. in binary multiples to 128 producing f_{CIN} frequency to the counter. The processor cannot write into or read from the prescaler; however, the prescaler is set to all "1s" by writing b3 of TCR to a "1", which allows for truncation-free counting.

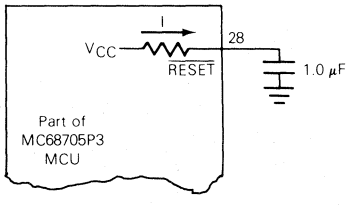
The MOR controlled mode of the timer is selected when the TOPT (Timer Option) bit (b6) in the MOR is programmed to a logical "1" to emulate the mask-programmable prescaler of the MC6805P2 and MC6805P4. The timer circuits are the same as described above; however, the Timer Control Register (TCR) is configured differently, as discussed below.

The logical level for the functions of bits b0, b1, b2, and b5 in the TCR are all determined at the time of EPROM programming. They are controlled by corresponding bits within the Mask Option Register (MOR, $784). The value programmed into MOR bits b0, b1, b2, and b5 controls the prescaler division and the timer clock selection. Bit b4 (TIE) and b3 (PSC) are set to a logical "1" in the MOR controlled mode. (When read by software, these six TCR bits always read as logical "1s".) As in the software programmable configuration, the TIM (b6) and TIR (b7) bits of the TCR are controlled by the counter and software as described above and in the TIMER CONTROL REGISTER section. The MOR controlled mode is designed to exactly emulate the MC6805P2 and MC6805P4 which has only TIM and TIR in the TCR and have the prescaler options defined as manufacturing mask options.

RESETS

The MCU can be reset in two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). Upon power-up, a delay of t_{RHL} is needed before allowing the $\overline{\text{RESET}}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{\text{RESET}}$ input, as shown in Figure 8, typically provides sufficient delay.

FIGURE 8 — POWER-UP RESET DELAY CIRCUIT



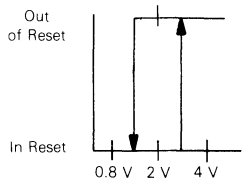
The internal circuit connected to the $\overline{\text{RESET}}$ pin consists of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. The Schmitt trigger provides an internal reset voltage when it senses logical "0" on the $\overline{\text{RESET}}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{\text{RESET}}$ pin voltage rises to V_{IRES+}. When the $\overline{\text{RESET}}$ pin voltage falls to a logical "0" for a period longer than one t_{CYC}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 9. See Figure 13 under INTERRUPTS for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The Mask Option Register (EPROM) is programmed to select crystal or resistor operation. The oscillator frequency is internally divided by four to produce the internal system clocks.

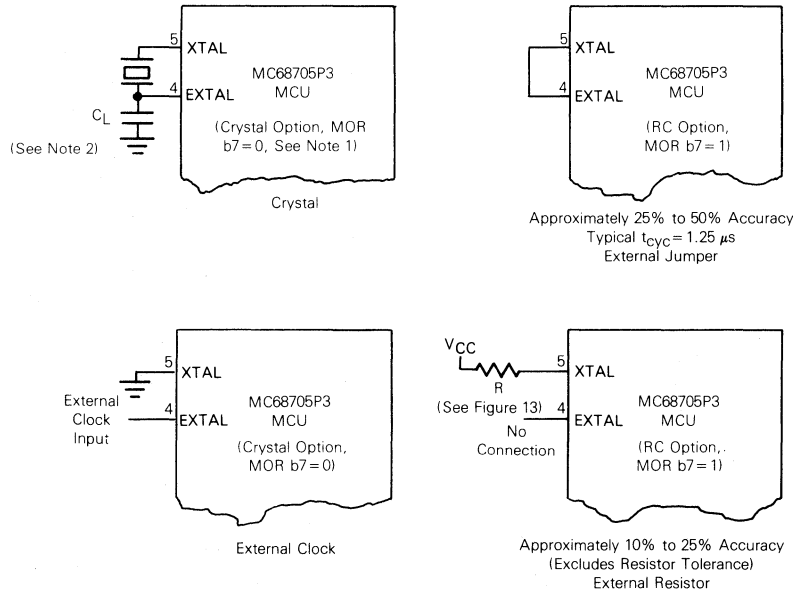
The different connection methods are shown in Figure 10.

FIGURE 9 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS



3

FIGURE 10 — CLOCK GENERATOR OPTIONS

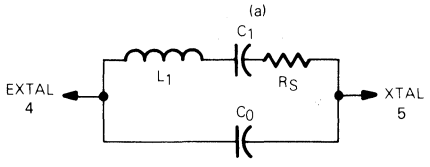


NOTES:

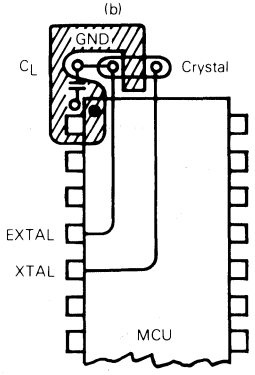
1. When the TIMER input pin is in the V_{IHTP} range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V_{CC}, the clock generator option is determined by bit 7 of the Mask Option Register (CLK).
2. The recommended C_L value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

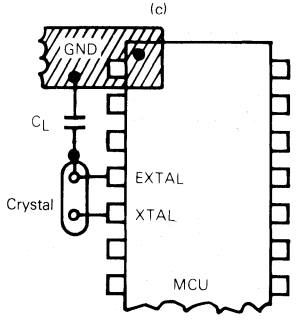
Crystal specifications and suggested PC board layouts are given in Figure 11. A resistor selection graph is given in Figure 12.

FIGURE 11 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



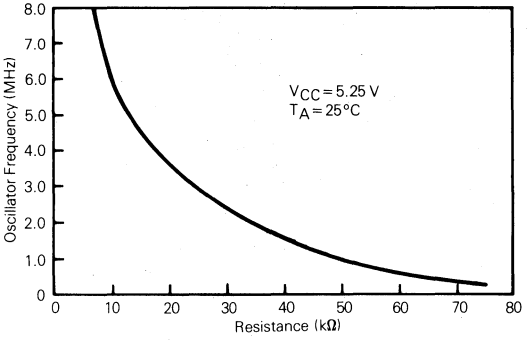
AT — Cut Parallel Resonance Crystal
C_O = 7 pF Max.
Freq. = 4.0 MHz @ C_L = 27 pF
R_S = 50 ohms Max.





NOTE: Keep crystal leads and circuit connections as short as possible.

FIGURE 12 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start-up neither the crystal characteristics nor the load capacitances should exceed recommendations.

BOOTSTRAP ROM

The bootstrap ROM contains a factory program which allows the MCU to fetch data from an external device and transfer it into the MC68705P3 EPROM. The bootstrap program provides: timing of programming pulses, timing of V_{PP} input, and verification after programming. See PROGRAMMING FIRMWARE section.

MASK OPTION REGISTER (MOR)

The Mask Option Register is an 8-bit user programmed (EPROM) register in which six of the bits are used. Bits in this register are used to select the type of system clock, the timer option, the timer/prescaler clock source, and the prescaler option. It is fully described in the MASK OPTIONS section.

INTERRUPTS

The MC68705P3 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I-bit, and vector fetching requires a total of 11 t_{CYC} periods for completion. A flowchart of the interrupt sequence is shown in

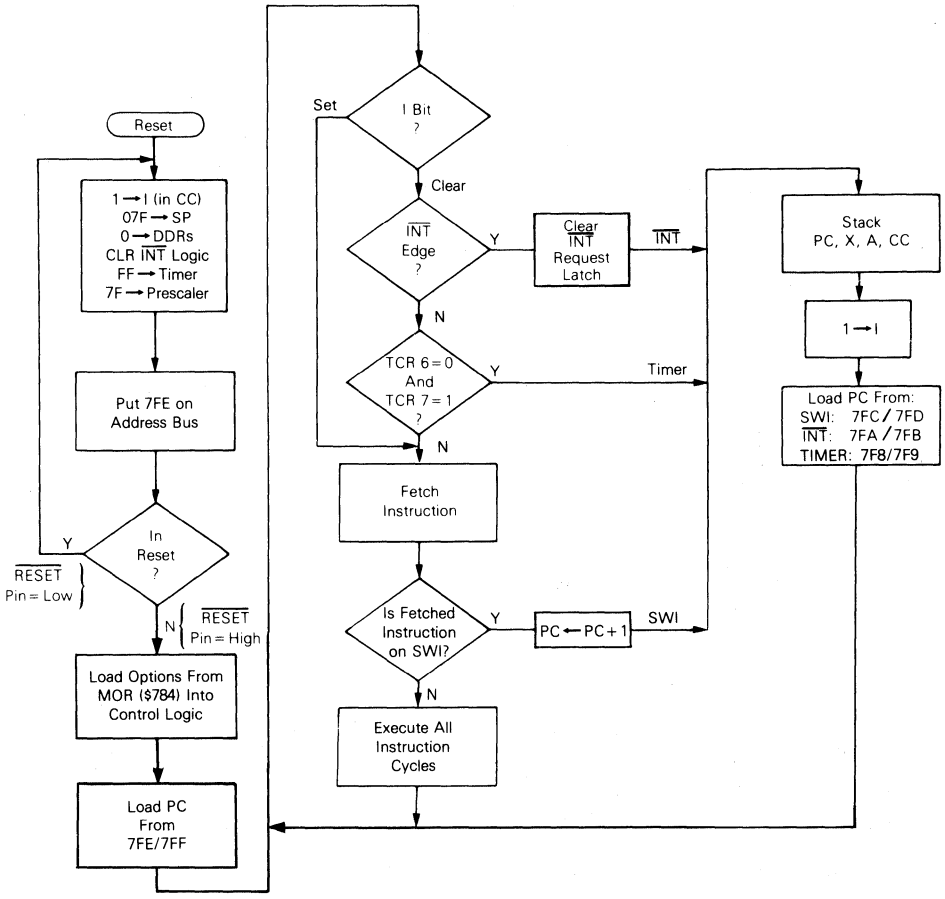
Figure 13. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{INT}}$. A sinusoidal input signal

FIGURE 13 — RESET AND INTERRUPT PROCESSING FLOWCHART



(f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 14(a), for use as a Zero-Crossing Detector (for negative transitions of AC sinusoid). This allows applications such as servicing time-of-day routines and engaging/disengaging AC power control devices. Off-chip full-wave rectification provides an interrupt at every zero crossing of the AC signal and thereby provides a 2f clock.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 14(b). For the $\overline{INT}$ function, the maximum allowable frequency is also determined by the software response of the $\overline{INT}$ service routine.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on Ports 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 "1" for output or a logic "0" for input. On Reset all the DDRs are initialized to a logic "0" state, placing the ports in the input mode. The port output registers are not initialized on Reset but may be written to before setting 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 15. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

FIGURE 14 — TYPICAL INTERRUPT CIRCUITS

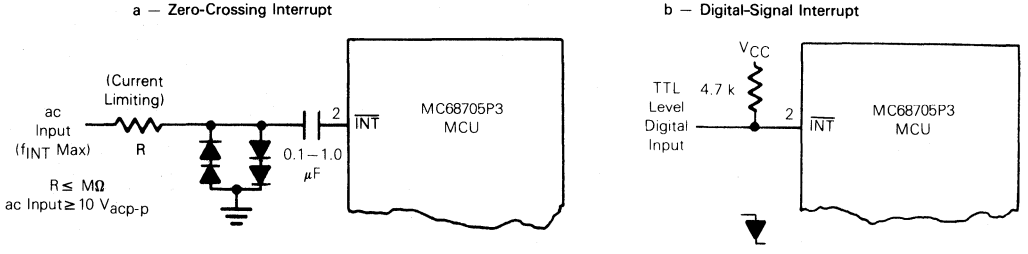
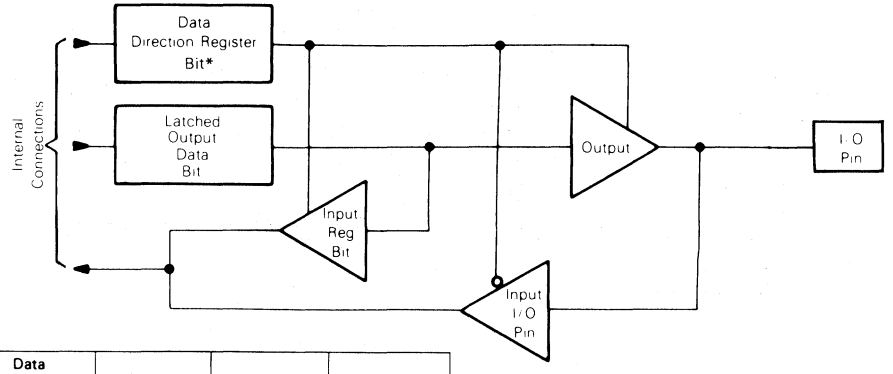


FIGURE 15 — TYPICAL PORT I/O CIRCUITRY



Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

*DDR is write-only register and reads as all "1s".

**Ports B and C are three-state ports. Port A has internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs while port B and C lines are CMOS compatible as inputs. The memory map in Figure 6 gives the address of data registers and DDRs. The Register configuration is provided in Figure 16. Figure 17 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

The latched output data bit (see Figure 15) may always be written. Therefore, any write to a port writes 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").

FIGURE 16 — MCU REGISTER CONFIGURATION

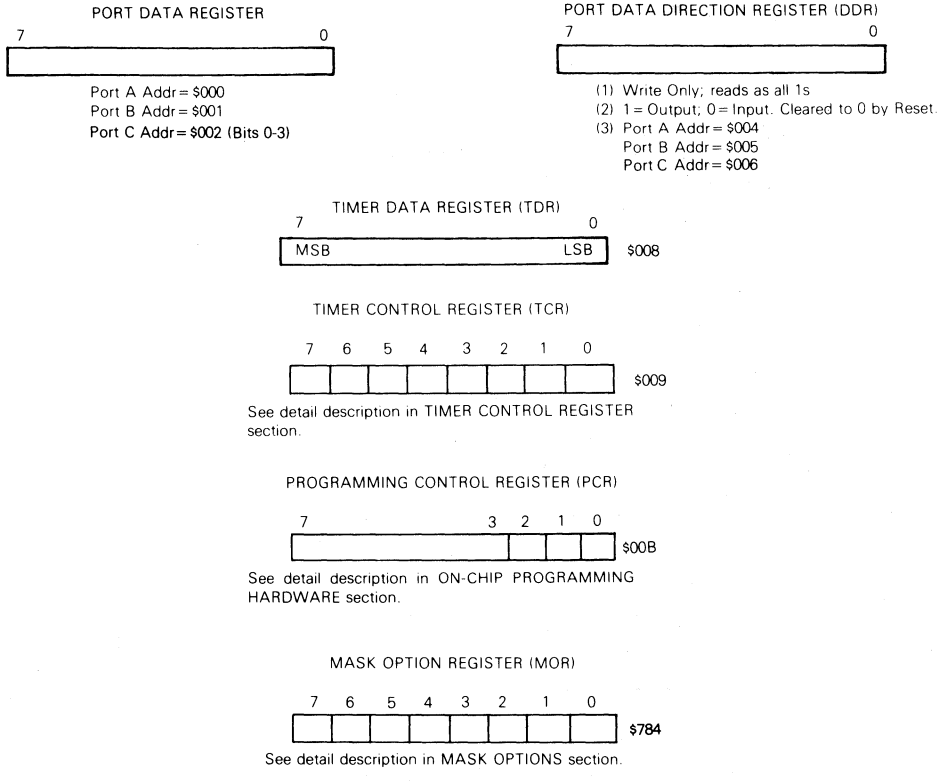
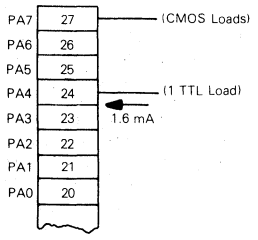
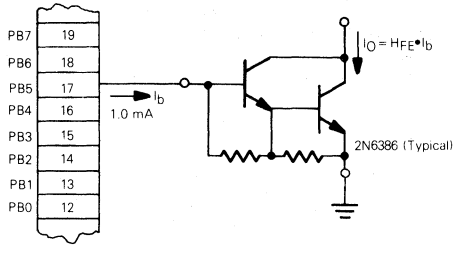


FIGURE 17 — TYPICAL PORT CONNECTIONS

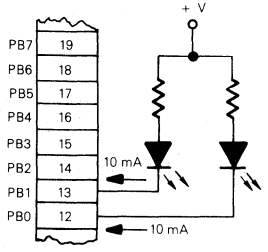
(a) Output Modes



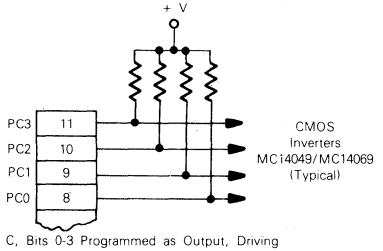
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

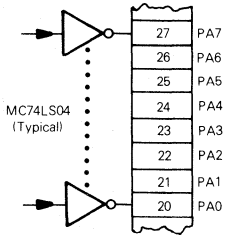


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

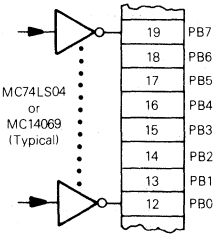


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors.

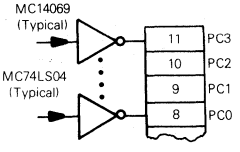
(b) Input Modes



TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly.



CMOS and TTL Driving Port C Directly.

TIMER CONTROL REGISTER (TCR)

The configuration of the TCR is determined by the logic level of bit 6 (Timer Option, TOPT) in the Mask Option Register (MOR). Two configurations of the TCR are shown below, one for TOPT = 1 and the other for TOPT = 0. TOPT = 1 configures the TCR to emulate the MC6805P2 or MC6805P4. When TPOT = 0, it provides software control of the TCR. When TOPT = 1, the prescaler "mask" options are user programmable via the MOR. A description of each TCR bit is provided below (also see Figure 8 and TIMER).

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	1	1	1	1	1	1	Timer Control Register $009

TCR with MOR TOPT = 1 (MC6805P2/P4 Emulation)

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	TIN	TIE	PSC*	PS2	PS1	PS0	Timer Control Register $009

TCR with MOR TOPT = 0 (Software Programmable Timer)

* = write only, reads as a zero

b7, TIR — Timer Interrupt Request — Used to initiate the timer interrupt or signal a timer Data Register underflow when it is a logical "1".

1 = Set when the Timer Data Register changes to all zeros.
0 = Cleared by external reset or under program control.

b6, TIM — Timer Interrupt Mask — Used to inhibit the timer interrupt, to the processor, when it is a logical "1".

1 = Set by an external reset or under program control.
0 = Cleared under program control.

b5, TIN — External or Internal — Selects the input clock source to be either the external TIMER pin (7) or the internal $\phi 2$.

1 = Selects the external clock source
0 = Selects the internal $\phi 2$ ($f_{OSC} \div 4$) clock source

b4, TIE — External Enable — Used to enable the external TIMER pin (7) or to enable the internal clock (if TIN = 0) regardless of the external timer pin state (disables gated clock feature). When TOPT = 1, TIE is always a logical "1".

1 = Enables external timer pin.
0 = Disables external timer pin.

TIN-TIE Modes

TIN	TIE	CLOCK
0	0	Internal Clock ($\phi 2$)
0	1	Gated (AND) of External and Internal Clocks
1	0	No Clock
1	1	External Clock

b3, PSC — Prescaler Clear — This is a write-only bit. It reads as a logical "0" (when TOPT = 0) so the BSET and BCLR on the TCR function correctly. Writing a "1" into PSC generates a pulse which clears the prescaler. (When TOPT = 1 this bit is always read as a logical "1" and has no effect on the prescaler.)

b2, PS2
b1, PS1
b0, PS0 — Prescaler Select — These bits are decoded to select one of eight outputs on the timer prescaler. The table below shows the prescaler division resulting from decoding these bits.

PS2	PS1	PS0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Note

When changing the PS2-PS0 bits in software, the PSC bit should be written to a "1" in the same write cycle to clear the prescaler. Changing the PS bits without clearing the prescaler may cause an extraneous toggle of the Timer Data Register.

MASK OPTIONS

The MC68705P3 Mask Option Register is implemented in EPROM. Like all other EPROM bytes, the MOR contains all zeros prior to programming.

When used to emulate the MC6805P2 or MC6805P4, five of the eight MOR bits are used in conjunction with the prescaler. Of the remaining, the b7 bit is used to select the type of oscillator clock, and bits b3 and b4 are not used. Bits b0, b1, and b2 determine the division of the Timer prescaler. Bit b5 determines the Timer clock source. The value of the TOPT bit (b6) is programmed to configure the TCR (a logic "1" for MC6805P2/P4 emulation).

If the MOR Timer Option (TOPT) bit is a 0, bits b5, b4, b2, b1, and b0 set the initial value of their respective TCR bits during reset. After initialization the TCR is software controllable.

A description of the MOR bits is as follows:

b7	b6	b5	b4	b3	b2	b1	b0	
CLK	TOPT	CLS			P2	P1	P0	Mask Option Register $784

b7, CLK Clock Oscillator Type
1 = RC
0 = Crystal

Note

V_{IHTP} on the TIMER pin (7) forces the crystal mode.

b6, TOPT Timer Option
1 = MC6805P2/P4 type timer/prescaler. All bits, except 6 and 7, of the Timer-Control Register (TCR) are invisible to the user. Bits 5, 2, 1, and 0 of the Mask Option Register determine the equivalent MC6805P2/P4 mask options.
0 = All TCR bits are implemented as a Software Programmable Timer. The state of MOR bits 5, 4, 2, 1, and 0 sets the initial values of their respective TCR bits (TCR is then software contolled after initialization).

b5, CLS Timer/Prescaler Clock Source
1 = External TIMER pin.
0 = Internal $\phi 2$

b4 Not used if MOR TOPT = 1 (MC6805P2/P4 emulation). Sets initial value of TCR TIE if MOR TOPT = 0.

b3 Not used.

b2, P2
b1, P1
b0, P0 Prescaler Option — the logical levels of these bits, when decoded, select one of eight outputs on the timer prescaler. The table below shows the division resulting from decoding combinations of these three bits.

P2	P1	P0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Two examples for programming the MOR are discussed below.

Example 1 To emulate an MC6805P2 to verfiy your program with an RC oscillator, and an event count input for the timer with no prescaling, the MOR would be set to "11111000". To write the MOR, it is simply programmed as any other EPROM byte.

Example 2 Suppose you wish to use the MC68705P3 programmable prescaler functions, and you wish the initial condition of the prescaler to be divided by 64, with the input disabled and an internal clock source. If the clock oscillator was to be in the crystal mode, the MOR would be set to "00001110".

ON-CHIP PROGRAMMING HARDWARE

The Programming Control Register (PCR) at location $00B is an 8-bit register which utilizes the three LSBs (the five MSBs are set to logic "1s"). This register provides the necessary control bits to allow programming the MC68705P3 EPROM. The bootstrap program manipulates the PCR when programming, so that users need not be concerned with the PCR in most applications. A description of each bit follows.

b7	b6	b5	b4	b3	b2	b1	b0	
1	1	1	1	1	$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Program Control Register $00B

b0, $\overline{PLE}$ Programming Latch Enable — When cleared this bit allows the address and data to be latched into the EPROM. When this bit is set, data can be read from the EPROM.
1 = (set) read EPROM
0 = (clear) latch address and data into EPROM (read disabled)
$\overline{PLE}$ is set during a Reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{VPON}$ is a logic "1".

b1, $\overline{PGE}$ Program Enable — When cleared, $\overline{PGE}$ enables programming of the EPROM. $\overline{PGE}$ can only be cleared if $\overline{PLE}$ is cleared. $\overline{PGE}$ must be set when changing the address and data; i.e., setting up the byte to be programmed.
1 = (set) inhibit EPROM programming
0 = (clear) enable EPROM programming (if $\overline{PLE}$ is low)
$\overline{PGE}$ is set during a Reset; however, it has no effect on EPROM circuits if $\overline{VPON}$ is a logic "1".

b2, $\overline{VPON}$ (V_{PP} ON) — $\overline{VPON}$ is a read-only bit and when at a logic "0" it indicates that a "high voltage" is present at the V_{PP} pin.
1 = no "high voltage" on V_{PP} pin
0 = "high voltage" on V_{PP} pin
VPON being "1" "disconnects" PGE and PLE from the rest of the chip, preventing accidental clearing of these bits from effecting the normal operating mode.

Note

$\overline{VPON}$ being "0" does not indicate that the V_{PP} level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

The Programming Control Register functions are shown below:

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	0	1	Invalid state; $\overline{PGE}=0$ iff $\overline{PLE}=0$
1	0	1	Invalid state; $\overline{PGE}=0$ iff $\overline{PLE}=0$
0	1	1	"High voltage" on V_{PP}
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled from system (Operating Mode)

ERASING THE EPROM

The MC68705P3 EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 Å. The recommended integrated dose (UV intensity x exposure time) is 25 Ws/cm^2. The lamps should be used without shortwave filters and the MC68705P3 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC68705P3 EPROM to the "0" state. Data is then entered by programming "1s" into the desired bit locations.

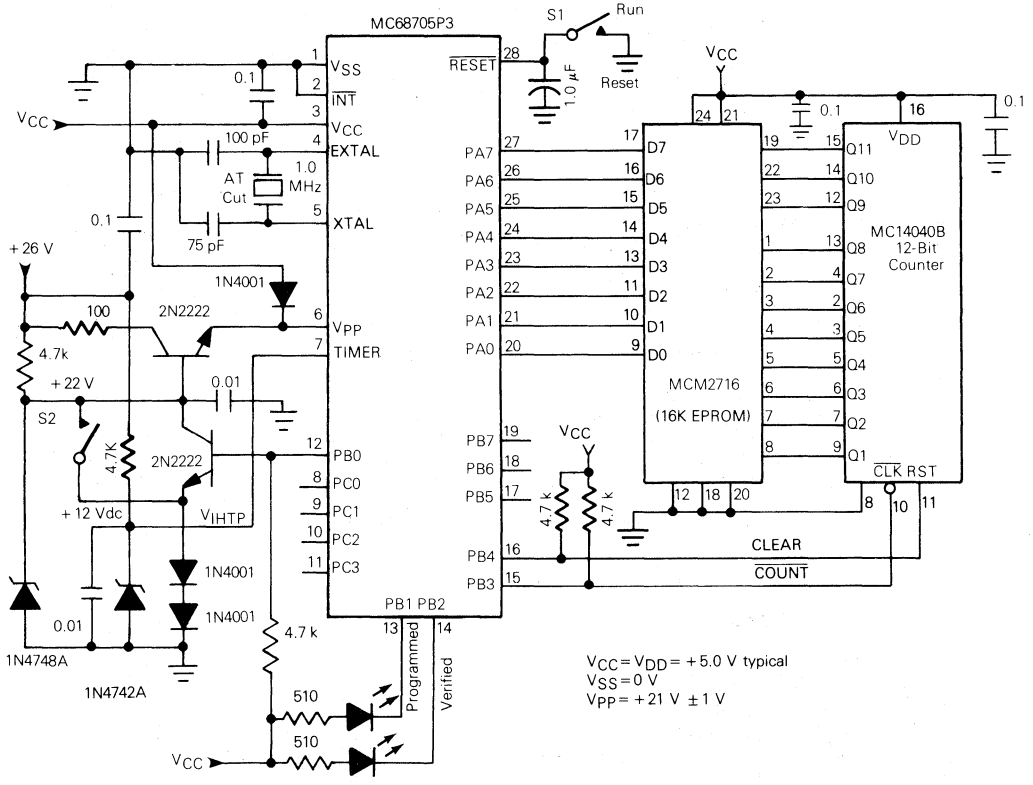
Caution

Be sure that the EPROM window is shielded from light except when erasing. This protects both the EPROM and light-sensitive nodes.

PROGRAMMING FIRMWARE

The MC68705P3 has 115 bytes of mask ROM containing a bootstrap program which can be used to program the MC68705P3 EPROM. The vector at addresses $7F6 and $7F7 is used to start executing the program. This vector is fetched when V_{IHTP} is applied to pin 7 (TIMER pin) of the MC68705P3 and the $\overline{RESET}$ pin is allowed to rise above V_{IRES+}. Figure 18 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM in the MC68705P3.

FIGURE 18 — PROGRAMMING CONNECTIONS SCHEMATIC DIAGRAM



Summary of Programming Steps:

1. When plugging in the MC68705P3 or the MCM2716, be sure that S1 and S2 are closed and that V_{CC} and +26 V are not applied.
2. To initiate programming, be sure S1 is closed; S2 is closed; and V_{CC} and +26 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705P3, first close S2 and then close S1. Disconnect V_{CC} and +26 V; then remove the MC68705P3.

PROGRAMMING STEPS

The MCM2716 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705P3. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705P3 and the MCM2716 are to be inserted and removed from the circuit they should be mounted in sockets. In addition, the precaution below must be observed (refer to Figure 18):

Caution

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705P3 and MCM2716 into their respective sockets. This ensures that $\overline{RESET}$ is held low while inserting the devices.

When ready to program the MC68705P3 it is only necessary to provide V_{CC} and +26 V, open switch S2 (to apply V_{PP} and V_{IHTP}), and then open S1 (to remove Reset). Once the voltages are applied and both S2 and S1 are open, the CLEAR output control line (PB4) goes high and then low, then the 11-bit counter (MC14040B) is clocked by the PB3 output ($\overline{COUNT}$). The counter selects the MCM2716 EPROM byte which is to load the equivalent MC68705P3 EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, $\overline{COUNT}$ clocks the counter to the next EPROM location. This continues until the MC68705P3 is completely programmed at which time the Programmed indicator LED is lit. The counter is cleared and the loop is repeated to verify the programmed data. The Verified indicator LED lights if the programming is correct.

Once the MC68705P3 has been programmed and verified, close switch S2 (to remove V_{PP} and V_{IHTP}) and close switch S1 (to Reset). Disconnect +26 V and V_{CC}; then remove the MC68705P3 from its socket.

MC6805P2 AND MC6805P4 EMULATION

The MC68705P3 emulates the MC6805P2 and MC6805P4 "exactly." MC6805P2/P4 mask features are implemented in the Mask Option Register (MOR) EPROM byte on the MC68705P3. There are a few minor exceptions to the exactness of emulation which are listed below:

1. The MC6805P2 "future ROM" area is implemented in the MC68705P3 and these 704 bytes must be left unprogrammed to accurately simulate the MC6805P2/P4. (The MC6805P2/P4 reads all "0s" from this area.)
2. The reserved ROM areas in the MC6805P2/P4 and MC68705P3 have different data stored in them and this data is subject to change without notice. The MC6805P2 uses the reserved ROM for the self-check feature and the MC68705P3 uses this area for the bootstrap program.
3. The MC6805P2 reads all "1s" in its 48 byte "future RAM" area. This RAM is not implemented in the MC6805P2 mask ROM version, but is implemented in the MC68705P3 and MC68705P4.
4. The V_{PP} line (pin 6) in the MC68705P3 must be tied to V_{CC} for normal operation. In the MC6805P2, pin 6 is the NUM pin and is grounded in normal operation. The MC6805P4 uses pin 6 for V_{SB} which is normally tied to V_{CC}, as with the MC68705P3.
5. The LVI feature is not available in the MC68705P3. Processing differences are not presently compatible with proper design of this feature in the EPROM version.
6. The function in the Non-User Mode is not identical to the MC6805P2/P4 version. Therefore, the MC68705P3 does not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805P2/P4 version, except for pin 6 as previously noted.
7. The MC6805P4 provides a standby RAM feature which is not available on the MC68705P3.

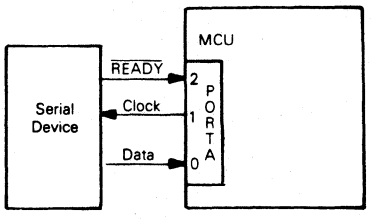
The operation of all other circuitry has been exactly duplicated or designed to function exactly the same in both devices including Interrupts, Timer, Data Ports, and Data Direction Registers (DDRs). A stated design goal has been to provide the user with a safe inexpensive way to verify his program and system design before committing to a factory programmed ROM.

SOFTWARE

BIT MANIPULATION

The MC68705P3 MCU has the ability to set or clear any single random-access memory or input/output bit (except the Data Direction Register, see Caution under INPUT/OUTPUT paragraph), with a single instruction (BSET, BCLR). Any bit in the page zero memory can be tested, using the BRSET and BRCLR instructions and the program branches as a result of its state. The Carry bit equals the value of the bit referenced by BRSET and BRCLR. A Rotate instruction may then be used to accumulate serial input data in a RAM location or register. This capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines. The coding example in Figure 19 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

FIGURE 19 — BIT MANIPULATION EXAMPLE




```
                .
                .
                .
SELF    BRSET   2,PORTA,SELF
                .
                .
                .
        BSET    1,PORTA
        BRCLR   0,PORTA,CONT
CONT    BCLR    1,PORTA
        ROR     RAMLOC
                .
                .
                .
```

has a data ready signal, a data output line, and a clock line to clock data one bit at a time, LSB 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 a RAM location.

ADDRESSING MODES

The MC68705P3 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE — In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM, I/O registers, and 128 bytes of EPROM. Direct addressing is an effective use of both memory and time.

EXTENDED — In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE — The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC, if and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 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.

INDEXED, NO OFFSET — In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET — In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET — In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with Direct and Extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the INPUT/OUTPUT paragraph.

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 and the condition (set or clear) is included in the opcode, and the address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC, if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the Condition Code Register. See Caution under the INPUT/OUTPUT paragraph.

INHERENT — In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC68705P3 MPU has a set of 59 basic instructions, which when combined with the 10 address modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS — Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 1.

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 (see Caution under INPUT/OUTPUT paragraph). The test for negative or zero (TST) instruction is included in the read-modify-write instructions, though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 3.

BIT MANIPULATION INSTRUCTIONS — These instructions are used on any bit in the first 256 bytes of the memory (see Caution under INPUT/OUTPUT paragraph). One group either sets or clears. The other group performs the bit and test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS — The control instructions control the MCU operations during program execution. Refer to Table 5.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 6.

OPCODE MAP SUMMARY — Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–		BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–		BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTION

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0....7)	—	—	—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0....7)	—	—	—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0....7)	10 + 2 • n	2	7	—	—	—
Clear Bit n	BCLR n (n = 0....7)	11 + 2 • n	2	7	—	—	—

TABLE 5 — CONTROL INSTRUCTIONS

Function	Mnemonic	Op Code	Inherent	
			# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Codes				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ

Condition Code Symbols

H	Half Carry (From Bit 3)
I	Interrupt Mask
N	Negative (Sign Bit)
Z	Zero
C	Carry/Borrow
Λ	Test and Set if True, Cleared Otherwise
•	Not Affected
?	Load CC Register From Stack
1	Set
0	Clear

TABLE 6 — INSTRUCTION SET (CONTINUED)

	Addressing Modes										Condition Codes				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEQ	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•

Condition Code Symbols

H	Half Carry (From Bit 3)
I	Interrupt Mask
N	Negative (Sign Bit)
Z	Zero
C	Carry/Borrow
Λ	Test and Set if True, Cleared Otherwise
•	Not Affected
?	Load CC Register From Stack
1	Set
0	Clear

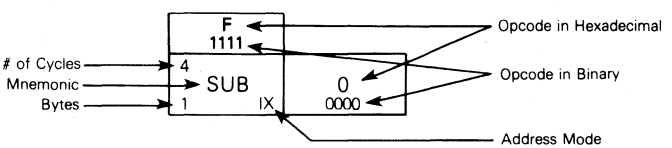
TABLE 7 — M6805 HMOS FAMILY INSTRUCTION SET OPCODE MAP

Hi / Low	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						Hi / Low
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



MOTOROLA

MC68705P5

Advance Information

8-BIT EPROM MICROCOMPUTER UNIT

The MC68705P5 Microcomputer Unit (MCU) is an EPROM member of the M6805 Family of low-cost single-chip microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turn-around time for prototype evaluation of the mask ROM versions. This 8-bit microcomputer contains a CPU, on-chip CLOCK, EPROM, bootstrap ROM, RAM, I/O, and a TIMER.

Because of these features, the MC68705P5 offers the user an economical means of designing an M6805 Family MCU into his system, either as a prototype evaluation, as a low-volume production run, or a pilot production run.

HARDWARE FEATURES

- 8-Bit Architecture
- 112 bytes of RAM
- Memory Mapped I/O
- 1804 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
  - External Timer Interrupt
- Vectored Interrupts — External, Timer, and Software
- Zero-Cross Detection on $\overline{INT}$ Input
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Master and Power-On Reset
- Complete Development System Support on EXORciser
- Emulates the MC6805P2 and MC6805P4 (Except for V_{SB})
- Bootstrap Program in ROM Simplifies EPROM Programming
- EPROM Security Features (Hardware and Software)

SOFTWARE FEATURES

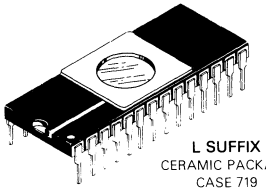
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to EPROM, RAM, and I/O

This document contains information on a new product. Specifications and information herein are subject to change without notice.

HMOS

(HIGH-DENSITY, N-CHANNEL DEPLETION LOAD, 5 V EPROM PROCESS)

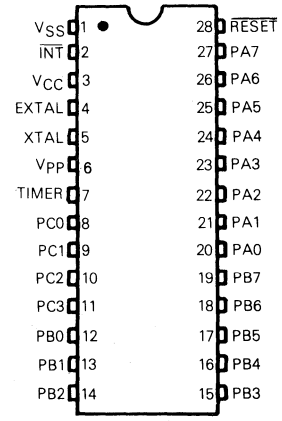
8-BIT EPROM MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719

S SUFFIX
CERDIP PACKAGE
ALSO AVAILABLE

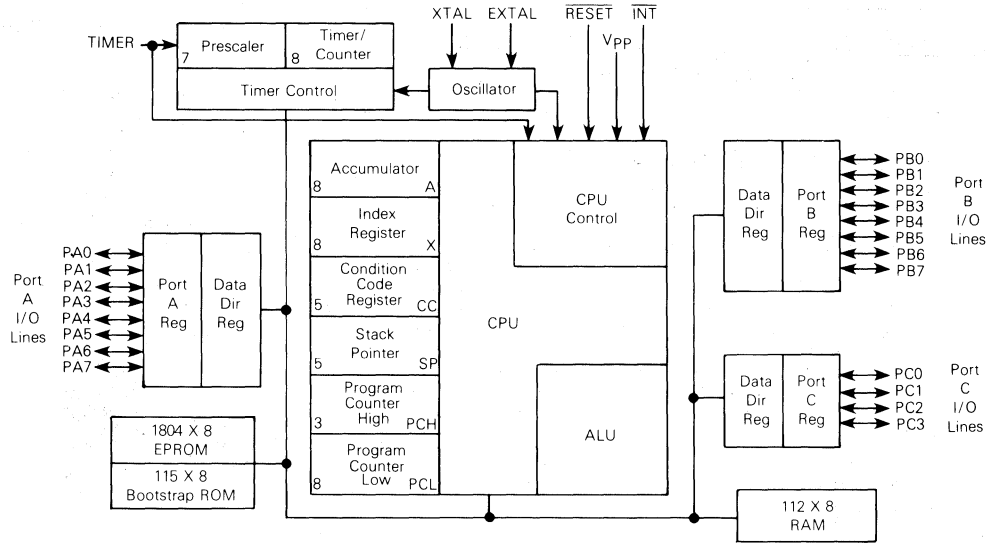
PIN ASSIGNMENT



GENERIC INFORMATION
($f = 1.0$ MHz, $T_A = 0$ to 70°C)

Package Type	Generic Number
Ceramic L Suffix	MC68705P5L
Cerdip S Suffix	MC68705P5S

MC68705P5 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage EPROM Programming Voltage (V_{PP} Pin) TIMER Pin	V_{PP}	−0.3 to +22.0	V
Normal Mode	V_{in}	−0.3 to +7.0	V
Bootstrap Programming Mode	V_{in}	−0.3 to +15.0	V
All Others	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature	T_J	+150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Package	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.25 Vdc ±0.5, V_{SS} = 0 Vdc, T_A = 20° to 30°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage (V_{PP} Pin)	V_{PP}	20.0	21.0	22.0	V
V_{PP} Supply Current V_{PP} = 5.25 V V_{PP} = 21.0 V	I_{PP}	 – –	 – –	 8 30	mA
Programming Oscillator Frequency	f_{oscp}	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) I_{in} = 100 μA Max	V_{IHTP}	9.0	12.0	15.0	V

SWITCHING CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	f_{osc}	0.4	–	4.2	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	μs
$\overline{INT}$ or Timer Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{RESET}$ Delay Time (External Cap = 1.0 μF)	t_{RHL}	100	–	–	ms
$\overline{INT}$ Zero Crossing Detection Input Frequency	f_{INT}	0.03	–	1.0	kHz
External Clock Duty Cycle (EXTAL) (See Figure 12)	–	40	50	60	%

ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage $\overline{RESET}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) $\overline{INT}$ (4.75 ≤ V_{CC} ≤ 5.75) (V_{CC} < 4.75) All Other	V_{IH}	 4.0 V_{CC} – 0.5 4.0 V_{CC} – 0.5 2.0	 – – ** ** –	 V_{CC} V_{CC} V_{CC} V_{CC} V_{CC}	V
Input High Voltage (TIMER Pin) Timer Mode Bootstrap Programming Mode	V_{IH}	 2.0 9.0	 – 12.0	 V_{CC} 15.0	V
Input Low Voltage $\overline{RESET}$ $\overline{INT}$ All Other	V_{IL}	 –0.3 –0.3 –0.3	 – ** –	 0.8 1.5 0.8	V
Internal Power Dissipation (No Port Loading, V_{CC} = 5.25 V, T_A = 0°C)	P_{INT}	–	450	TBD	mW
Input Capacitance XTAL All Other	C_{in}	 – –	 25 10	 – –	pF
$\overline{INT}$ Zero-Crossing Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	V_{acp-p}
$\overline{RESET}$ Hysteresis Voltage (See Figure 11) Out of Reset Voltage Into Reset Voltage	 V_{IRES+} V_{IRES-}	 2.1 0.8	 – –	 4.0 2.0	V
Programming Voltage (V_{PP} Pin) Programming EPROM Operating Mode	V_{PP}*	 20.0 4.0	 21.0 V_{CC}	 22.0 5.75	V
Input Current TIMER (V_{in} = 0.4 V) $\overline{INT}$ (V_{in} = 0.4 V) EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option) (V_{in} = 0.4 V Crystal Option) $\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	I_{in}	 – – – – –4.0	 – 20 – – –	 20 50 10 –1600 –40	μA

*V_{PP} is Pin 6 on the MC68705P5 and is connected to V_{CC} in the Normal Operating Mode. In the MC6805P2, Pin 6 is NUM and is connected to V_{SS} in the Normal Operating Mode. The user must allow for this difference when emulating the MC6805P2 ROM-based MCU.

**Due to internal biasing, this input when not used) floats to approximately 2.0 V.

PORT ELECTRICAL CHARACTERISTICS (V_{CC} = +5.25 Vdc, ±0.5 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Port A					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Output High Voltage, I_{Load} = – 10 μA	V_{OH}	V_{CC} – 1.0	–	–	V
Input High Voltage, I_{Load} = – 300 μA (Max)	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage, I_{Load} = – 500 μA (Max)	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current (V_{in} = 2.0 V to V_{CC})	I_{IH}	–	–	– 300	μA
Hi-Z State Input Current (V_{in} = 0.4 V)	I_{IL}	–	–	– 500	μA
Port B					
Output Low Voltage, I_{Load} = 3.2 mA	V_{OL}	–	–	0.4	V
Output Low Voltage, I_{Load} = 10 mA (Sink)	V_{OL}	–	–	1.0	V
Output High Voltage, I_{Load} = – 200 μA	V_{OH}	2.4	–	–	V
Darlington Current Drive (Source), V_O = 1.5 V	I_{OH}	– 1.0	–	– 10	mA
Input High Voltage	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA
Port C					
Output Low Voltage, I_{Load} = 1.6 mA	V_{OL}	–	–	0.4	V
Output High Voltage, I_{Load} = – 100 μA	V_{OH}	2.4	–	–	V
Input High Voltage	V_{IH}	2.0	–	V_{CC} + 0.7	V
Input Low Voltage	V_{IL}	V_{SS}	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

FIGURE 1 — TTL EQUIVALENT TEST LOAD (PORT B)

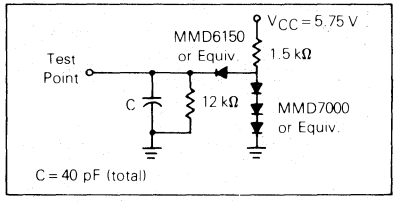


FIGURE 2 — CMOS EQUIVALENT TEST LOAD (PORT A)

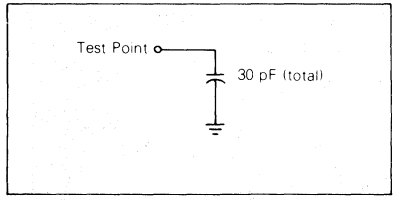
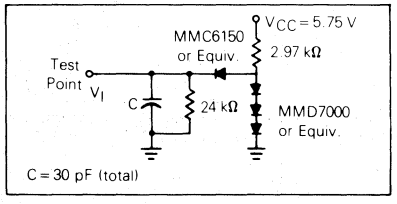


FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU are described in the following paragraphs.

V_{CC} and V_{SS} — Power is supplied to the MCU using two pins. V_{CC} is power and V_{SS} is the ground connection.

$\overline{INT}$ — This pin allows an external event to asynchronously interrupt the processor. It can also be used as a polled input using the BIL and BIH instructions. Refer to **INTERRUPTS** for additional information.

XTAL and EXTAL — These pins provide connections to the on-chip clock oscillator circuit. A crystal, a resistor, or an external signal, depending on the CLK bit (see **MASK OPTIONS**), is 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 **INTERNAL CLOCK GENERATOR OPTIONS** for recommendations about these inputs.

TIMER — This is used as an external input to control the internal timer/circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program (see **PROGRAMMING FIRMWARE**). Refer to **TIMER** for additional information about the timer circuitry.

$\overline{RESET}$ — This pin has a Schmitt Trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to **RESETS** for additional information.

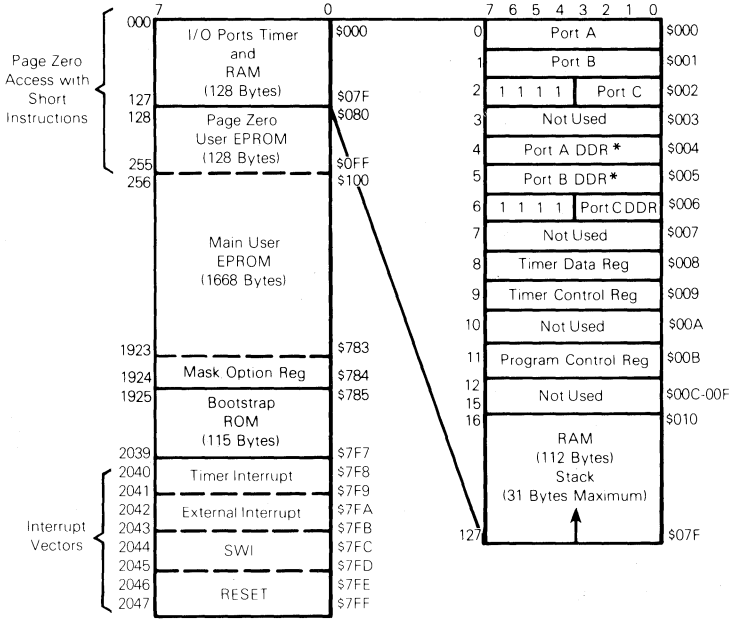
V_{PP} — This pin is used when programming the EPROM. By applying the programming voltage to this pin, one of the requirements is met for programming the EPROM. In normal operation, this pin is connected to V_{CC}. Refer to **PROGRAMMING FIRMWARE** and **ELECTRICAL CHARACTERISTICS.**

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) — 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 (DDRs). Refer to **INPUT/OUTPUT** for additional information, being sure to observe the Caution.

MEMORY

As shown in Figure 4, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC68705P5 MCU has implemented 2041 bytes of these locations. This consists of: 1804 bytes of user EPROM, 115 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM Mask Option Register (MOR), a Program Control Register (PCR), and eight bytes of I/O. The user EPROM is located in two areas. The main EPROM area is memory locations $080 to $783. The second area is reserved for eight interrupt/reset vector bytes at memory locations $7F8 to $7FF. The MCU uses nine of the lowest 16 memory locations for program control and I/O features such as ports, the port DDRs, and the timer. The Mask Option Register at memory location $784 completes the total. The 112 bytes of user RAM include up to 31 bytes for the stack.

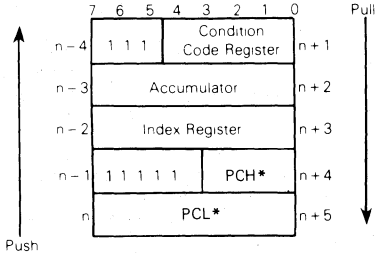
FIGURE 4 — MC68705P5 MCU MEMORY CONFIGURATION



Caution: Data Direction Registers (DDRs) are write-only; they read as $FF.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The register contents are pushed onto the stack in the order shown in Figure 5. Because the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly during pulls from the stack since the stack pointer increments during pulls. A subroutine call results in only the program counter (PCL, PCH) contents being pushed onto the stack; the remaining CPU registers are not pushed.

FIGURE 5 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCH and PCL are stacked.

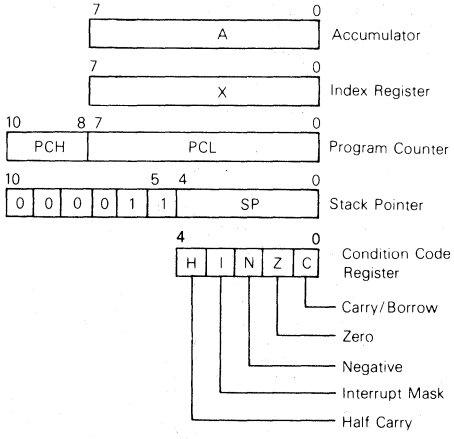
CENTRAL PROCESSING UNIT

The CPU of the M6805 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 address, data, and control buses.

REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 6 and are explained in the following paragraphs.

FIGURE 6 — PROGRAMMING MODEL



ACCUMULATOR (A) — The accumulator is a general purpose 8-bit register used to hold operands and results of the arithmetic calculations or data manipulations.

INDEX REGISTER (X) — The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an instruction value to create an effective address. The index register can also be used for data manipulations using read-modify-write instructions. The index register may also be used as a temporary storage area.

PROGRAM COUNTER (PC) — The program counter is an 11-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP) — The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the Reset Stack Pointer (RSP) instruction, the stack pointer is set to location $07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000011. Subroutines and interrupts may be nested down to location $061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).

CONDITION CODE REGISTER (CC) — The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each of the five bits is explained below.

Half Carry (H) — Set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I) — When this bit is set the timer and external interrupt ($\overline{INT}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

Negative (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical "1").

Zero (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C) — When set, this bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

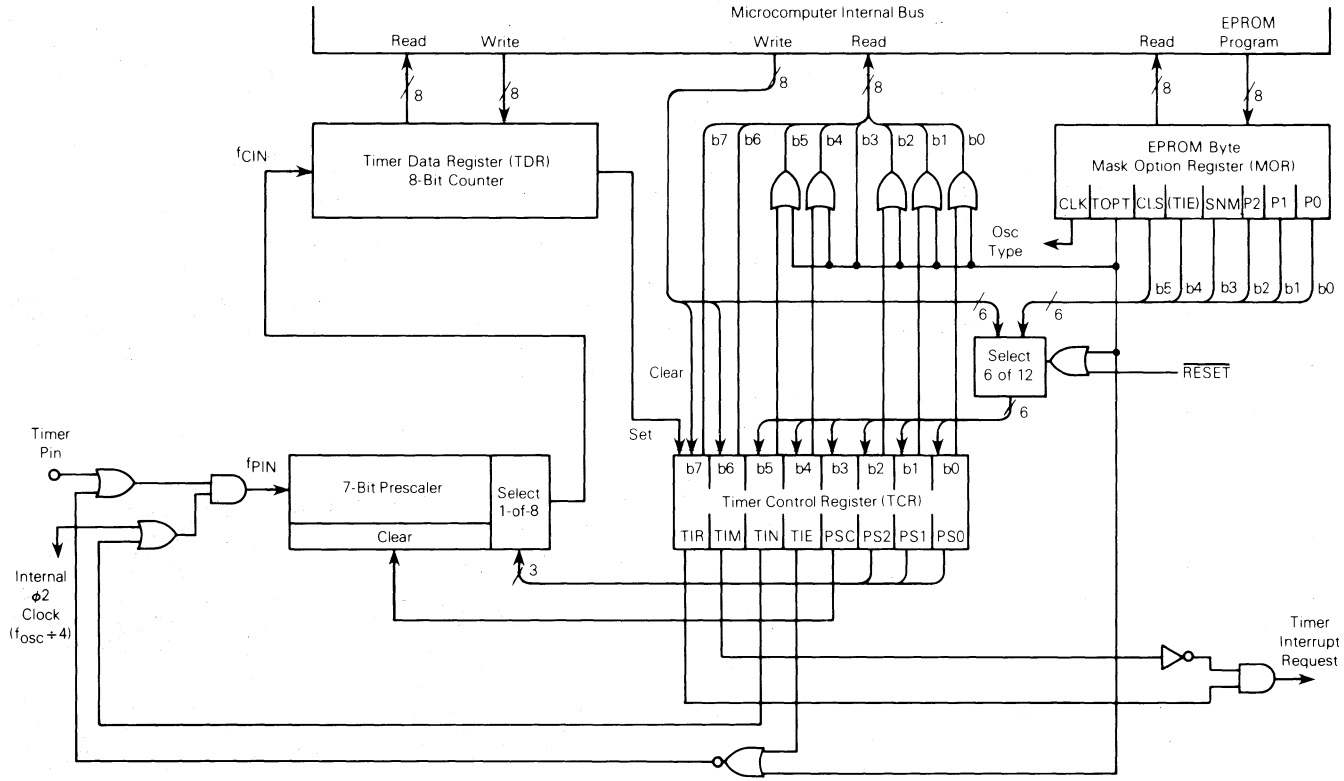
TIMER

The MC68705P5 MCU timer consists of an 8-bit software-programmable counter which is driven by a 7-bit software-programmable prescaler. Various timer clock sources may be selected ahead of the prescaler and counter. The timer selections are made via the Timer Control Register (TCR) and/or the Mask Option Register (MOR). The TCR also contains the interrupt control bits. The sections elsewhere entitled **TIMER CONTROL REGISTER** and **MASK OPTIONS** include additional details on controlling this timer.

The MCU timer circuitry is shown in Figure 7. The 8-bit counter may be loaded under program control and is decremented toward zero by the f_{CIN} counter input (output of the prescaler option selection). Once the 8-bit counter has decremented to zero, it sets the TIR (Timer Interrupt Request) bit 7 (b7 of TCR). The TIM (Timer Interrupt Mask) bit (b6) can be software set to inhibit the interrupt request, or software cleared to pass the interrupt request to the processor. When the I-bit in the Condition Code Register is cleared, the processor receives the Timer Interrupt. The MCU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9 and executing the interrupt routine. The processor is sensitive to the level of the timer interrupt request line; therefore if the interrupt is masked, the TIR bit may be cleared by software (e.g., BCLR) without generating an interrupt. When servicing a timer interrupt, the TIR bit MUST be cleared by the timer interrupt service routine software in order to clear the timer interrupt request.

The counter continues to count (decrement) after falling through to $FF from zero. Thus, the counter can be read at any time by the processor without disturbing the count. This allows a program to determine the length of time since the occurrence of a timer interrupt and does not disturb the counting process.

FIGURE 7 — MC68705P5 TIMER FUNCTIONAL BLOCK DIAGRAM



f_{PIN} — Prescaler Input Frequency
f_{CIN} — Counter Input Frequency

Timer Control Register Bits:
- TIR — Timer Interrupt Request Status
- TIM — Timer Interrupt Mask
- TIN — Timer Input Select
- TIE — Timer External Input Enable
- PSC — Prescaler Clear
- PS2, PS1, PS0 — Prescaler Select

Mask Option Register Bits:
- CLK — Clock Oscillator Type
- TOPT — Timer Mask/Programmable Option
- CLS — Timer Clock Source
- (TIE) — (Timer External Input Enable)
- SNM — Secure/Non-Secure Mode Option
- P2, P1, P0 — Prescaler Option

NOTES: The TOPT bit in the mask option register selects whether the timer is software programmable via the timer control register or emulates the mask programmable parts via the MOR PROM byte.

The TIE bit in the mask option register is not used if MOR TOPT = 1 (MC6805P2 emulation). It sets the intial value of TCR TIE if MOR TOPT = 0.

The clock input to the timer can be from an external source (decrementing the counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high state on the pin must also recognize the low state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250 \text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice).

When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER pin allowing the user to easily perform pulse-width measurements. (Note: When the MOR TOPT bit is set and the CLS bit is clear, an ungated $\phi 2$ clock input is obtained by tying the TIMER pin to V_{CC}.) The source of the clock input is selected via the TCR or the MOR as described later.

A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter. This prescaling option selects one of eight outputs on the 7-bit binary divider; one output bypasses prescaling. To avoid truncation errors, the prescaler is cleared when bit b3 of the TCR is written to a logic "1", when in the software controlled mode (TOPT = 0, more on these modes in later paragraphs); however, TCR bit b3 reads as a logic "0" when TOPT = 0 and as a "1" when TOPT = 1 to ensure proper operation with read-modify-write instructions (bit set and clear for example).

At Reset, the prescaler and counter are initialized to an all "1s" condition; the Timer Interrupt Request bit (TCR, b7) is cleared and the Timer Interrupt Request mask (TCR, b6) is set. TCR bits b0, b1, b2, b4, and b5 are initialized by the corresponding Mask Option Register (MOR) bits at Reset. They are then software selectable after Reset if TOPT = 0.

Note that the timer block diagram in Figure 7 reflects two separate timer control configurations: a) software controlled mode via the Timer Control Register (TCR), and b) MOR controlled mode to emulate a mask ROM version with the Mask Option Register. In the software controlled mode, all TCR bits are read/write, except bit b3 which is write-only (always reads as a logic "0"). In the MOR controlled mode, TCR bits b7 and b6 are read/write, the other six have no effect on a write and read as logic "1s". The two configurations provide the user with the capability to freely select timer options as well as accurately emulate the MC6805P2 and MC6805P4 mask ROM version. In the following paragraphs refer to Figure 9 as well as the TIMER CONTROL REGISTER and MASK OPTIONS sections.

The TOPT (Timer Option) bit (b6) in the Mask Option Register is EPROM programmed to a logical "0" to select the software controlled mode, which is described first. TCR bits b5, b4, b3, b2, b1, and b0 give the program direct control of the prescaler and input selection options.

The Timer Prescaler input (f_{PIN}) can be configured for three different operating modes, plus a disable mode, depending upon the value written to TCR control bits b4 and b5 (TIE and TIN).

When the TIE and TIN bits are programmed to "0", the timer input is from the internal clock ($\phi 2$) and TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

When TIE = 1 and TIN = 0, the internal clock and the TIMER input pin signals are ANDed to form the timer input f_{PIN}. This mode can be used to measure external pulse widths. The external pulse simply gates in the internal clock for the duration of the pulse. The accuracy of the count in this mode is ± one count.

When TIE = 0 and TIN = 1, no f_{PIN} input is applied to the prescaler and the timer is disabled.

When TIE and TIN are both programmed to a "1", the timer is from the external clock. The external clock can be used to count external events as well as provide an external frequency for generating periodic interrupts.

Bits b0, b1, and b2 in the TCR are program controlled to choose the appropriate prescaler output. The prescaling divides the f_{PIN} frequency by 1, 2, 4, etc. in binary multiples to 128 producing f_{CIN} frequency to the counter. The processor cannot write into or read from the prescaler; however, the prescaler is set to all "1s" by writing b3 of TCR to a "1", which allows for truncation-free counting.

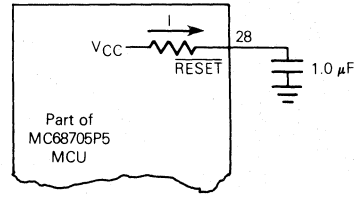
The MOR controlled mode of the timer is selected when the TOPT (Timer Option) bit (b6) in the MOR is programmed to a logical "1" to emulate the mask-programmable prescaler of the MC6805P2 and MC6805P4. The timer circuits are the same as described above; however, the Timer Control Register (TCR) is configured differently, as discussed below.

The logical level for the functions of bits b0, b1, b2, and b5 in the TCR are all determined at the time of EPROM programming. They are controlled by corresponding bits within the Mask Option Register (MOR, $784). The value programmed into MOR bits b0, b1, b2, and b5 controls the prescaler division and the timer clock selection. Bit b4 (TIE) and b3 (PSC) are set to a logical "1" in the MOR controlled mode. (When read by software, these six TCR bits always read as logical "1s".) As in the software programmable configuration, the TIM (b6) and TIR (b7) bits of the TCR are controlled by the counter and software as described above and in **TIMER CONTROL REGISTER.** The MOR controlled mode is designed to exactly emulate the MC6805P2 and MC6805P4 which has only TIM and TIR in the TCR and have the prescaler options defined as manufacturing mask options.

RESETS

The MCU can be reset in two ways: by initial power-up and by the external reset input ($\overline{\text{RESET}}$). Upon power-up, a delay of t_{RHL} is needed before allowing the $\overline{\text{RESET}}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{\text{RESET}}$ input, as shown in Figure 8, typically provides sufficient delay.

FIGURE 8 — POWER-UP RESET DELAY CIRCUIT



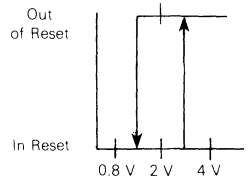
The internal circuit connected to the $\overline{\text{RESET}}$ pin consists of a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level. The Schmitt trigger provides an internal reset voltage when it senses logical "0" on the $\overline{\text{RESET}}$ pin. During power-up, the Schmitt trigger switches on (removes reset) when the $\overline{\text{RESET}}$ pin voltage rises to V_{IRES+}. When the $\overline{\text{RESET}}$ pin voltage falls to a logical "0" for a period longer than one t_{CYC}, the Schmitt trigger switches off to provide an internal reset voltage. The "switch off" voltage occurs at V_{IRES-}. A typical reset Schmitt trigger hysteresis curve is shown in Figure 9. See Figure 13 under **INTERRUPTS** for the complete reset sequence.

INTERNAL CLOCK GENERATOR OPTIONS

The internal clock generator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoffs. The Mask Option Register (EPROM) is programmed to select crystal or resistor operation. The oscillator frequency is internally divided by four to produce the internal system clocks.

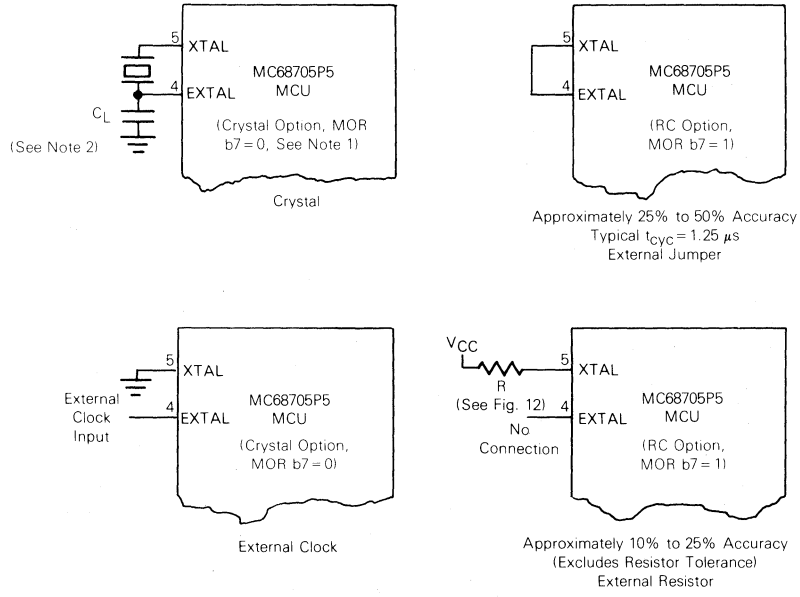
The different connection methods are shown in Figure 10.

FIGURE 9 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS



3

FIGURE 10 — CLOCK GENERATOR OPTIONS

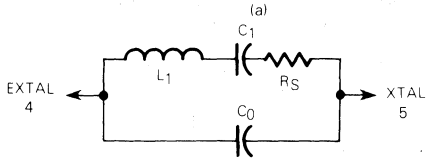


NOTES:

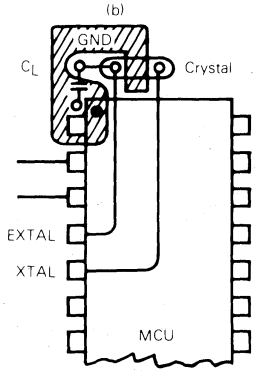
1. When the TIMER input pin is in the V_{IHTP} range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V_{CC}, the clock generator option is determined by bit 7 of the Mask Option Register (CLK).
2. The recommended C_L value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the Motional-Arm parameters of the crystal used.

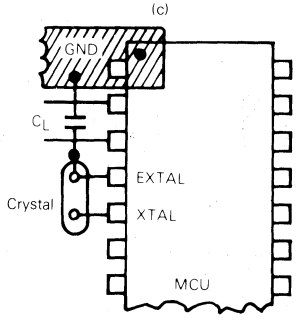
Crystal specifications and suggested PC board layouts are given in Figure 11. A resistor selection graph is given in Figure 12.

FIGURE 11 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



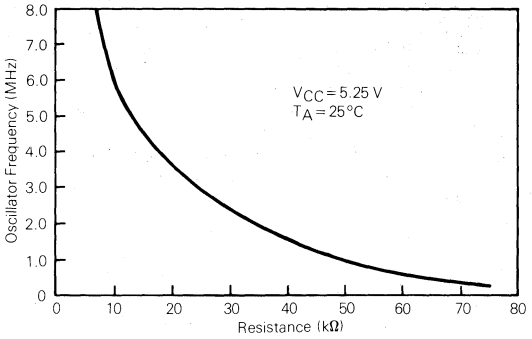
AT — Cut Parallel Resonance Crystal
C_O = 7 pF Max.
Freq. = 4.0 MHz @ C_L = 27 pF
R_S = 50 ohms Max.





NOTE: Keep crystal leads and circuit connections as short as possible.

FIGURE 12 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



The crystal oscillator start-up time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator start-up neither the crystal characteristics nor the load capacitances should exceed recommendations.

BOOTSTRAP ROM

The bootstrap ROM contains a factory program which allows the MCU to fetch data from an external device and transfer it into the MC68705P5 EPROM. The bootstrap program provides: timing of programming pulses, timing of V_{PP} input, and verification after programming. See **PROGRAMMING FIRMWARE.**

MASK OPTION REGISTER (MOR)

The Mask Option Register is an 8-bit user programmed (EPROM) register. Bits in this register are used to select the type of system clock, the timer option, the timer/prescaler clock source, the prescaler option, and the secure mode. It is fully described in the **MASK OPTIONS.**

INTERRUPTS

The MC68705P5 MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, or the software interrupt instruction (SWI). When any interrupt occurs: the current instruction (including SWI) is completed, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I-bit, and vector fetching requires a total of 11 t_{CYC} periods for completion. A flowchart of the interrupt sequence is shown in

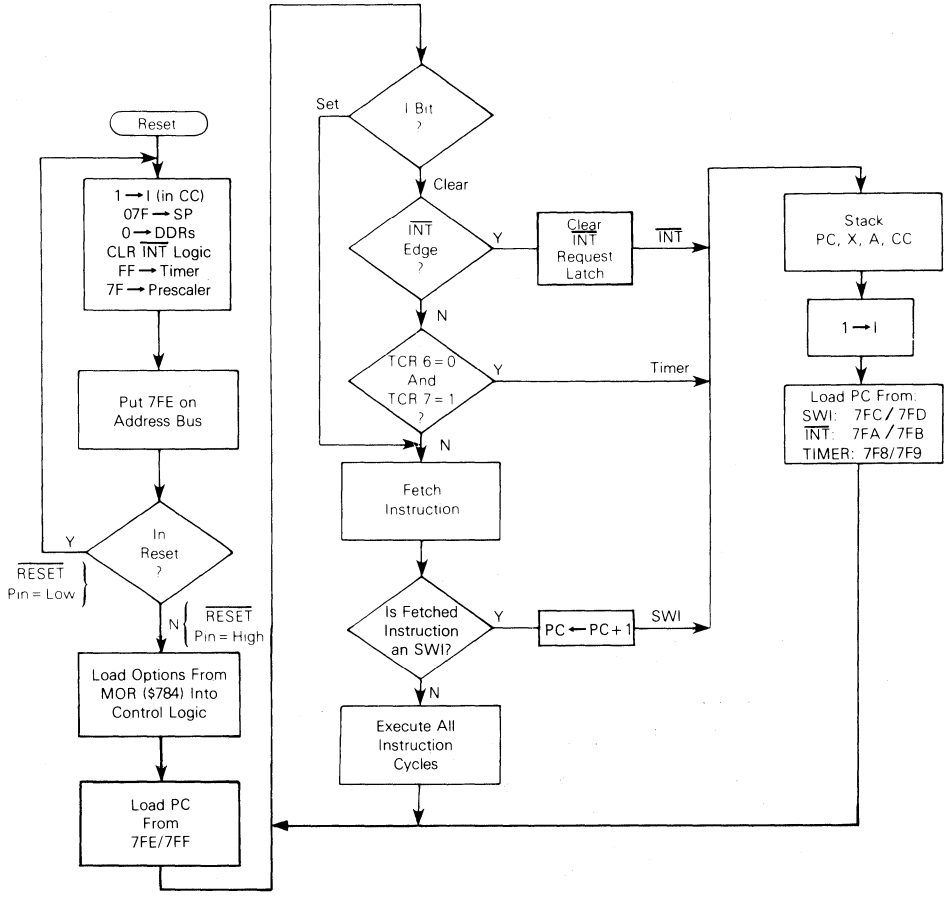
Figure 13. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction.

The external interrupt is internally synchronized and then latched on the falling edge of $\overline{\text{INT}}$. A sinusoidal input signal

FIGURE 13 — RESET AND INTERRUPT PROCESSING FLOWCHART



(f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 14(a), for use as a Zero-Crossing Detector (for negative transitions of AC sinusoid). This allows applications such as servicing time-of-day routines and engaging/disengaging AC power control devices. Off-chip full-wave rectification provides an interrupt at every zero crossing of the AC signal and thereby provides a 2f clock.

For digital applications, the $\overline{INT}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{INT}$ pin logic is dependent on the parameter labeled t_{WL}, t_{WH}. The pin logic that recognizes the high (or low) state on the pin must also recognize the low (or high) state on the pin in order to "re-arm" the internal logic. Therefore, the period can be calculated as follows: (assumes 50/50 duty cycle for a given period)

$$t_{cyc} \times 2 + 250\ ns = period = \frac{1}{freq}$$

The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily long period (250 ns twice). See Figure 14(b). For the $\overline{INT}$ function, the maximum allowable frequency is also determined by the software response of the $\overline{INT}$ service routine.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{INT}$ pin may be polled with branch instructions to provide an additional input pin. All pins on Ports 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 "1" for output or a logic "0" for input. On Reset all the DDRs are initialized to a logic "0" state, placing the ports in the input mode. The port output registers are not initialized on Reset but may be written to before setting 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 15. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

FIGURE 14 — TYPICAL INTERRUPT CIRCUITS

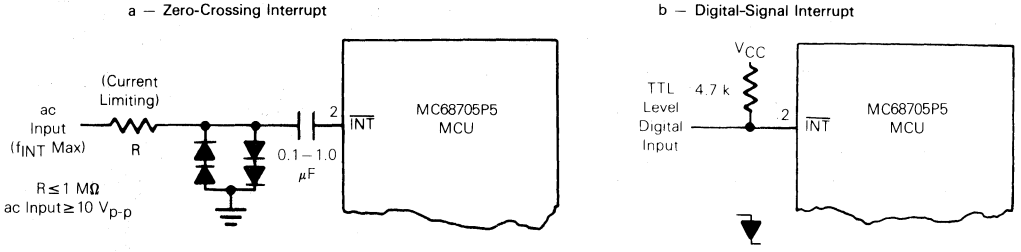
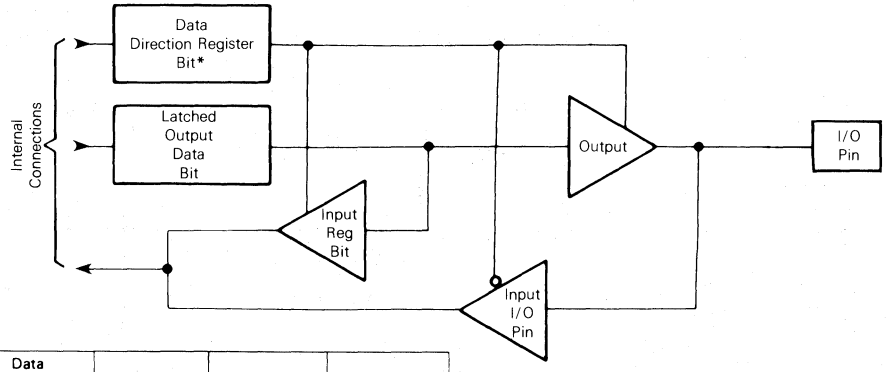


FIGURE 15 — TYPICAL PORT I/O CIRCUITRY



Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	High-Z**	Pin

*DDR is write-only register and reads as all "1s".

**Ports B and C are three-state ports. Port A has internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs while port B and C lines are CMOS compatible as inputs. The memory map in Figure 4 gives the address of data registers and DDRs. The Register configuration is provided in Figure 16. Figure 17 provides some examples of port connections.

Caution

The corresponding DDRs for ports A, B, and C are write-only registers (registers at $004, $005, and $006). A read operation on these registers is undefined. Since BSET and BCLR are read-modify-write functions they cannot be used to set or clear a single DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port must be written using a single-store instruction.

The latched output data bit (see Figure 15) may always be written. Therefore, any write to a port writes 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").

FIGURE 16 — MCU REGISTER CONFIGURATION

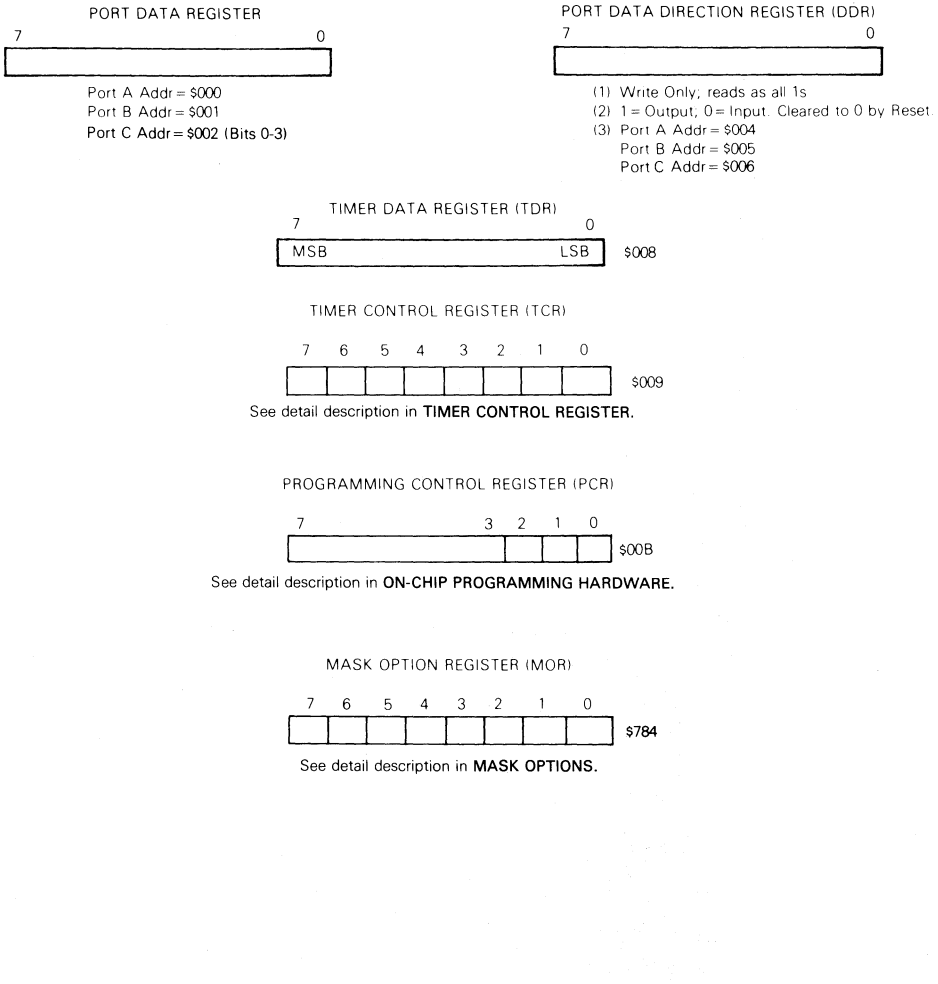
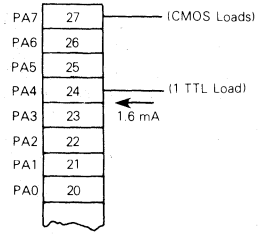
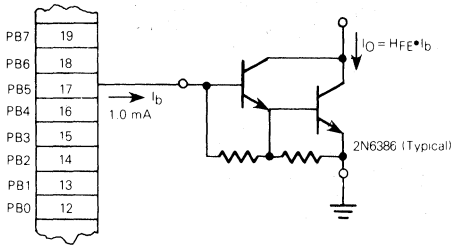


FIGURE 17 — TYPICAL PORT CONNECTIONS

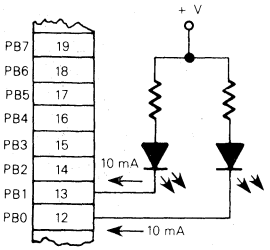
(a) Output Modes



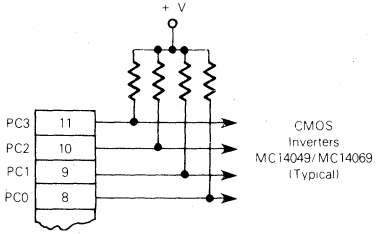
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

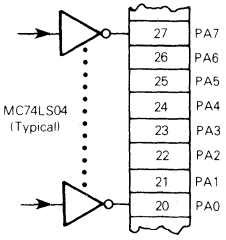


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

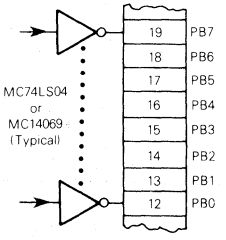


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors

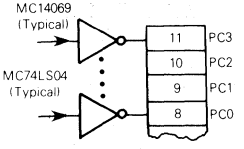
(b) Input Modes



TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly.



CMOS and TTL Driving Port C Directly.

TIMER CONTROL REGISTER (TCR)

The configuration of the TCR is determined by the logic level of bit 6 (Timer Option, TOPT) in the Mask Option Register (MOR). Two configurations of the TCR are shown below, one for TOPT = 1 and the other for TOPT = 0. TOPT = 1 configures the TCR to emulate the MC6805P2 or MC6805P4. When TPOT = 0, it provides software control of the TCR. When TOPT = 1, the prescaler "mask" options are user programmable via the MOR. A description of each TCR bit is provided below (also see Figure 7 and **TIMER**).

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	1	1	1	1	1	1	Timer Control Register $009

TCR with MOR TOPT = 1 (MC6805P2/P4 Emulation)

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	TIN	TIE	PSC*	PS2	PS1	PS0	Timer Control Register $009

TCR with MOR TOPT = 0 (Software Programmable Timer)

* = write only, reads as a zero

b7, TIR — Timer Interrupt Request — Used to initiate the timer interrupt or signal a timer Data Register underflow when it is a logical "1".

1 = Set when the Timer Data Register changes to all zeros.
0 = Cleared by external reset, power-on reset, or under program control.

b6, TIM — Timer Interrupt Mask — Used to inhibit the timer interrupt, to the processor, when it is a logical "1".

1 = Set by an external reset, power-on reset, or under program control.
0 = Cleared under program control.

b5, TIN — External or Internal — Selects the input clock source to be either the external TIMER pin (7) or the internal $\phi 2$.

1 = Selects the external clock source (event count mode)
0 = Selects the internal $\phi 2$ ($f_{OSC} \div 4$) clock source

b4, TIE — External Enable — Used to enable the external TIMER pin (7) or to enable the internal clock (if TIN = 0) regardless of the external timer pin state (disables gated clock feature). When TOPT = 1, TIE is always a logical "1".

1 = Enables external timer pin.
0 = Disables external timer pin.

TIN-TIE Modes

TIN	TIE	CLOCK
0	0	Internal Clock ($\phi 2$)
0	1	Gated (AND) of External and Internal Clocks
1	0	No Clock
1	1	External Clock

b3, PSC — Prescaler Clear — This is a write-only bit. It reads as a logical "0" (when TOPT = 0) so the BSET and BCLR on the TCR function correctly. Writing a "1" into PSC generates a pulse which clears the prescaler. (When TOPT = 1 this bit is always read as a logical "1" and has no effect on the prescaler.)

b2, PS2
b1, PS1
b0, PS0 — Prescaler Select — These bits are decoded to select one of eight outputs on the timer prescaler. The table below shows the prescaler division resulting from decoding these bits.

PS2	PS1	PS0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Note

When changing the PS2-PS0 bits in software, the PSC bit should be written to a "1" in the same write cycle to clear the prescaler. Changing the PS bits without clearing the prescaler may cause an extraneous toggle of the Timer Data Register.

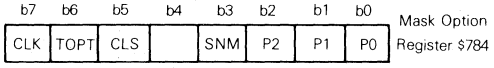
MASK OPTIONS

The MC68705P5 Mask Option Register is implemented in EPROM. Like all other EPROM bytes, the MOR contains all zeros prior to programming.

When used to emulate the MC6805P2 or MC6805P4, five of the eight MOR bits are used in conjunction with the prescaler. Of the remaining, the b7 bit is used to select the type of oscillator clock, b3 is the secure/non-secure mode option, and b4 is not used. Bits b0, b1, and b2 determine the division of the Timer prescaler. Bit b5 determines the Timer clock source. The value of the TOPT bit (b6) is programmed to configure the TCR (a logic "1" for MC6805P2/P4 emulation).

If the MOR Timer Option (TOPT) bit is a 0, bits b5, b4, b2, b1, and b0 set the initial value of their respective TCR bits during reset. After initialization the TCR is software controllable.

A description of the MOR bits is as follows:



b7, CLK Clock Oscillator Type
1 = RC
0 = Crystal

Note

V_{IHTP} on the TIMER pin (7) forces the crystal mode.

b6, TOPT Timer Option
1 = MC6805P2/P4 type timer/prescaler. All bits, except 6 and 7, of the Timer-Control Register (TCR) are invisible to the user. Bits 5, 2, 1, and 0 of the Mask Option Register determine the equivalent MC6805P2/P4 mask options.
0 = All TCR bits are implemented as a Software Programmable Timer. The state of MOR bits 5, 4, 2, 1, and 0 sets the initial values of their respective TCR bits (TCR is then software contolled after initialization).

b5, CLS Timer/Prescaler Clock Source
1 = External TIMER pin.
0 = Internal $\phi 2$

b4, (TIE) Not used if MOR TOPT = 1 (MC6805P2/P4 emulation). Sets initial value of TCR TIE if MOR TOPT = 0.

b3, SNM When this bit is set, i.e., programmed to a "1", it is not possible to access the EPROM contents of the MC68705P5 externally. For further information see **PROGRAMMING FIRMWARE.**

b2, P2
b1, P1
b0, P0 Prescaler Option—the logical levels of these bits, when decoded, select one of eight outputs on the timer prescaler. The table below shows the division resulting from decoding combinations of these three bits.

P2	P1	P0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

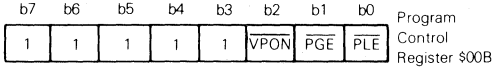
Two examples for programming the MOR are discussed below.

Example 1 To emulate an MC6805P2 to verify your program with an RC oscillator, and an event count input for the timer with no prescaling, the MOR would be set to "11111000". To write the MOR, it is simply programmed as any other EPROM byte.

Example 2 Suppose you wish to use the MC68705P5 programmable prescaler functions, and you wish the initial condition of the prescaler to be divided by 64, with the input disabled and an internal clock source. If the clock oscillator was to be in the crystal mode, the MOR would be set to "00001110".

ON-CHIP PROGRAMMING HARDWARE

The Programming Control Register (PCR) at location $00B is an 8-bit register which utilizes the three LSBs (the five MSBs are set to logic "1s"). This register provides the necessary control bits to allow programming the MC68705P5 EPROM. The bootstrap program manipulates the PCR when programming, so that users need not be concerned with the PCR in most applications. A description of each bit follows.



b0, $\overline{PLE}$ Programming Latch Enable—When cleared this bit allows the address and data to be latched into the EPROM. When this bit is set, data can be read from the EPROM.
1 = (set) read EPROM
0 = (clear) latch address and data into EPROM (read disabled)
$\overline{PLE}$ is set during a Reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{VPON}$ is a logic "1".

b1, $\overline{PGE}$ Program Enable—When cleared, $\overline{PGE}$ enables programming of the EPROM. $\overline{PGE}$ can only be cleared if $\overline{PLE}$ is cleared. $\overline{PGE}$ must be set when changing the address and data; i.e., setting up the byte to be programmed.
1 = (set) inhibit EPROM programming
0 = (clear) enable EPROM programming (if $\overline{PLE}$ is low)

$\overline{PGE}$ is set during a Reset; however, it has no effect on EPROM circuits if $\overline{VPON}$ is a logic "1".

b2, $\overline{VPON}$ (VPP ON) — $\overline{VPON}$ is a read-only bit and when at a logic "0" it indicates that a "high voltage" is present at the VPP pin.

1 = no "high voltage" on VPP pin
0 = "high voltage" on VPP pin

VPON being "1" "disconnects" PGE and PLE from the rest of the chip, preventing accidental clearing of these bits from effecting the normal operating mode.

Note

$\overline{VPON}$ being "0" does not indicate that the VPP level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

The Programming Control Register functions are shown below:

$\overline{VPON}$	$\overline{PGE}$	$\overline{PLE}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{PGE}$ and $\overline{PLE}$ disabled from system
0	0	1	Invalid state; $\overline{PGE} = 0$ iff $\overline{PLE} = 0$
1	0	1	Invalid state; $\overline{PGE} = 0$ iff $\overline{PLE} = 0$
0	1	1	"High voltage" on VPP
1	1	1	$\overline{PGE}$ and $\overline{PLE}$ disabled from system (Operating Mode)

ERASING THE EPROM

The MC68705P5 EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 Å. The recommended integrated dose (UV intensity x exposure time) is a 25 Ws/cm^2. The lamps should be used without shortwave filters and the MC68705P5 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC68705P5 EPROM to the "0" state. Data is then entered by programming "1s" into the desired bit locations.

Caution

Be sure that the EPROM window is shielded from light except when erasing. This protects both the EPROM and light-sensitive nodes.

PROGRAMMING FIRMWARE

The MC68705P5 has 115 bytes of mask ROM containing a bootstrap program which can be used to program the MC68705P5 EPROM. The vector at addresses $7F6 and $7F7 is used to start executing the program. This vector is fetched when V_{IHTP} is applied to pin 7 (TIMER pin) of the MC68705P5 and the $\overline{RESET}$ pin is allowed to rise above V_{IRES+}. Figure 18 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM in the MC68705P5.

Note that the MC68705P5 will not execute the bootstrap program when in the secure mode. Therefore, the on-chip EPROM must be completely erased before programming. To enter the secure mode, bit 3 of the mask option register must be programmed to a logic one and memory locations $782 and $783 must be programmed with $20 and $FE respectively. After programming, the only way to revert the non-secure mode is by erasing the entire EPROM.

PROGRAMMING STEPS

The MCM2716 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705P5. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705P5 and the MCM2716 are to be inserted and removed from the circuit they should be mounted in sockets. In addition, the precaution below must be observed (refer to Figure 18):

Caution

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705P5 and MCM2716 into their respective sockets. This ensures that $\overline{RESET}$ is held low while inserting the devices.

When ready to program the MC68705P5 it is only necessary to provide V_{CC} and +26 V, open switch S2 (to apply VPP and V_{IHTP}), and then open S1 (to remove Reset). Once the voltages are applied and both S2 and S1 are open, the CLEAR output control line (PB4) goes high and then low, then the 11-bit counter (MC14040B) is clocked by the PB3 output ($\overline{COUNT}$). The counter selects the MCM2716 EPROM byte which is to load the equivalent MC68705P5 EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, $\overline{COUNT}$ clocks the counter to the next EPROM location. This continues until the MC68705P5 is completely programmed at which time the Programmed indicator LED is lit. The counter is cleared and the loop is repeated to verify the programmed data. The Verified indicator LED lights if the programming is correct.

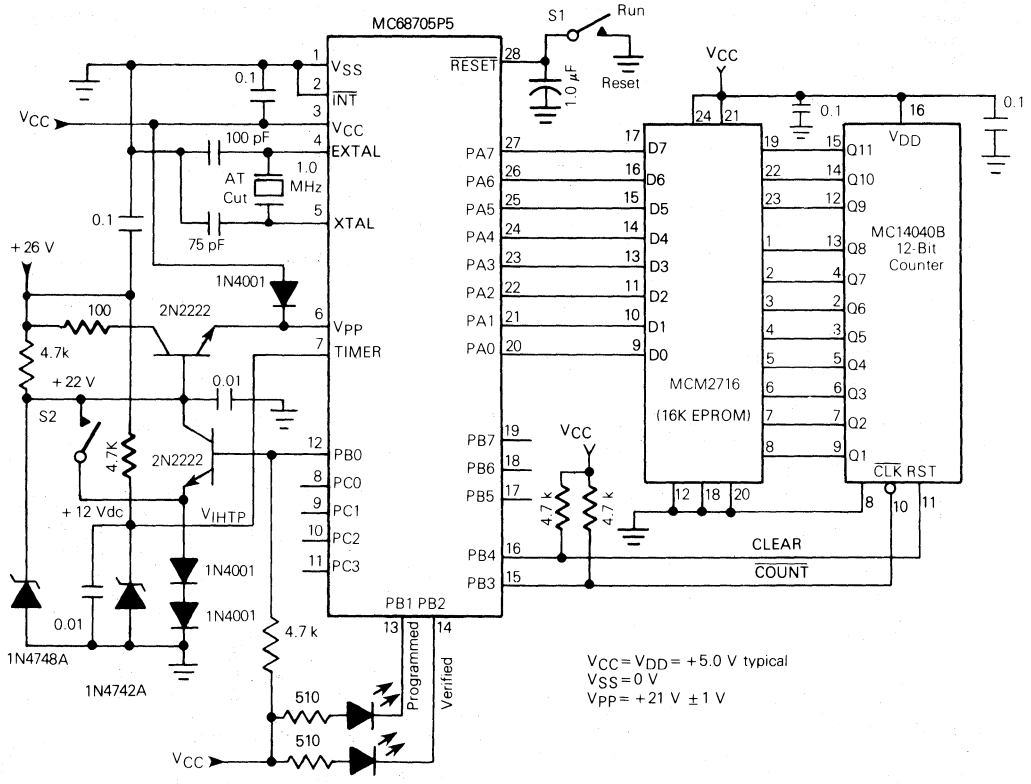
Once the MC68705P5 has been programmed and verified, close switch S2 (to remove VPP and V_{IHTP}) and close switch S1 (to Reset). Disconnect +26 V and V_{CC}; then remove the MC68705P5 from its socket.

MC6805P2 AND MC6805P4 EMULATION

The MC68705P5 emulates the MC6805P2 and MC6805P4 "exactly." MC6805P2/P4 mask features are implemented in the Mask Option Register (MOR) EPROM byte on the MC68705P5. There are a few minor exceptions to the exactness of emulation with are listed below:

1. The MC6805P2 "future ROM" area is implemented in the MC68705P5 and these 704 bytes must be left unprogrammed to accurately simulate the MC6805P2/P4. (The MC6805P2/P4 reads all "0s" from this area.)
2. The reserved ROM areas in the MC6805P2/P4 and MC68705P5 have different data stored in them and this data is subject to change without notice. The MC6805P2 uses the reserved ROM for the self-check feature and the MC68705P5 uses this area for the bootstrap program.
3. The MC6805P2 reads all "1s" in its 48 byte "future RAM" area. This RAM is not implemented in the MC6805P2 mask ROM version, but is implemented in the MC68705P5 and MC68705P3.
4. The VPP line (pin 6) in the MC68705P5 must be tied to V_{CC} for normal operation. In the MC6805P2, pin 6 is the NUM pin and is grounded in normal operation. The MC6805P4 uses pin 6 for V_{SB} which is normally tied to V_{CC}, as with the MC68705P5.
5. The function in the Non-User Mode is not identical to the MC6805P2/P4 version. Therefore, the MC68705P5

FIGURE 18 — PROGRAMMING CONNECTIONS SCHEMATIC DIAGRAM

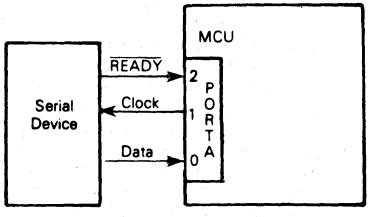


$V_{CC} = V_{DD} = +5.0$ V typical
$V_{SS} = 0$ V
$V_{PP} = +21$ V ±1 V

Summary of Programming Steps:

1. When plugging in the MC68705P5 or the MCM2716, be sure that S1 and S2 are closed and that V_{CC} and +26 V are not applied.
2. To initiate programming, be sure S1 is closed; S2 is closed; and V_{CC} and +26 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705P5, first close S2 and then close S1. Disconnect V_{CC} and +26 V; then remove the MC68705P5.

FIGURE 19 — BIT MANIPULATION EXAMPLE




```
                .
                .
                .
SELF    BRSET   2, PORTA, SELF
                .
                .
                .
        BSET    1, PORTA
        BRCLR   0, PORTA, CONT
CONT    BCLR    1, PORTA
        ROR     RAMLOC
                .
                .
                .
```

does not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805P2/P4 version, except for pin 6 as previously noted.

6. The MC6805P4 provides a standby RAM feature which is not available on the MC68705P5.

The operation of all other circuitry has been exactly duplicated or designed to function exactly the same in both devices including Interrupts, Timer, Data Ports, and Data Direction Registers (DDRs). A stated design goal has been to provide the user with a safe inexpensive way to verify his program and system design before committing to a factory programmed ROM.

SOFTWARE

BIT MANIPULATION

The MC68705P5 MCU has the ability to set or clear any single random-access memory or input/output bit (except the Data Direction Register, see Caution under **INPUT/OUTPUT**), with a single instruction (BSET, BCLR). Any bit in the page zero memory can be tested, using the BRSET and BRCLR instructions and the program branches as a result of its state. The Carry bit equals the value of the bit referenced by BRSET and BRCLR. A Rotate instruction may then be used to accumulate serial input data in a RAM location or register. This capability to work with any bit in RAM, ROM, or I/O allows the user to have individual flags in RAM or to handle I/O bits as control lines. The coding example in Figure 19 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, LSB 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 a RAM location.

ADDRESSING MODES

The MC68705P5 MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustrations, refer to the M6805 Family Users Manual.

The term "effective address" (EA) is used in describing the addressing modes. EA is defined as the address from which the argument for an instruction is fetched or stored.

IMMEDIATE – In the immediate addressing mode, the operand is contained in the byte immediately 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).

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 lowest 256 bytes in memory with a single 2-byte instruction. This address area includes all on-chip RAM, I/O registers, and 128 bytes of EPROM. Direct addressing is an effective use of both memory and time.

EXTENDED – In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions using extended addressing are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the programmer need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

RELATIVE – The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC, if and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 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.

INDEXED, NO OFFSET – In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

INDEXED, 8-BIT OFFSET – In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and unsigned byte following the opcode. This addressing mode is useful in selecting the kth element in an n element table. With this 2-byte instruction, k would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

INDEXED, 16-BIT OFFSET – In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset, except that this 3-byte instruction allows tables to be anywhere in memory. As with Direct and Extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

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 read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction. See Caution under the **INPUT/OUTPUT.**

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 and the condition (set or clear) is included in the opcode, and the

address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset is in the third byte and is added to the value of the PC, if the branch condition is true. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of 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 bit of the Condition Code Register. See Caution under **INPUT/OUTPUT.**

INHERENT – In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as control instruction with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC68705P5 MPU has a set of 59 basic instructions, which when combined with the 10 address modes produce 207 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.

REGISTER/MEMORY INSTRUCTIONS – Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 1.

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 (see Caution under **INPUT/OUTPUT**). The test for negative or zero (TST) instruction is included in the read-modify-write instructions, though it does not perform the write. Refer to Table 2.

BRANCH INSTRUCTIONS – The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 3.

BIT MANIPULATION INSTRUCTIONS – These instructions are used on any bit in the first 256 bytes of the memory (see Caution under **INPUT/OUTPUT**). One group either sets or clears. The other group performs the bit and test branch operations. Refer to Table 4.

CONTROL INSTRUCTIONS – The control instructions control the MCU operations during program execution. Refer to Table 5.

ALPHABETICAL LISTING – The complete instruction set is given in alphabetical order in Table 6.

OPCODE MAP SUMMARY – Table 7 is an opcode map for the instructions used on the MCU.

TABLE 1 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	OP Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	—	—	—	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	—	—	—	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–		BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–		BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

TABLE 2 — READ-MODIFY-WRITE INSTRUCTION

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8 Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

TABLE 3 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IFF Higher	BHI	22	2	4
Branch IFF Lower or Same	BLS	23	2	4
Branch IFF Carry Clear	BCC	24	2	4
(Branch IFF Higher or Same)	(BHS)	24	2	4
Branch IFF Carry Set	BCS	25	2	4
(Branch IFF Lower)	(BLO)	25	2	4
Branch IFF Not Equal	BNE	26	2	4
Branch IFF Equal	BEQ	27	2	4
Branch IFF Half Carry Clear	BHCC	28	2	4
Branch IFF Half Carry Set	BHCS	29	2	4
Branch IFF Plus	BPL	2A	2	4
Branch IFF Minus	BMI	2B	2	4
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IFF Interrupt Line is Low	BIL	2E	2	4
Branch IFF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

TABLE 4 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is set	BRSET n (n = 0....7)	—	—	—	2 • n	3	10
Branch IFF Bit n is clear	BRCLR n (n = 0....7)	—	—	—	01 + 2 • n	3	10
Set Bit n	BSET n (n = 0....7)	10 + 2 • n	2	7	—	—	—
Clear Bit n	BCLR n (n = 0....7)	11 + 2 • n	2	7	—	—	—

TABLE 5 — CONTROL INSTRUCTIONS

Function	Mnemonic	Op Code	Inherent	
			# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

TABLE 6 — INSTRUCTION SET

	Addressing Modes										Condition Codes				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ

Condition Code Symbols

H	Half Carry (From Bit 3)
I	Interrupt Mask
N	Negative (Sign Bit)
Z	Zero
C	Carry/Borrow
Λ	Test and Set if True, Cleared Otherwise
•	Not Affected
?	Load CC Register From Stack
1	Set
0	Clear

TABLE 6 — INSTRUCTION SET (CONTINUED)

	Addressing Modes										Condition Codes				
Mnemonic	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•

Condition Code Symbols

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
Λ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack
1 Set
0 Clear

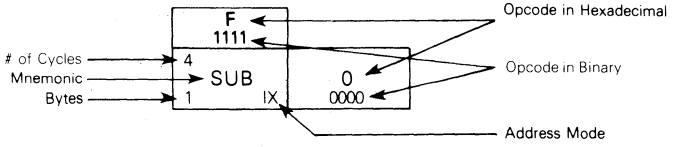
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

Hi / Low	Bit Manipulation BTB 0 0000	Bit Manipulation BSC 1 0001	Branch REL 2 0010	Read-Modify-Write DIR 3 0011	Read-Modify-Write INH 4 0100	Read-Modify-Write INH 5 0101	Read-Modify-Write IX1 6 0110	Read-Modify-Write IX 7 0111	Control INH 8 1000	Control INH 9 1001	Register/Memory IMM A 1010	Register/Memory DIR B 1011	Register/Memory EXT C 1100	Register/Memory IX2 D 1101	Register/Memory IX1 E 1110	Register/Memory IX F 1111	Hi / Low
0 0000	10 BRSET0 3 BTB	7 BSET0 2 BSC	4 BRA 2 REL	6 NEG 2 DIR	4 NEG 1 INH	4 NEG 1 INH	7 NEG 2 IX1	6 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	4 SUB 2 DIR	5 SUB 3 EXT	6 SUB 3 IX2	5 SUB 2 IX1	4 SUB 1 IX	0 0000
1 0001	10 BRCLR0 3 BTB	7 BCLR0 2 BSC	4 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	4 CMP 2 DIR	5 CMP 3 EXT	6 CMP 3 IX2	5 CMP 2 IX1	4 CMP 1 IX	1 0001
2 0010	10 BRSET1 3 BTB	7 BSET1 2 BSC	4 BHI 2 REL								2 SBC 2 IMM	4 SBC 2 DIR	5 SBC 3 EXT	6 SBC 3 IX2	5 SBC 2 IX1	4 SBC 1 IX	2 0010
3 0011	10 BRCLR1 3 BTB	7 BCLR1 2 BSC	4 BLS 2 REL	6 COM 2 DIR	4 COMA 1 INH	4 COMX 1 INH	7 COM 2 IX1	6 COM 1 IX	11 SWI 1 INH		2 CPX 2 IMM	4 CPX 2 DIR	5 CPX 3 EXT	6 CPX 3 IX2	5 CPX 2 IX1	4 CPX 1 IX	3 0011
4 0100	10 BRSET2 3 BTB	7 BSET2 2 BSC	4 BCC 2 REL	6 LSR 2 DTR	4 LSRA 1 INH	4 LSRX 1 INH	7 LSR 2 IX1	6 LSR 1 IX			2 AND 2 IMM	4 AND 2 DIR	5 AND 3 EXT	6 AND 3 IX2	5 AND 2 IX1	4 AND 1 IX	4 0100
5 0101	10 BRCLR2 3 BTB	7 BCLR2 2 BSC	4 BCS 2 REL								2 BIT 2 IMM	4 BIT 2 DIR	5 BIT 3 EXT	6 BIT 3 IX2	5 BIT 2 IX1	4 BIT 1 IX	5 0101
6 0110	10 BRSET3 3 BTB	7 BSET3 2 BSC	4 BNE 2 REL	6 ROR 2 DIR	4 RORA 1 INH	4 RORX 1 INH	7 ROR 2 IX1	6 ROR 1 IX			2 LDA 2 IMM	4 LDA 2 DIR	5 LDA 3 EXT	6 LDA 3 IX2	5 LDA 2 IX1	4 LDA 1 IX	6 0110
7 0111	10 BRCLR3 3 BTB	7 BCLR3 2 BSC	4 BEQ 2 REL	6 ASR 2 DIR	4 ASRA 1 INH	4 ASRX 1 INH	7 ASR 2 IX1	6 ASR 1 IX		2 TAX 1 INH		5 STA 2 DIR	6 STA 3 EXT	7 STA 3 IX2	6 STA 2 IX1	5 STA 1 IX	7 0111
8 1000	10 BRSET4 3 BTB	7 BSET4 2 BSC	4 BHCC 2 REL	6 LSL 2 DIR	4 LSLA 1 INH	4 LSLX 1 INH	7 LSL 2 IX1	6 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	4 EOR 2 DIR	5 EOR 3 EXT	6 EOR 3 IX2	5 EOR 2 IX1	4 EOR 1 IX	8 1000
9 1001	10 BRCLR4 3 BTB	7 BCLR4 2 BSC	4 BHCS 2 REL	6 ROL 2 DIR	4 ROLA 1 INH	4 ROLX 1 INH	7 ROL 2 IX1	6 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	4 ADC 2 DIR	5 ADC 3 EXT	6 ADC 3 IX2	5 ADC 2 IX1	4 ADC 1 IX	9 1001
A 1010	10 BRSET5 3 BTB	7 BSET5 2 BSC	4 BPL 2 REL	6 DEC 2 DIR	4 DECA 1 INH	4 DECX 1 INH	7 DEC 2 IX1	6 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	4 ORA 2 DIR	5 ORA 3 EXT	6 ORA 3 IX2	5 ORA 2 IX1	4 ORA 1 IX	A 1010
B 1011	10 BRCLR5 3 BTB	7 BCLR5 2 BSC	4 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	4 ADD 2 DIR	5 ADD 3 EXT	6 ADD 3 IX2	5 ADD 2 IX1	4 ADD 1 IX	B 1011
C 1100	10 BRSET6 3 BTB	7 BSET6 2 BSC	4 BMC 2 REL	6 INC 2 DIR	4 INCA 1 INH	4 INCX 1 INH	7 INC 2 IX1	6 INC 1 IX		2 RSP 1 INH		3 JMP 2 DIR	4 JMP 3 EXT	5 JMP 3 IX2	4 JMP 2 IX1	3 JMP 1 IX	C 1100
D 1101	10 BRCLR6 3 BTB	7 BCLR6 2 BSC	4 BMS 2 REL	6 TST 2 DIR	4 TSTA 1 INH	4 TSTX 1 INH	7 TST 2 IX1	6 TST 1 IX		2 NOP 1 INH	8 BSR 2 REL	7 JSR 2 DIR	8 JSR 3 EXT	9 JSR 3 IX2	8 JSR 2 IX1	7 JSR 1 IX	D 1101
E 1110	10 BRSET7 3 BTB	7 BSET7 2 BSC	4 BIL 2 REL								2 LDX 2 IMM	4 LDX 2 DIR	5 LDX 3 EXT	6 LDX 3 IX2	5 LDX 2 IX1	4 LDX 1 IX	E 1110
F 1111	10 BRCLR7 3 BTB	7 BCLR7 2 BSC	4 BIH 2 REL	6 CLR 2 DIR	4 CLRA 1 INH	4 CLRX 1 INH	7 CLR 2 IX1	6 CLR 1 IX		2 TXA 1 INH		5 STX 2 DIR	6 STX 3 EXT	7 STX 3 IX2	6 STX 2 IX1	5 STX 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND



MOTOROLA

MC146805E2

Advance Information

CMOS
(HIGH PERFORMANCE SILICON GATE)

8-BIT MICROPROCESSOR

8-BIT MICROPROCESSOR UNIT

The MC146805E2 Microprocessor Unit (MPU) belongs to the M6805 Family of Microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and TIMER. It is a low-power, low-cost processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the MC146805E2 MPU:

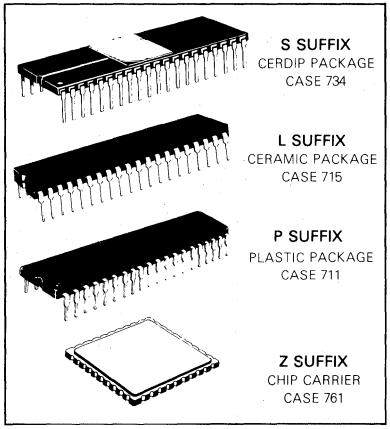
3

HARDWARE FEATURES

- Typical Full Speed Operating Power of 35 mW @ 5 V
- Typical WAIT Mode Power of 5 mW
- Typical STOP Mode Power of 25 μW
- 112 Bytes of On-Chip RAM
- 16 Bidirectional I/O Lines
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- Full External and Timer Interrupts
- Multiplexed Address/Data Bus
- Master Reset and Power-On Reset
- Capable of Addressing Up to 8K Bytes of External Memory
- Single 3- to 6-Volt Supply
- On-Chip Oscillator
- 40-Pin Dual-In-Line Package
- Chip Carrier Also Available

SOFTWARE FEATURES

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power Saving Standby Modes



S SUFFIX
CERDIP PACKAGE
CASE 734

L SUFFIX
CERAMIC PACKAGE
CASE 715

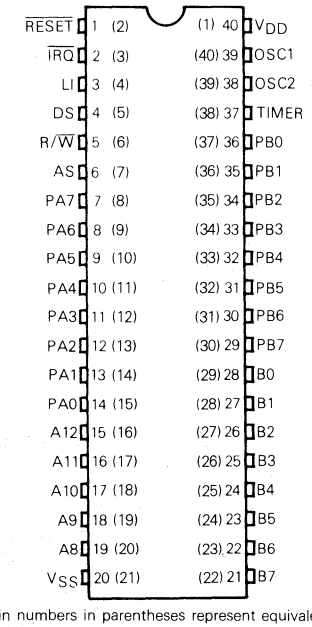
P SUFFIX
PLASTIC PACKAGE
CASE 711

Z SUFFIX
CHIP CARRIER
CASE 761

GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic	1.0	0°C to 70°C	MC146805E2L
L Suffix	1.0	−40°C to 85°C	MC146805E2CL
Cerdip	1.0	0°C to 70°C	MC146805E2S
S Suffix	1.0	−40°C to 85°C	MC146805E2CS
Plastic	1.0	0°C to 70°C	MC146805E2P
P Suffix	1.0	−40°C to 85°C	MC146805E2CP
Leadless Chip Carrier	1.0	0°C to 70°C	MC146805E2Z
Z Suffix	1.0	−40°C to 85°C	MC146805E2CZ

PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS (voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805E2 MC146805E2C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

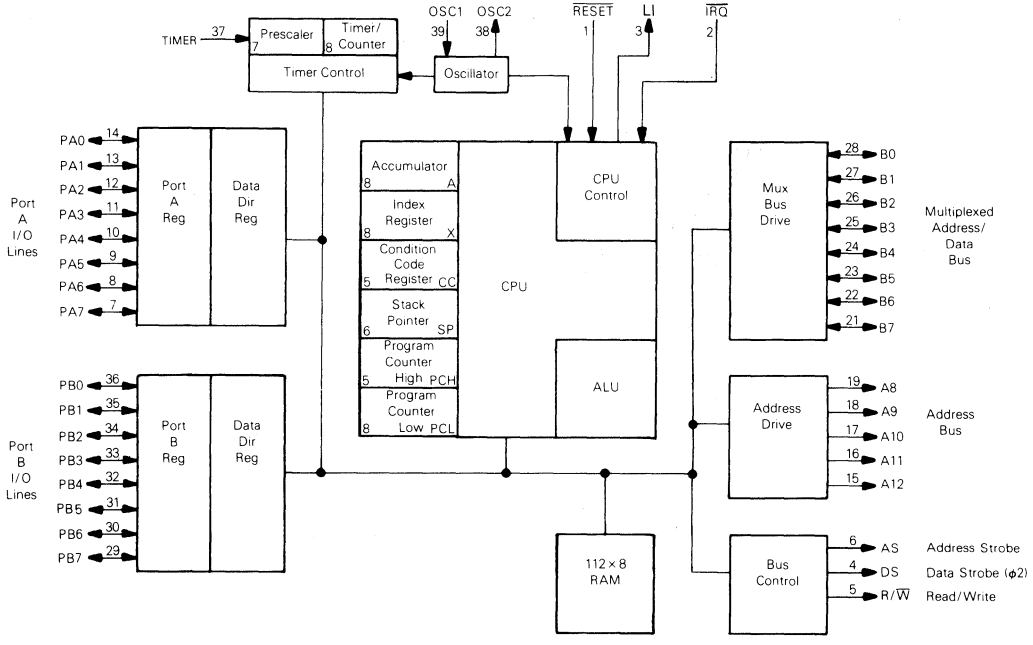
THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic Chip-Carrier	θ_{JA}	 100 60 50 TBD	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

3

FIGURE 1 — MICROPROCESSOR BLOCK DIAGRAM



DC ELECTRICAL CHARACTERISTICS @ 3.0 V (V_{DD} = 3.0 Vdc, V_{SS} = 0, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10.0\ \mu A$)	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Total Supply Current (C_L = 50 pF – No dc Loads, t_{cyc} = 5 µs)				
Run (V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V)	I_{DD}	–	1.3	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	200	µA
Stop (Test Conditions – See Note Below)	I_{DD}	–	100	µA
Output High Voltage				
(I_{Load} = 0.25 mA) A8-A12, B0-B7, DS, AS, R/$\overline{W}$	V_{OH}	2.7	–	V
(I_{Load} = 0.1 mA) PA0-PA7, PB0-PB7	V_{OH}	2.7	–	V
Output Low Voltage				
(I_{Load} = 0.25 mA) A8-A12, B0-B7, PB0-PB7, DS, AS, R/$\overline{W}$, PA0-PA7	V_{OL}	–	0.3	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
$\overline{TIMER}$, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	–	V
OSC1	V_{IH}	2.1	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.5	V
Frequency of Operation				
Crystal	f_{osc}	–	1.0	MHz
External Clock	f_{osc}	dc	1.0	MHz
Input Current				
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	µA
Hi-Z Output Leakage				
PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	–	±10	µA
Capacitance				
$\overline{RESET}$, $\overline{IRQ}$, TIMER	C_{in}	–	8.0	pF
Capacitance				
DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
$V_{IH} = V_{DD} - 0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode I_{DD} is affected linearly by this capacitance.

DC ELECTRICAL CHARACTERISTICS @ 5.0 V (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10.0\ \mu A$)	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Total Supply Current (C_L = 130 pF — On Bus, C_L = 50 pF — On Ports, No dc Loads, t_{cyc} = 1.0 µs, V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V) Run	I_{DD}	–	10	mA
$\overline{\text{Wait}}$ (Test Conditions — See Note Below)	I_{DD}	–	1.5	mA
$\overline{\text{Stop}}$ (Test Conditions — See Note Below)	I_{DD}	–	200	µA
Output High Voltage (I_{Load} = 1.6 mA) A8-A12, B0-B7, DS, AS, R/$\overline{W}$	V_{OH}	4.1	–	V
(I_{Load} = 0.36 mA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
Output Low Voltage (I_{Load} = 1.6 mA) A8-A12, B0-B7, PA0-PA7, PB0-PB7, DS, AS, R/$\overline{W}$	V_{OL}	–	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, B0-B7	V_{IH}	$V_{DD}-2.0$	–	V
$\overline{\text{TIMER}}$, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$	V_{IH}	$V_{DD}-0.8$	–	V
OSC1	V_{IH}	$V_{DD}-1.5$	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.8	V
Frequency of Operation Crystal	f_{osc}	–	5.0	MHz
External Clock	f_{osc}	dc	5.0	MHz
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	I_{in}	–	±1	µA
Hi-Z Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	–	±10	µA
Capacitance $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER	C_{in}	–	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

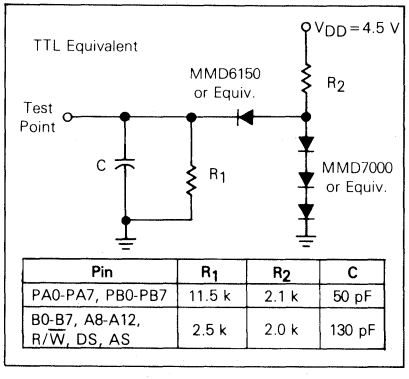
NOTE: Test conditions for Quiescent Current Values are:
- Port A and B programmed as inputs.
- V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
- $V_{IH} = V_{DD}-0.2$ V for $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, and TIMER.
- OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} − 0.2 V
- OSC2 output load (including tester) is 35 pF maximum.
- Wait mode (I_{DD}) is affected linearly by this capacitance.

TABLE 1 — CONTROL TIMING (V_{SS} = 0, $T_A = T_L$ to T_H)

		V_{DD} = 3.0 V, f_{osc} = 1 MHz			V_{DD} = 5.0 V ± 10%, f_{osc} = 5.0 MHz			
Characteristics	Symbol	Min	Typ	Max	Min	Typ	Max	Unit
I/O Port Timing — Input Setup Time (Figure 3)	t_{PVASL}	500	–	–	250	–	–	ns
Input Hold Time (Figure 3)	t_{ASLPX}	100	–	–	100	–	–	ns
Output Delay Time (Figure 3)	t_{ASLPV}	–	–	0	–	–	0	ns
Interrupt Setup Time (Figure 6)	t_{ILASL}	2	–	–	0.4	–	–	μs
Crystal Oscillator Startup Time (Figure 5)	t_{OXOV}	–	30	300	–	15	100	ms
Wait Recovery Startup Time (Figure 7)	t_{IVASH}	–	–	10	–	–	2	μs
Stop Recovery Startup Time (Crystal Oscillator) (Figure 8)	t_{ILASH}	–	30	300	–	15	100	ms
Required Interrupt Release (Figure 6)	t_{DSLIH}	–	–	5	–	–	1.0	μs
Timer Pulse Width (Figure 7)	t_{TH}, t_{TL}	0.5	–	–	0.5	–	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	5.5	–	–	1.5	–	–	μs
Timer Period (Figure 7)	t_{TLTL}	1.0	–	–	1.0	–	–	t_{cyc}
Interrupt Pulse Width Low (Figure 16)	t_{ILIH}	1.0	–	–	1.0	–	–	t_{cyc}
Interrupt Pulse Period (Figure 16)	t_{ILIL}	*	–	–	*	–	–	t_{cyc}
Oscillator Cycle Period (1/5 of t_{cyc})	t_{OLOL}	1000	–	–	200	–	–	ns
OSC1 Pulse Width High	t_{OH}	350	–	–	75	–	–	ns
OSC1 Pulse Width Low	t_{OL}	350	–	–	75	–	–	ns

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

FIGURE 2 — EQUIVALENT TEST LOADS



Pin	R_1	R_2	C
PA0-PA7, PB0-PB7	11.5 k	2.1 k	50 pF
B0-B7, A8-A12, R/W, DS, AS	2.5 k	2.0 k	130 pF

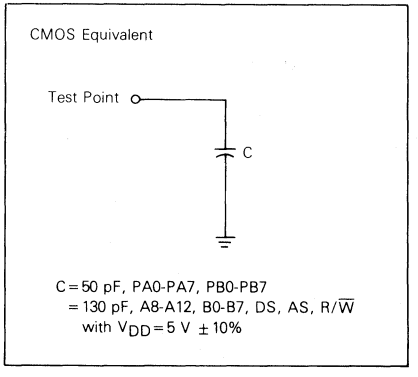
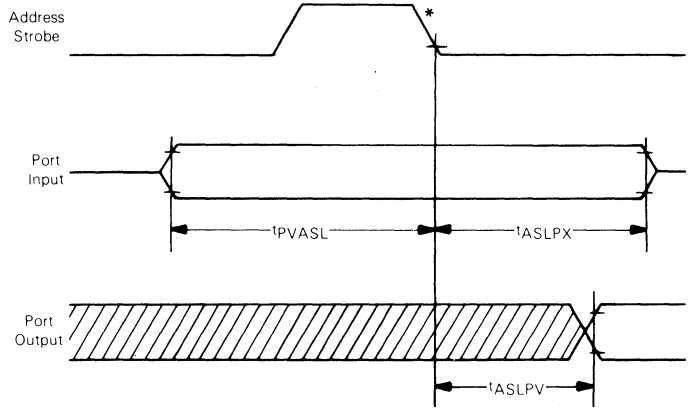


FIGURE 3 — I/O PORT TIMING
(V_{Low} = 0.8 V, V_{High} = V_{DD} − 2.0 V, V_{DD} = 5.0 ± 10%
$T_A = T_L$ to T_H, C_L on Port = 50 pF, f_{osc} = 5 MHz)

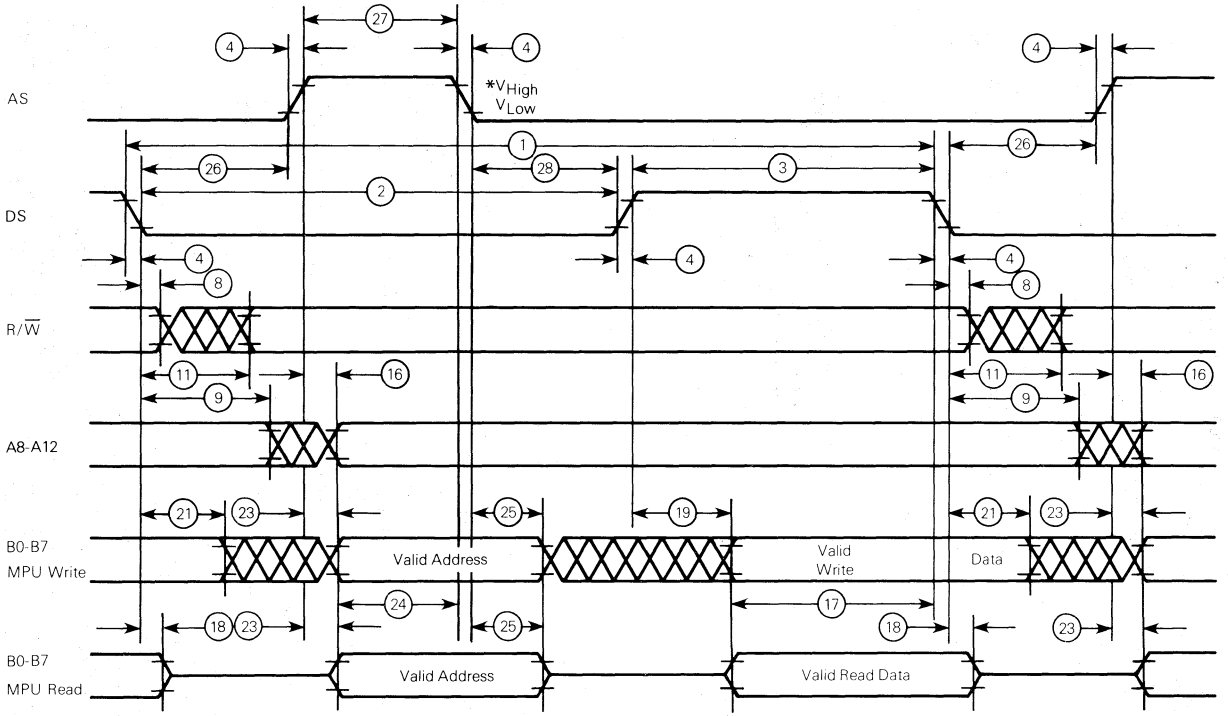


* The address strobe of the first cycle of the next instruction.

TABLE 2 — BUS TIMING ($T_A = T_L$ to T_H, V_{SS} = 0 V) See Figure 4

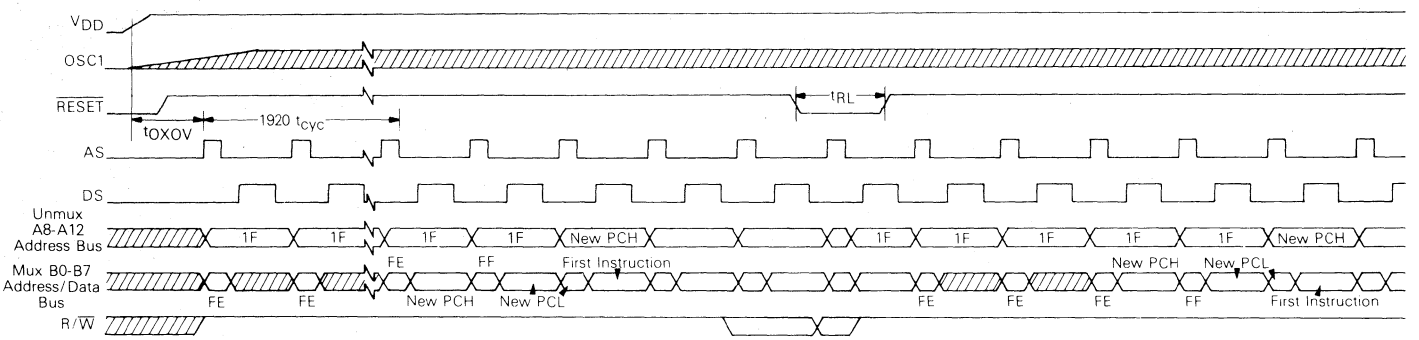
Num	Characteristics	Symbol	f_{osc} = 1 MHz V_{DD} = 3.0 V 50 pF Load		f_{osc} = 5 MHz V_{DD} = 5.0 V ± 10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	dc	1000	dc	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	800	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	Write Data Hold	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns

FIGURE 4 — MC146805E2 BUS TIMING

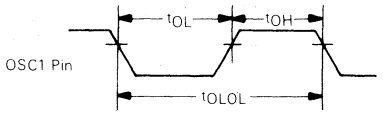


* V_{High} = 2.0 V, V_{Low} = 0.5 V for V_{DD} = 3 V for outputs only.
V_{High} = V_{DD} − 2.0 V, V_{Low} = 0.8 V for V_{DD} = 5 V ± 10% for outputs only.

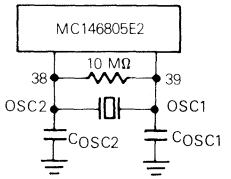
FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$ TIMING



Oscillator Waveform



Crystal Oscillator Connections



Crystal Parameters Representative Frequencies

	5.0 MHz	4.0 MHz	1.0 MHz
R_S max	50Ω	75Ω	400Ω
C0	8 pF	7 pF	5 pF
C1	0.02 pF	0.012 pF	0.008 pF
Q	50 k	40 k	30 k
C_{OSC1}	15-30 pF	15-30 pF	15-40 pF
C_{OSC2}	15-25 pF	15-30 pF	15-30 pF

Crystal Circuit

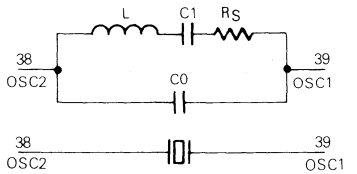
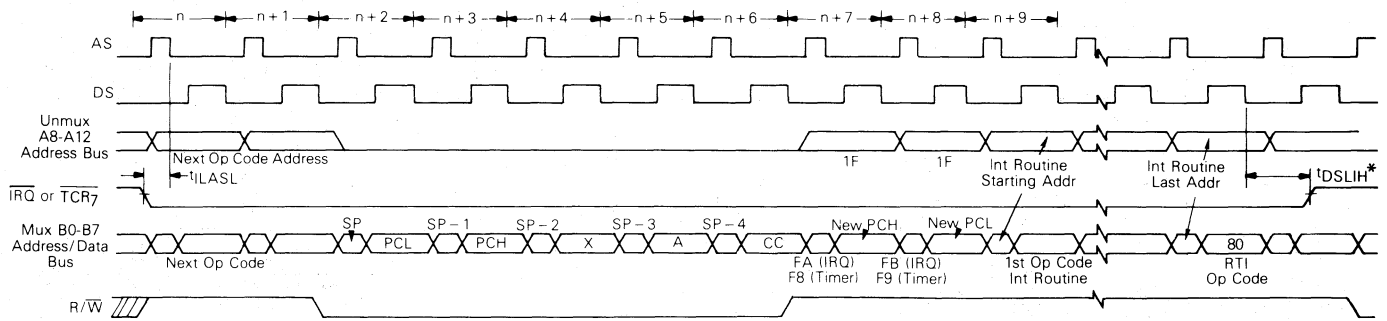


FIGURE 6 — $\overline{IRQ}$ AND $\overline{TCR_7}$ INTERRUPT TIMING



*t_{DSLIH} — The interrupting device must release the $\overline{IRQ}$ line within this time to prevent subsequent recognition of the same interrupt.

FIGURE 7 — TIMER INTERRUPT AFTER WAIT INSTRUCTION: TIMING

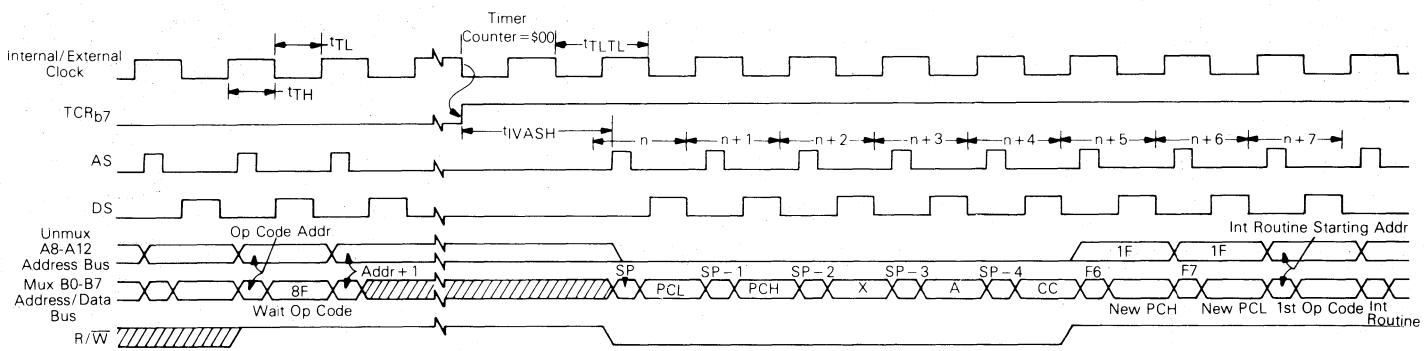
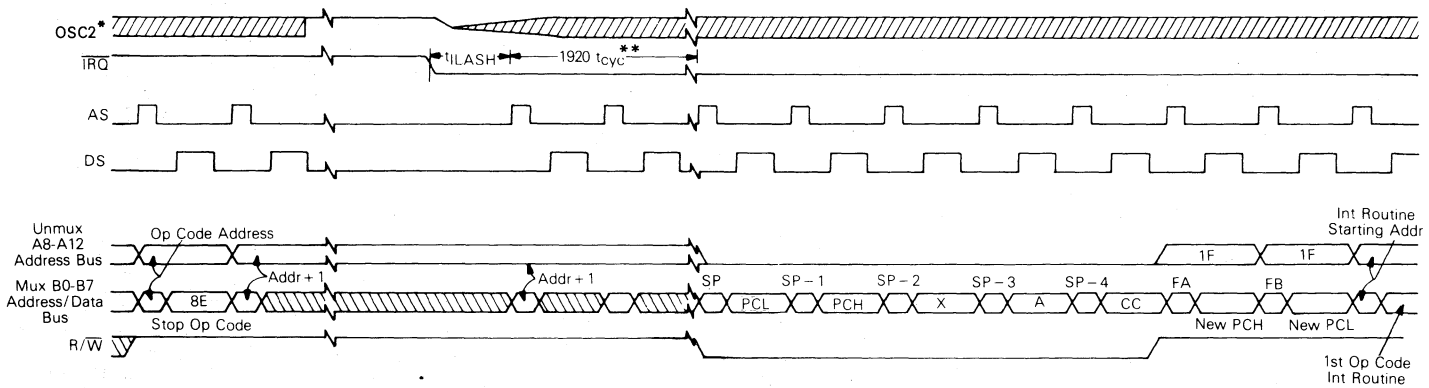


FIGURE 8 — INTERRUPT RECOVERY FROM STOP INSTRUCTION: TIMING



* Represents the internal gating of the OSC1 input pin.

** t_{cyc} is one instruction cycle (for $f_{osc} = 5$ MHz, $t_{cyc} = 1\ \mu s$)

FUNCTIONAL PIN DESCRIPTION

V_{DD} AND V_{SS}

V_{DD} and V_{SS} provide power to the chip. V_{DD} provides power and V_{SS} is ground.

$\overline{IRQ}$ (MASKABLE INTERRUPT REQUEST)

$\overline{IRQ}$ is both a level-sensitive and edge-sensitive input which can be used to request an interrupt sequence. The MPU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MPU begins an interrupt sequence at the end of the current instruction. The interrupt circuit recognizes both a "wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt section for more details). $\overline{IRQ}$ requires an external resistor to V_{DD} for "wire OR" operation.

3

$\overline{RESET}$

The $\overline{RESET}$ input is not required for start-up but can be used to reset the MPU internal state and provide an orderly software start-up procedure. Refer to the Reset section for a detailed description.

TIMER

The TIMER input is used for clocking the on-chip timer. Refer to Timer section for a detailed description.

AS (ADDRESS STROBE)

Address strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus. The AS line is used to demultiplex the eight least significant address bits from the data bus. A latch controlled by address strobe should capture addresses on the negative edge. This output is capable of driving one standard TTL load and 130 pF and is available at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP states.

DS (DATA STROBE)

This output is used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write. DS also occurs when the MPU does a data transfer to or from the MPU internal memory. Refer to Table 2 and Figure 4 for timing characteristics. This output is capable of driving one standard TTL load and 130 pF. DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP state. Some bus cycles are redundant reads of opcode bytes.

R/$\overline{W}$ (READ/WRITE)

The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next data strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read). The R/$\overline{W}$ output is capable of driving one standard TTL load and 130 pF. The normal standby state is read (high).

A8-A12 (HIGH ORDER ADDRESS LINES)

The A8-A12 output lines constitute the higher order nonmultiplexed addresses. Each output line is capable of driving one standard TTL load and 130 pF.

B0-B7 (ADDRESS/DATA BUS)

The B0-B7 bidirectional lines constitute the lower order addresses and data. These lines are multiplexed, with address present at address strobe time and data present at data strobe time. When in the data mode, these lines are bidirectional, transferring data to and from memory and peripheral devices as indicated by the R/$\overline{W}$ pin. As outputs in either the data or address modes, these lines are capable of driving one standard TTL load and 130 pF.

OSC1, OSC2

The MC146805E2 provides for two types of oscillator inputs — crystal circuit or external clock. The two oscillator pins are used to interface to a crystal circuit, as shown in Figure 5. If an external clock is used, it must be connected to OSC1. The input at these pins is divided by five to form the cycle rate seen on the AS and DS pins. The frequency range is specified by f_{OSC}. The OSC1 to bus transitions relationships are provided in Figure 9 for system designs using oscillators slower than 5 MHz.

CRYSTAL — The circuit shown in Figure 5 is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electrical characteristics table. An external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 10.

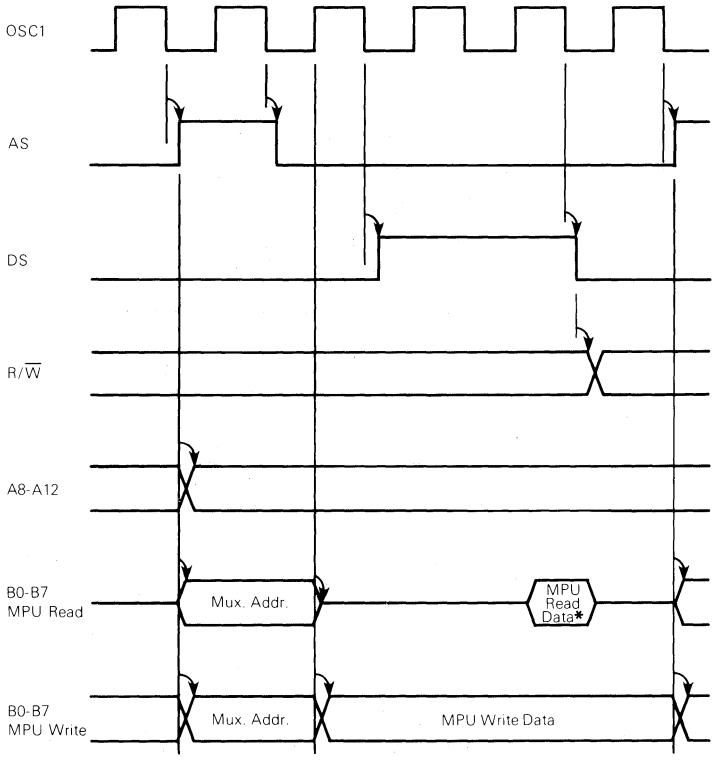
LI (LOAD INSTRUCTION)

This output is used to indicate that a fetch of the next opcode is in progress. LI remains low during an external or timer interrupt. The LI output is used only for certain debugging and test systems. For normal operations this pin is not connected. The LI output is capable of driving two standard LSTTL loads and 50 pF. This signal overlaps data strobe.

PA0-PA7

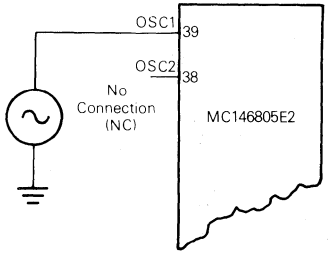
These eight pins constitute input/output port A. Each line is individually programmed to be either an input or output under software control via its data direction register as shown in Figure 11(b). An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1", and as an input when it is set to a "0". In the output mode the bits are latched and appear on the corresponding output pins. An MPU read of the port bits programmed as outputs reflects the last value written to that location. When programmed as an input, the input data bit(s) are not latched. An MPU read of the port bits programmed as inputs reflects the current status of the corresponding input pins. The I/O port timing is shown in Figure 3. See typical I/O port circuitry in Figure 11. During a power-on reset or external reset, all lines are configured as inputs (zero in data direction register). The output port register is not initialized by reset. The TTL compatible three-state output buffers are capable of driving one standard TTL load and 50 pF. The DDR is a read/write register.

FIGURE 9 — OSC1 TO BUS TRANSITIONS



*Read data "latched" on DS fall.

FIGURE 10 — EXTERNAL CLOCK CONNECTION



PB0-PB7

These eight pins interface with input/output port B. Refer to PA0-PA7 description for details of operation.

MEMORY ADDRESSING

The MC146805E2 is capable of addressing 8192 bytes of memory and I/O registers. The address space is divided into internal memory space and external memory space, as shown in Figure 12.

The internal memory space is located within the first 128 bytes of memory (first half of page zero) and is comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The MPU can read from or write to any of these locations. A program write to on-chip locations is repeated on the external bus to permit off-chip memory to duplicate the content of on-chip memory. Program reads to on-chip locations also appear on the external bus, but the MPU accepts data only from the addressed on-chip location. Any read data appearing on the input bus is ignored.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $007F and it is decremented as data is pushed onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allotted stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

All memory locations above location $007F are part of the external memory map. In addition, ten locations in the I/O portion of the lower 128 bytes of memory space, as shown in

FIGURE 11 — TYPICAL PORT I/O CIRCUITRY

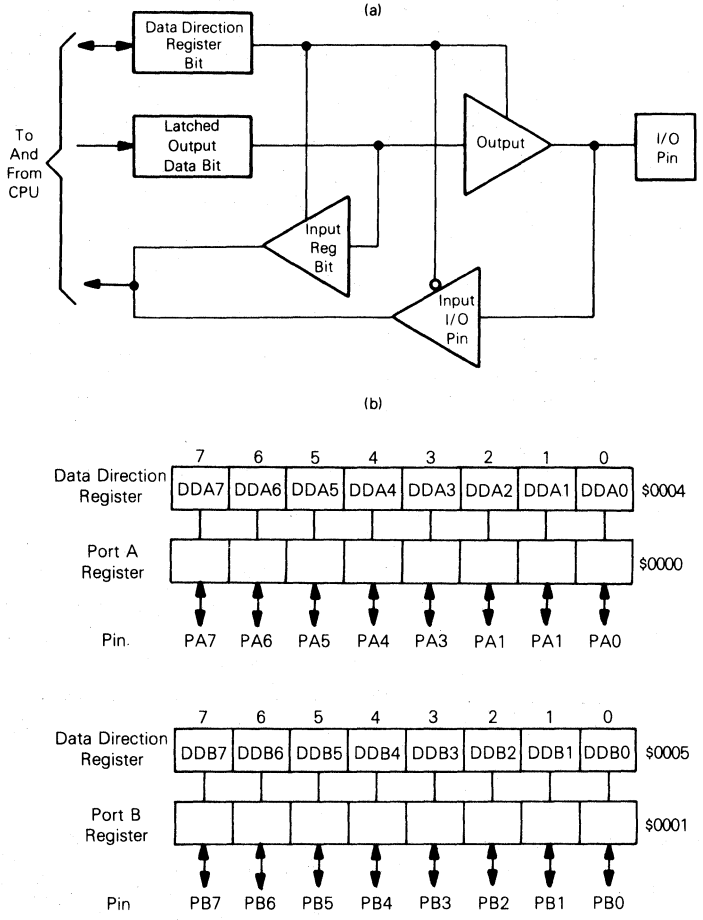


TABLE 3 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

Figure 12, are part of the external memory map. All of the external memory space is user definable except the highest 10 locations. Locations $1FF6 to $1FFF of the external address space are reserved for interrupt and reset vectors (see Figure 12).

REGISTERS

The MC146805E2 contains five registers as shown in the programming model in Figure 13. The interrupt stacking order is shown in Figure 14.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands and results of arithmetic calculations and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

FIGURE 12 — MPU ADDRESS MAP

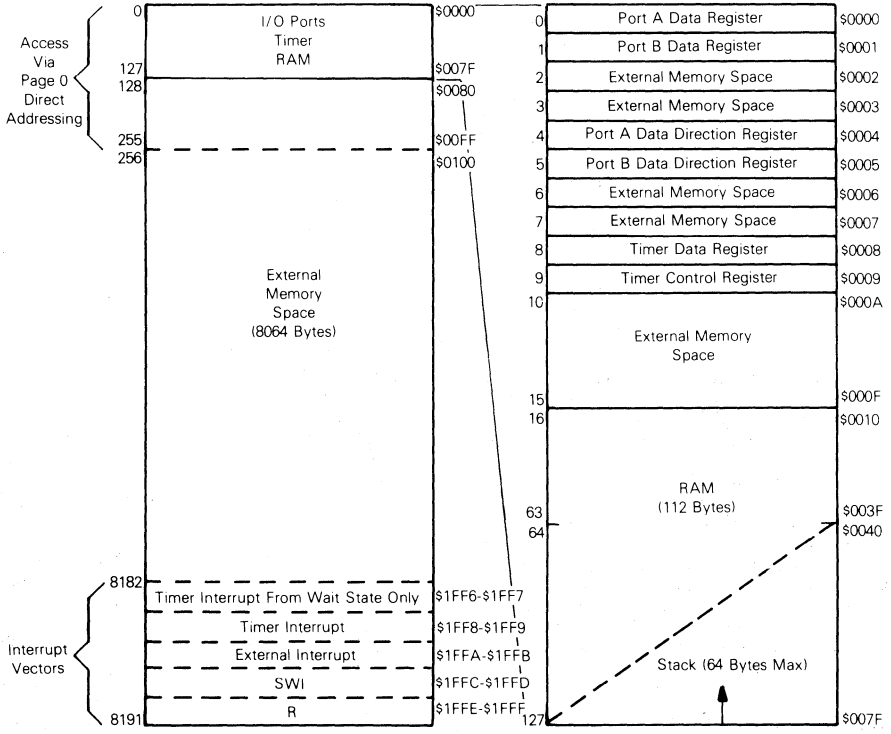


FIGURE 13 – PROGRAMMING MODEL

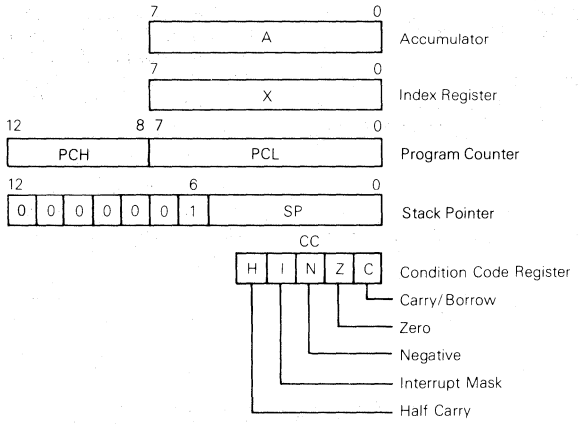
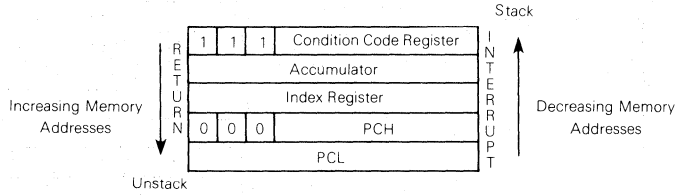


FIGURE 14 – STACKING ORDER



NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently set to 0000001. They are appended to the six least significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer "wraps around" and points to its upper limit, thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each of the five bits is explained below.

HALF CARRY BIT (H) – The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal addition subroutines.

INTERRUPT MASK BIT (I) – When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and will be processed when the I bit is next cleared.

NEGATIVE BIT (N) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logical one).

ZERO BIT (Z) – When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

CARRY BIT (C) – The C bit is set when a carry or a borrow out of the ALU occurs during an arithmetic instruction. The C bit is also modified during bit test, shift, rotate, and branch types of instruction.

RESETS

The MC146805E2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$ (PIN #1)

The $\overline{\text{RESET}}$ input pin is used to reset the MPU and provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{RL}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt trigger to improve its noise immunity capability.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{CYC} delay from the time of the first oscillator operation. If the external reset pin is low at the end of the 1920 t_{CYC} time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (bit 7) is cleared to a "0".
- Timer control register interrupt mask bit (bit 6) is set to a "1".
- All data direction register bits are cleared to a "0" (inputs).
- Stack pointer is set to $007F.
- The address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports), the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

The MC146805E2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a non-maskable software interrupt (SWI). Systems often require that normal processing be interrupted so that some external event may be serviced.

Interrupts cause the processor registers to be saved on the stack and the interrupt mask set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and a return to normal processing. The stacking order is shown in Figure 14.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction excution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction. Refer to Figure 15 for the interrupt and instruction processing sequence.

TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode, in which case users of mask versions BP4XX-XX and AW9XXXX should refer to the appendix for additional information regarding exceptions to this function. The contents of $1FF6 and $1FF7 specify the service routine. Also, software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{\text{IRQ}}$ is "low," then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $1FFA and $1FFB. The interrupt logic recognizes both a "wire ORed" level and pulses on the external interrupt line. Figure 16 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized. The second method is single pulses on the interrupt line spaced far enough apart to be serviced. Users of mask versions BP4XXXX and AW9XXXX should refer to the appendix regarding exceptions to this function. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse ocurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (one cycle $t_{CYC} = 5/f_{OSC}$) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 6.

SOFTWARE INTERRUPT (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 15 for interrupt and instruction processing flowchart.

STOP

The STOP instruction places the MC146805E2 in a low power consumption mode. In the STOP function the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 17. The DS and AS lines go to a low state and the R/$\overline{\text{W}}$ line goes to a high state.

FIGURE 15 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

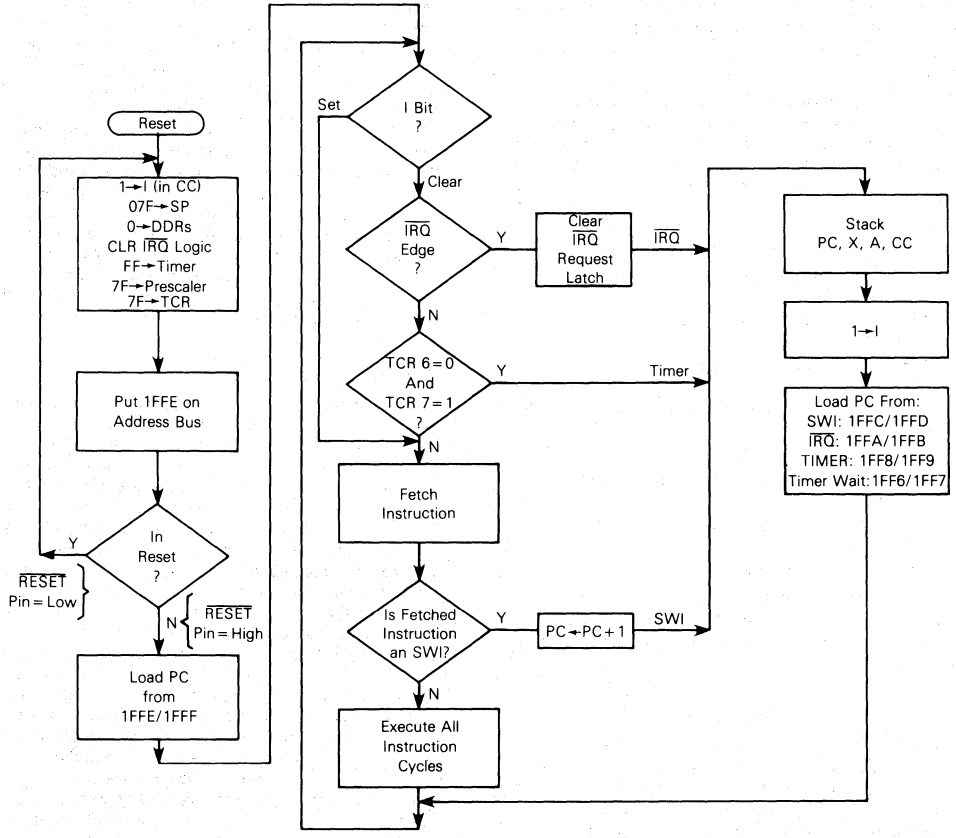
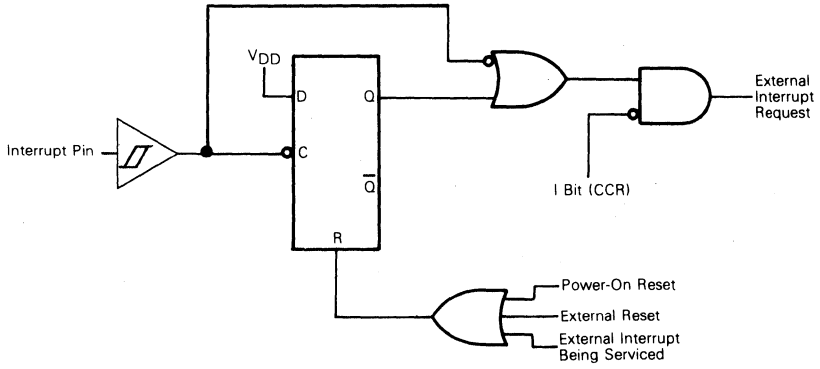
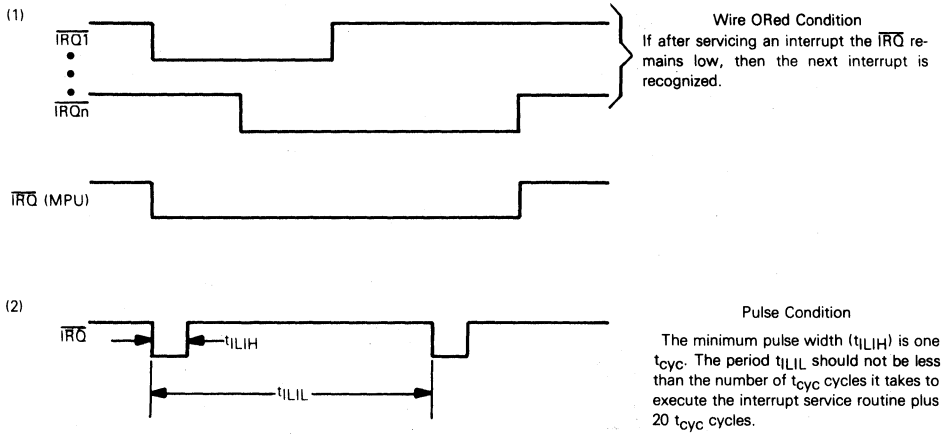


FIGURE 16 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Wire ORed Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

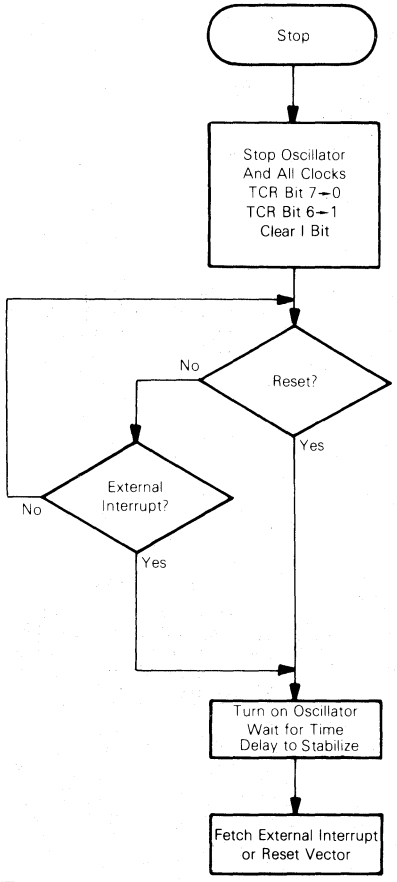
Pulse Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

The multiplexed address/data bus goes to the data input state (as shown in Figure 8). The high order address lines remain at the address of the next instruction. The MPU remains in the STOP mode until an external interrupt or reset occurs.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

FIGURE 17 — STOP FUNCTION FLOWCHART



WAIT

The WAIT instruction places the MC146805E2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1. In the WAIT function, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 18. Thus, all internal processing is halted except the timer which is allowed to count in a normal sequence. The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state (as shown in Figure 7). The high order address lines remain at the address of the next instruction. The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MPU is no longer in the WAIT mode.

TIMER

The MPU timer contains a single 8-bit software programmable counter (timer data register) with 7-bit software selectable prescaler. Figure 19 shows a block diagram of the timer. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer interrupt vector from locations $1FF8 and $1FF9 in order to begin servicing the interrupt. If the MPU is interrupted by the timer while in the WAIT mode, the interrupt vector fetch would be from locations $1FF6 and $1FF7.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The content of the counter becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software. If a read occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

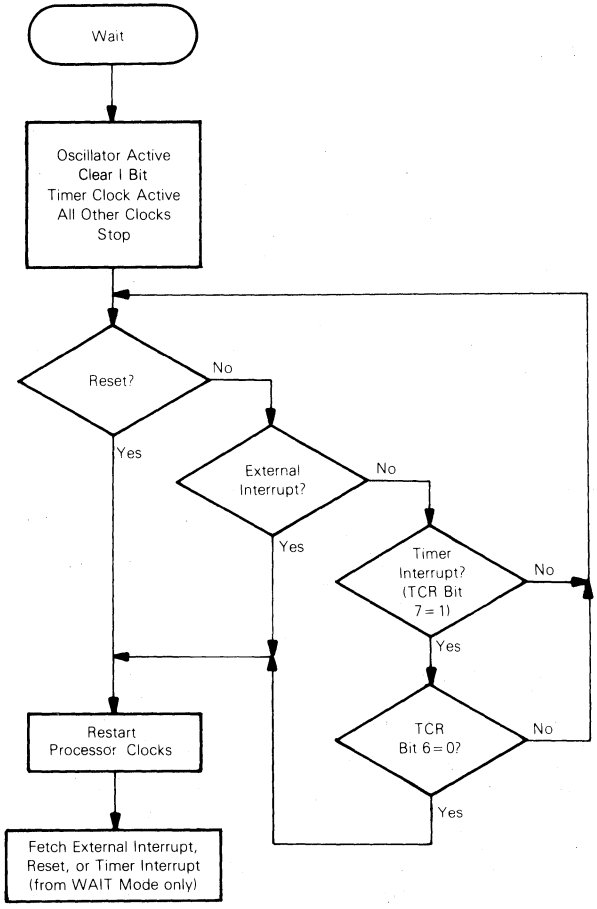
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals 1, which allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode, depending on the value written to the TCR4, TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a "0", the input to the timer is from an internal clock and the external TIMER input is disabled. The internal clock mode can be

FIGURE 18 – WAIT FUNCTION FLOWCHART



used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with address strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock and therefore accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

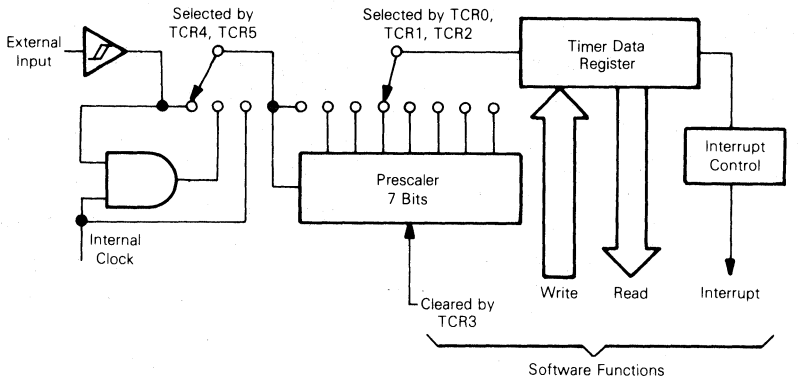
If TCR4 = 0 and TCR5 = 1, then all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR4 = 1 and TCR5 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The external TIMER pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the timer subsystem.

FIGURE 19 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register are clocked on the falling edge of the internal clock (AS) or external input.
2. Timer data register is written to during data strobe (DS) and counts down continuously.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are read/write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".

1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic "1" it inhibits the timer interrupt to the processor.

1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock (unaffected by $\overline{\text{RESET}}$).

1 — Select external clock source.
0 — Select internal clock source (AS).

TCR4 — External enable bit: control bit used to enable the external TIMER pin (unaffected by $\overline{\text{RESET}}$).

1 — Enable external TIMER pin.
0 — Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal clock (AS) to timer
0	1	AND of internal clock (AS) and TIMER pin to timer
1	0	Inputs to timer disabled
1	1	TIMER pin to timer

TCR3 — Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0" (unaffected by $\overline{\text{RESET}}$).

TCR2, TCR1, TCR0 — Prescaler address bits: decoded to select one of eight outputs of the prescaler (unaffected by $\overline{\text{RESET}}$).

Prescaler			
TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MPU has a set of 61 basic instructions. 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. 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 operand. Refer to Table 4.

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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 9.

OPCODE MAP SUMMARY

Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MPU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 9 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 10.

The term "effective address" or EA is used in describing the various addressing modes, and is defined as the address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by," and a colon indicates concatenation of two bytes.

INHERENT

In inherent instructions all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediatley following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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 lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and up to 128 bytes of off-chip ROM. Direct addressing is efficient in both memory and speed.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High ← 0; Address Bus Low ← (PC + 1)

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

Address Bus High ← 0, Address Bus Low ← X

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$

Address Bus High ← K; Address Bus Low ← X + (PC + 1)

where: K = The carry from the addition of X + (PC + 1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode the effective address is the sum of the contents of the unsiged 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset — 8 or 16 bit. The content of the index register is not changed.

$EA = X + [(PC + 1):(PC + 2)]$; $PC \leftarrow PC + 3$
Address Bus High $\leftarrow (PC + 1) + K$
Address Bus Low $\leftarrow K + (PC + 2)$
where: K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is used only in branch instructions. In relative addressing the content of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

$EA = PC + 2 + (PC + 1)$; $PC \leftarrow EA$ if branch is taken;
otherwise, $PC \leftarrow PC + 2$

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the second to address the byte which contains the bit of interest.

$EA = (PC + 1)$; $PC \leftarrow PC + 2$
Address Bus High $\leftarrow 0$; Address Bus Low $\leftarrow (PC + 1)$

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit addressing, and relative addressing. The bit address and condition (set or clear) to be tested are part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

$EA1 = (PC + 1)$
Address Bus High $\leftarrow 0$; Address Bus Low $\leftarrow (PC + 1)$
$EA2 = PC + 3 + (PC + 2)$; $PC \leftarrow EA2$ if branch taken;
otherwise, $PC \leftarrow PC + 3$

SYSTEM CONFIGURATION

Figures 20 through 25 show in general terms how the MC146805E2 bus structure may be utilized. Specified interface details vary with the various peripheral and memory devices employed.

Table 11 provides a detailed description of the information present on the bus, read/write (R/$\overline{W}$) pin and the load instruction (LI) pin during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction.

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	—	—	—	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	—	—	—	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	—	—	—

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9 – INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
ROR	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- Λ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

TABLE 10 — MC146805 CMOS INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

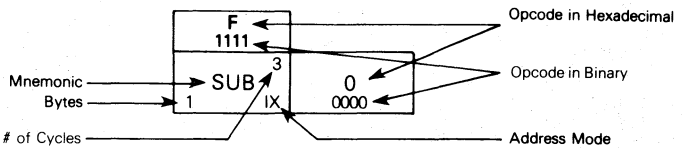


FIGURE 20 — CONNECTION TO CMOS PERIPHERALS

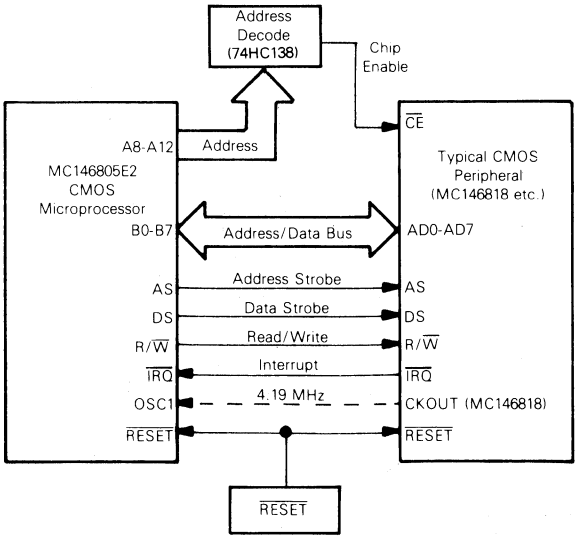


FIGURE 21 — CONNECTION TO CMOS MULTIPLEXED MEMORIES

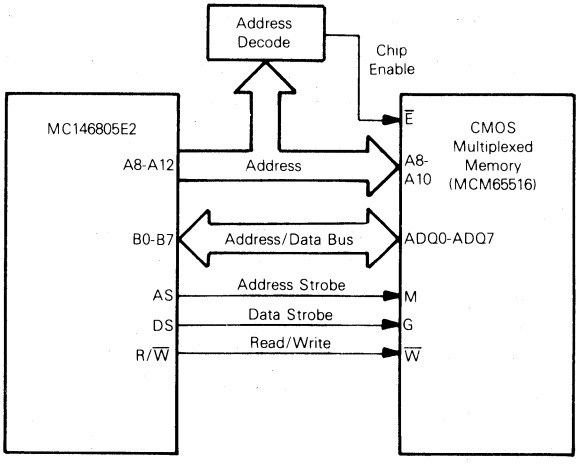
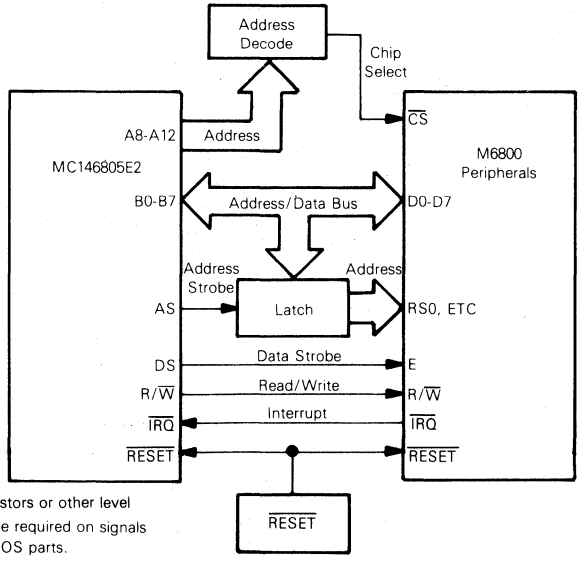


FIGURE 22 — CONNECTION TO M6800 PERIPHERALS



NOTE: In some cases, pullup resistors or other level shifting techniques may be required on signals going from NMOS to CMOS parts.

FIGURE 23 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS ROM AND EPROM

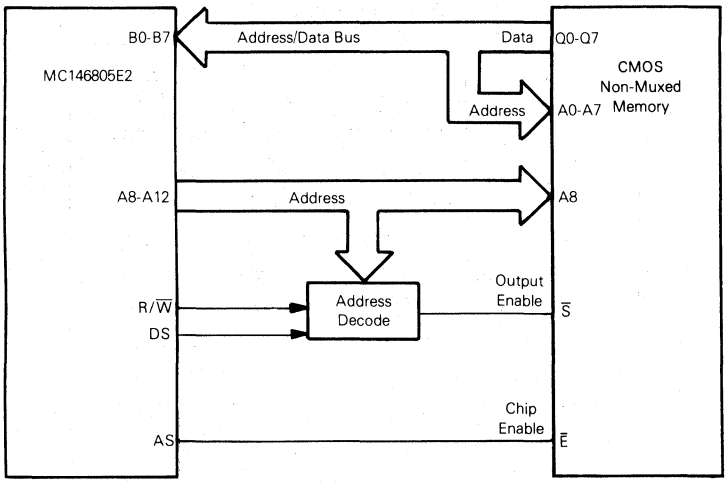


FIGURE 24 — CONNECTION TO STATIC CMOS RAMS

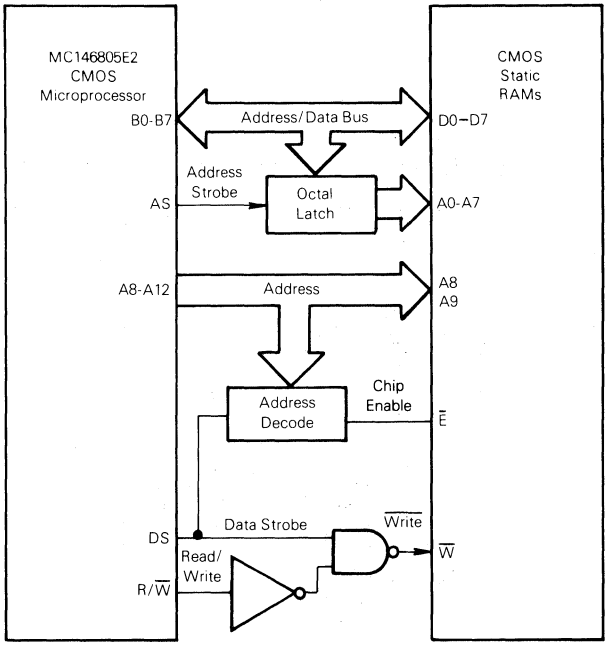


FIGURE 25 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS RAM

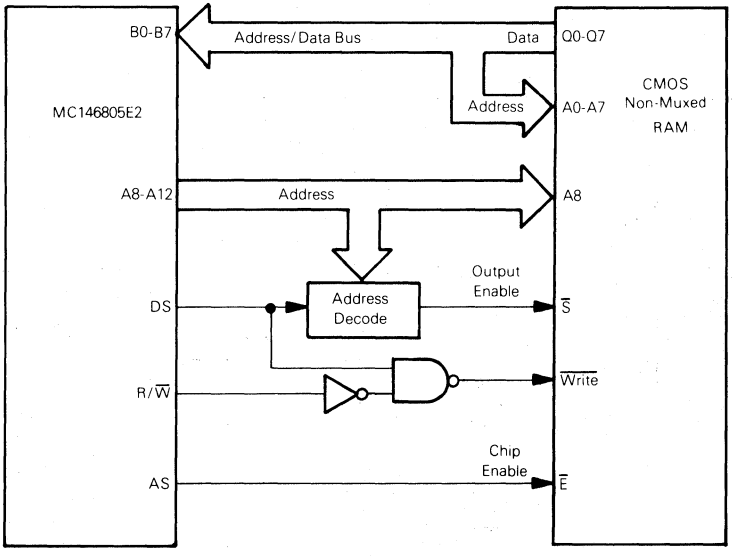


TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/W Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Op Code Next Instruction Op Code Next Instruction
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Op Code Next Instruction
RTS	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 New Op Code Address	1 1 1 1 1 1	1 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
SWI	10	1 2 3 4 5 6 7 8 9 10	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer −1 Stack Pointer −2 Stack Pointer −3 Stack Pointer −4 Vector Address 1FFC (Hex) Vector Address 1FFD (Hex) Interrupt Routine Starting Address	1 1 0 0 0 0 0 1 1 1	1 0 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Return Address (LO Byte) Return Address (HI Byte) Contents of Index Register Contents of Accumulator Contents of CC Register Address of Int. Routine (HI Byte) Address of Int. Routine (LO Byte) Interrupt Routine First Opcode
RTI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2 Stack Pointer +3 Stack Pointer +4 Stack Pointer +5 New Op Code Address	1 1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 0 0	Op Code Op Code Next Instruction Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data Irrelevant Data New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address +1	1 1	1 0	Op Code Operand Data
Bit Set/Clear						
BSET n BCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Address of Operand Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Operand Data Operand Data Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	1 2 3 4 5	Op Code Address Op Code Address +1 Address of Operand Op Code Address +2 Op Code Address +2	1 1 1 1 1	1 0 0 0 0	Op Code Address of Operand Operand Data Branch Offset Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1 2 3	Op Code Address Op Code Address +1 Op Code Address +1	1 1 1	1 0 0	Op Code Branch Offset Branch Offset
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address +1 Op Code Address +1 Subroutine Starting Address Stack Pointer Stack Pointer −1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Branch Offset Branch Offset First Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Direct						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Address of Operand	1	0	Operand Data
		4	Op Code Address + 2	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Adrress + 1	1	0	Address of Operand
		3	Op Code Address + 1	1	0	Address of Operand
		4	Address of Operand	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand
		3	Operand Address	1	0	Current Operand Data
		4	Operand Address	1	0	Current Operand Data
		5	Operand Address	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Subroutine Address (LO Byte)
		3	Subroutine Starting Address	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)
Extended						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Jump Address (HI Byte)
		3	Op Code Address + 2	1	0	Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address Operand (LO Byte)
		4	Address of Operand	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Operand (HI Byte)
		3	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		4	Op Code Address + 2	1	0	Address of Operand (LO Byte)
		5	Address of Operand	0	0	Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Address of Subroutine (HI Byte)
		3	Op Code Address + 2	1	0	Address of Subroutine (LO Byte)
		4	Subroutine Starting Address	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	Stack Pointer − 1	0	0	Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
TST	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Operand Data
		4	Op Code Address + 1	1	0	Op Code Next Instruction
STA STX	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Op Code Address + 1	1	0	Op Code Next Instruction
		4	Index Register	0	0	Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	Current Operand Data
		4	Index Register	1	0	Current Operand Data
		5	Index Register	0	0	New Operand Data
JSR	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Index Register	1	0	1st Subroutine Op Code
		4	Stack Pointer	0	0	Return Address (LO Byte)
		5	Stack Pointer − 1	0	0	Return Address (HI Byte)

TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer − 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer − 1	0	0	Return Address (HO Byte)

TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycle #	Address Bus	$\overline{\text{RESET}}$ Pin	R/$\overline{\text{W}}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{\text{RESET}}$	5		$1FFE	0	1	0	Irrelevant Data
			$1FFE	0	1	0	Irrelevant Data
		1	$1FFE	1	1	0	Irrelevant Data
		2	$1FFE	1	1	0	Irrelevant Data
		3	$1FFE	1	1	0	Vector High
		4	$1FFF	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$1FFE	1	1	0	Irrelevant Data
		•	•	•	•	•	•
		•	•	•	•	•	•
		•	•	•	•	•	•
		1919	$1FFE	1	1	0	Irrelevant Data
		1920	$1FFE	1	1	0	Vector High
		1921	$1FFF	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycles #**	**Address Bus**	$\overline{\text{IRQ}}$ Pin	R/$\overline{\text{W}}$ Pin	LI Pin	**Data Bus**
$\overline{\text{IRQ}}$ Interrupt (Timer Vector $1FF8, $1FF9)	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP – 1	X	0	0	Return Address (HI Byte)
		5	SP – 2	X	0	0	Contents Index Reg
		6	SP – 3	X	0	0	Contents Accumulator
		7	SP – 4	X	0	0	Contents CC Register
		8	$1FFA	X	1	0	Vector High
		9	$1FFB	X	1	0	Vector Low
		10	$\overline{\text{IRQ}}$ Vector	X	1	0	Int Routine First

APPENDIX

MC146805E2 INTERRUPT CLARIFICATION

Under certain circumstances, the MC146805E2 (BP4XXXX and AW9XXXX) 8-bit Microprocessor Unit $\overline{\text{IRQ}}$ interrupt does not conform to the operation described in this Advanced Information Sheet.

1. The level sensitive $\overline{\text{IRQ}}$ mode, which is by far the most frequently used, is **FULLY OPERATIONAL:** thus, most MC146805E2 applications are unaffected. However, the edge-triggered $\overline{\text{IRQ}}$ interrupt mode **MIGHT NOT BE SERVICED** under certain programming circumstances; therefore, it is recommended that the edge-triggered mode not be used.

2. An interrupt-vector address **CAN BE** improperly generated in some circumstances. There is a possibility that when an external interrupt ($\overline{\text{IRQ}}$) and timer interrupt occur during the WAIT mode (following wait instruction), address locations $1FF2 and $1FF3 are selected instead of vector locations $1FF6 and $1FF7. There are three specific examples listed below; two of these require no action and the third has a recommended solution.

   a. Those not using the WAIT mode need not take any action.

   b. If the WAIT mode is used without external interrupt ($\overline{\text{IRQ}}$ pin held high), no precautions are required.

   c. When $\overline{\text{IRQ}}$ can be active (low) during the WAIT mode, the vector in locations $1FF6 and $1FF7 (the WAIT mode timer interrupt vector) should be duplicated in $1FF2 and $1FF3. In this way the circumstances that caused selection of the second vector do not disturb normal program execution.

On future MC146805E2 parts, no special actions will be necessary. If you have questions, contact your Motorola distributor or Motorola sales office, or contact Motorola Microprocessor Applications Engineering in Austin, Texas.

MOTOROLA

MC146805F2

Advance Information

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT
MICROCOMPUTER

3

8-BIT MICROCOMPUTER UNIT

The MC146805F2 Microcomputer Unit (MCU) belongs to the M146805 Family of Microcomputers. This 8-bit MCU contains on-chip oscillator, CPU, RAM, ROM, I/O, and TIMER. The fully static design allows operation at frequencies down to dc, further reducing its already low-power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low-power consumption constitutes an important factor.

HARDWARE FEATURES

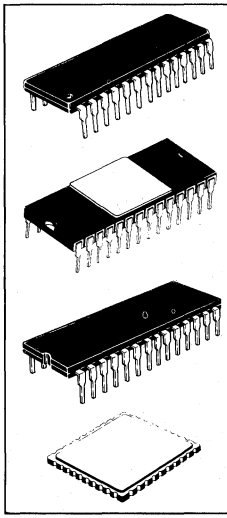
- Typical Full Speed Operating Power of 10 mW at 5 V
- Typical WAIT Mode Power of 3 mW
- Typical STOP Mode Power of 25 μW
- 8-Bit Architecture
- Fully Static Operation
- Single 3- to 6-Volt Supply
- 1089 Bytes of On-Chip User ROM
- 64 Bytes of On-Chip RAM
- Memory Mapped I/O
- 16 Bidirectional I/O Lines
- 4 Input-Only Lines
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Self-Check Mode
- Master Reset and Power-On Reset
- On-Chip Oscillator
- 1 μs Cycle Time
- 28-Pin Dual-In-Line Package
- Chip Carrier Also Available

SOFTWARE FEATURES

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Ten Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- User Callable Self-Check Routines
- Two Power Saving Standby Modes

USER SELECTABLE OPTIONS

- Crystal or Low-Cost Resistor Oscillator Option
- Oscillator Internally Divided by 2 or 4
- Interrupts Edge Sensitive Only or Level and Edge Sensitive



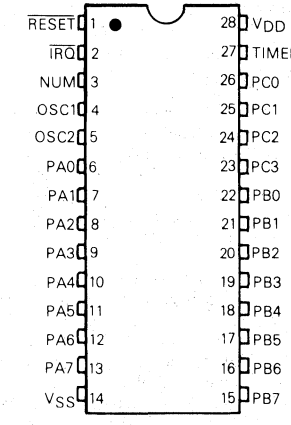
P SUFFIX
PLASTIC PACKAGE
CASE 710

L SUFFIX
CERAMIC PACKAGE
CASE 719

S SUFFIX
CERDIP PACKAGE
CASE 733

Z SUFFIX
CHIP CARRIER
CASE 761

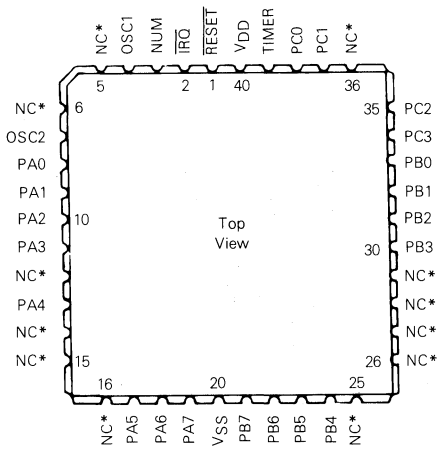
PIN ASSIGNMENT



Chip carrier pin assignments are shown on the next page.

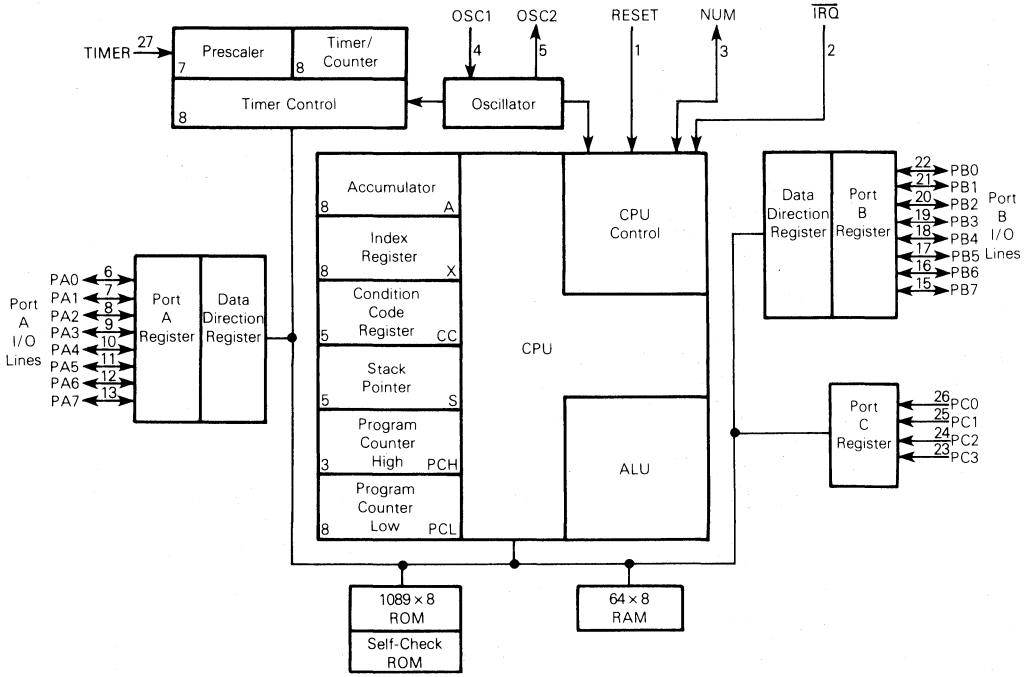
This document contains information on a new product. Specifications and information herein are subject to change without notice.

PIN ASSIGNMENT (CONTINUED)



* NC = No Connection

MC146805F2 CMOS MICROCOMPUTER



MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +6.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805F2 MC146805F2C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages 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 $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and NUM are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic Chip Carrier	θ_{JA}	 115 65 60 100	°C/W

3

DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H, unless otherwise noted) (See Note 1)

Characteristics	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \le 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Output High Voltage ($I_{Load}=-200\ \mu A$) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
Output Low Voltage, ($I_{Load}=800\ \mu A$) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{IRQ}$, $\overline{RESET}$, OCS1	V_{IH}	$V_{DD}-2.0$ $V_{DD}-0.8$	V_{DD} V_{DD}	V
Input Low Voltage, All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current ($C_L=50$ pF on Ports, No dc Loads, $t_{cyc}=1\ \mu s$) RUN ($V_{IL}=0.2$ V, $V_{IH}=V_{DD}-0.2$ V) WAIT (See Note 2) STOP (See Note 2)	I_{DD}	– – –	4 1.5 150	mA mA μA
I/O Ports Input Leakage – PA0-PA7, PB0-PB7	I_{IL}	–	±10	μA
Input Current – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	I_{in}	–	±1	μA
Output Capacitance – Ports A and B	C_{out}	–	12	pF
Input Capacitance – $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1, PC0-PC3	C_{in}	–	8	pF

NOTES:
1. Electrical Characteristics for $V_{DD}=3$ V available soon.
2. Test Conditions for I_{DD} are as follows:
   All ports programmed as inputs
   $V_{IL}=0.2$ V (PA0-PA7, PB0-PB7, PC0-PC3)
   $V_{IH}=V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, TIMER
   OSC1 input is a square wave from 0.2 V to $V_{DD}-0.2$ V (for WAIT I_{DD} measurement only)
   OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)

TABLE 1 — CONTROL TIMING CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, $T_A = T_L$ to T_H, f_{osc} = 4 MHz, t_{cyc} = 1 μs)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time – Crystal Oscillator (See Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width (See Figure 15)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (See Figure 15)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width (See Figure 7)	t_{OH}, t_{OL}	100	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal External Clock	f_{osc}	– dc	4.0 4.0	MHz

*The minimum period, t_{ILIL}, should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

3

FIGURE 1 — EQUIVALENT TEST LOAD

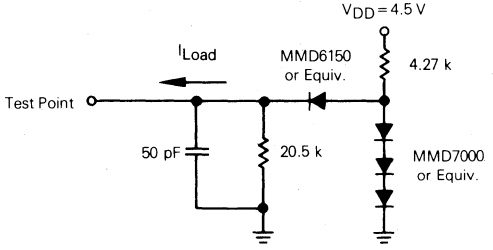


FIGURE 2 — MAXIMUM OPERATING CURRENT vs INTERNAL FREQUENCY ($T_A = T_L$ to T_H)

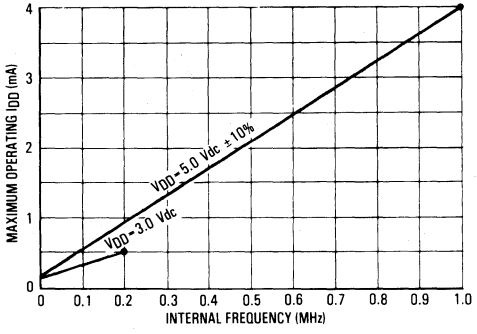


FIGURE 3 — MAXIMUM WAIT CURRENT vs INTERNAL FREQUENCY ($T_A = T_L$ to T_H)

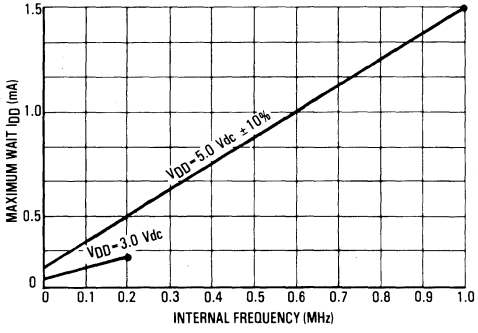


FIGURE 4 — TIMER RELATIONSHIPS

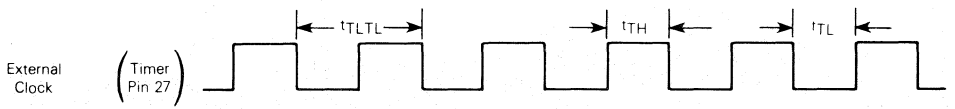
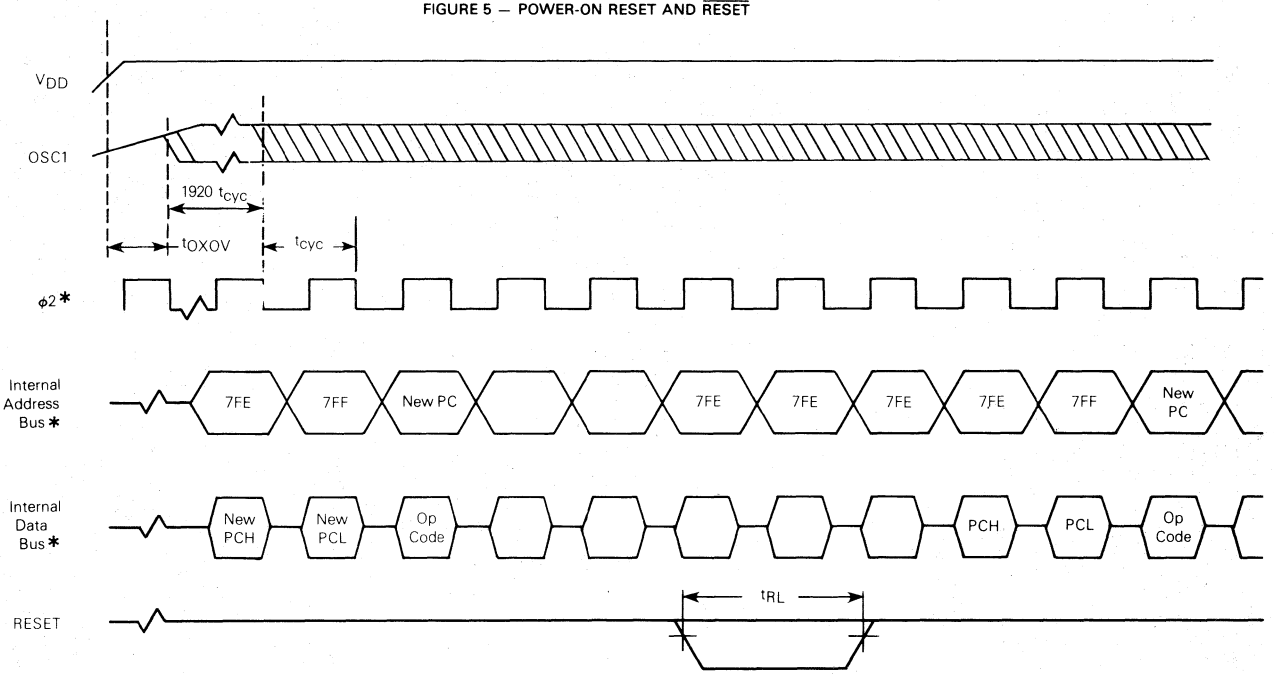
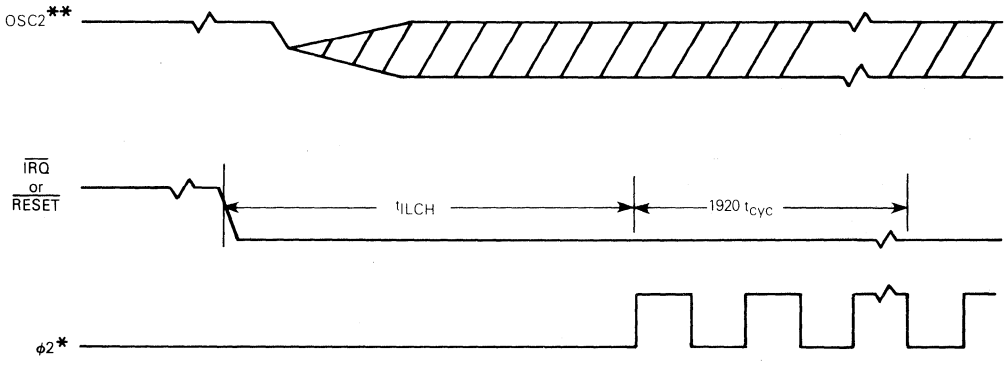


FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$



* Internal timing signal not available externally.

FIGURE 6 — STOP RECOVERY



* Internal timing signals not available externally.

** Represents the internal gating of the OSC1 input pin.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

$\overline{IRQ}$ (MASKABLE INTERRUPT REQUEST)

$\overline{IRQ}$ is photomask option selectable with the choice of interrupt sensitivity being both level and negative edge or negative edge only. The MCU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the photomask option is selected to include level sensitivity, then the $\overline{IRQ}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See the Interrupt section for more detail.

$\overline{RESET}$

The $\overline{RESET}$ input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to the Resets section for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to the Timer section for a detailed description.

NUM (NON-USER MODE)

This pin is intended for use in self-check only. User applications should leave this pin connected to ground through a 10 kilohm resistor.

OSC1, OSC2

The MC146805F2 can be configured to accept either a crystal input or an RC network. Additionally, the internal clocks can be derived from either a divide-by-two or divide-by-four of the external frequency (f_{OSC}). Both of these options are photomask selectable.

RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 7(b). The relationship between R and f_{OSC} is shown in Figure 8.

CRYSTAL — The circuit shown in Figure 7(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Table 1, Control Timing Characteristics, for limits.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 7(c). An external clock should be used with the crystal oscillator mask option only. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

PA0-PA7

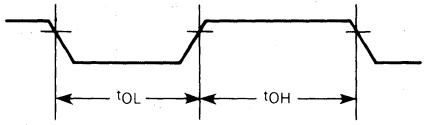
These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

FIGURE 7 – OSCILLATOR CONNECTIONS

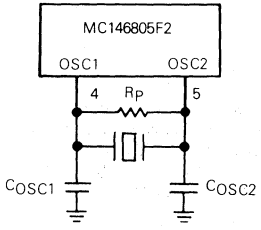
Crystal Parameters

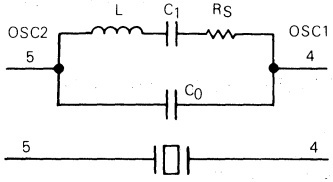
	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30 k	40 k	–

Oscillator Waveform

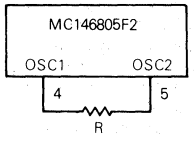


(a) Crystal Oscillator Connections and Equivalent Crystal Circuit





(b) RC Oscillator Connection



(c) External Clock Source Connections

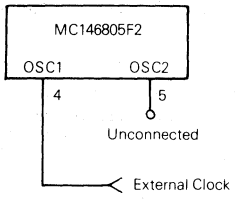
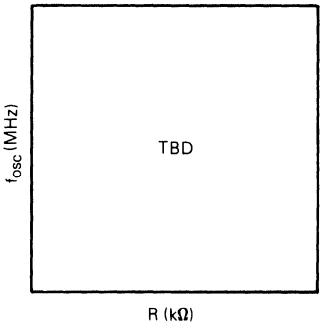


FIGURE 8 — FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

PC0-PC3

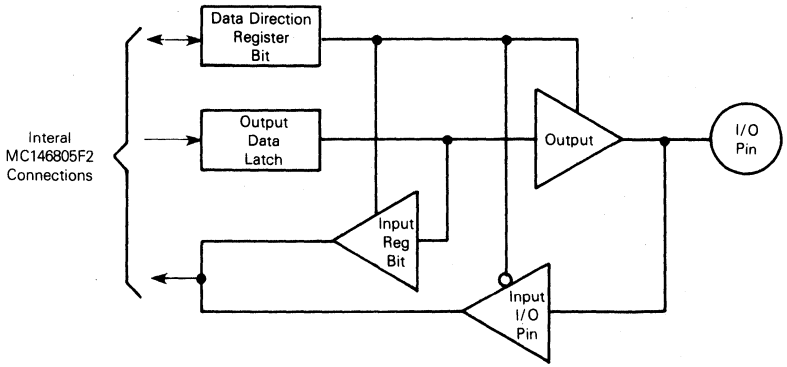
These four lines comprise Port C, a fixed input port. When Port C is read, the four most-significant bits on the data bus are "1s" . There is no data direction register associated with Port C.

INPUT/OUTPUT PROGRAMMING

Any Port A or B pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic "1". A pin is configured as an input if its corresponding DDR bit is cleared to a logic "0". At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 9 and Table 2.

FIGURE 9 — TYPICAL PORT I/O CIRCUITRY

(a)



(b)

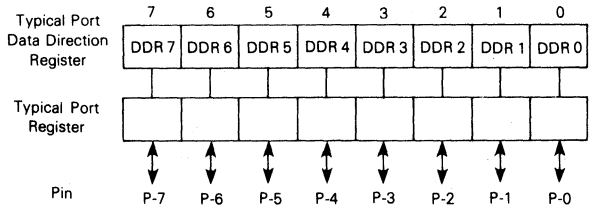


TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

SELF-CHECK

The MC146805F2 self-check is performed using the circuit in Figure 10. Self-check is initiated by tying NUM and TIMER pins to a logic "1" then executing a reset. After reset, the following five tests are executed automatically:

I/O – Functionally Exercise Ports A, B, C

RAM – Walking Bit Test

ROM – Exclusive OR with ODD "1s" Parity Result

Timer – Functionally Exercise Timer

Interrupts – Functionally Exercise External and Timer Interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

TABLE 3 – SELF-CHECK RESULTS

PB3	PB2	PB1	PB0	Remarks
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
All Cycling				Good Part
All Others				Bad Part

RAM SELF-CHECK SUBROUTINE

Returns with the Z bit clear if any error is detected; otherwise, the Z bit is set.

The RAM test must be called with the stack pointer at $7F and the accumulator zeroed. When run, the test checks every RAM cell except for $7F and $7E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $78B.)

ROM CHECKSUM SUBROUTINE

Returns with Z bit cleared if any error was found; otherwise Z = 1, X = 0 on return, and A is zero if the test passed. RAM locations $40-$43 are overwritten. (Enter at location $7A4.)

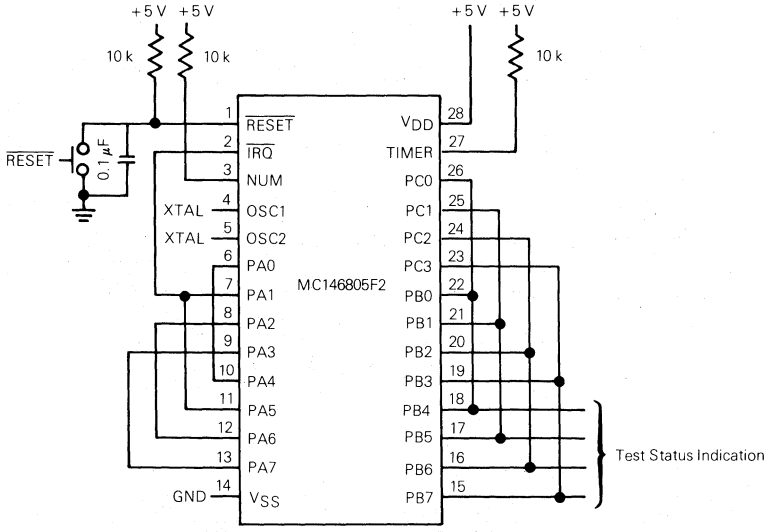
TIMER TEST SUBROUTINE

Return with Z bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask will not be set, so the caller must protect himself from interrupts if necessary.

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $7BE.)

FIGURE 10 – SELF-CHECK PINOUT CONFIGURATION



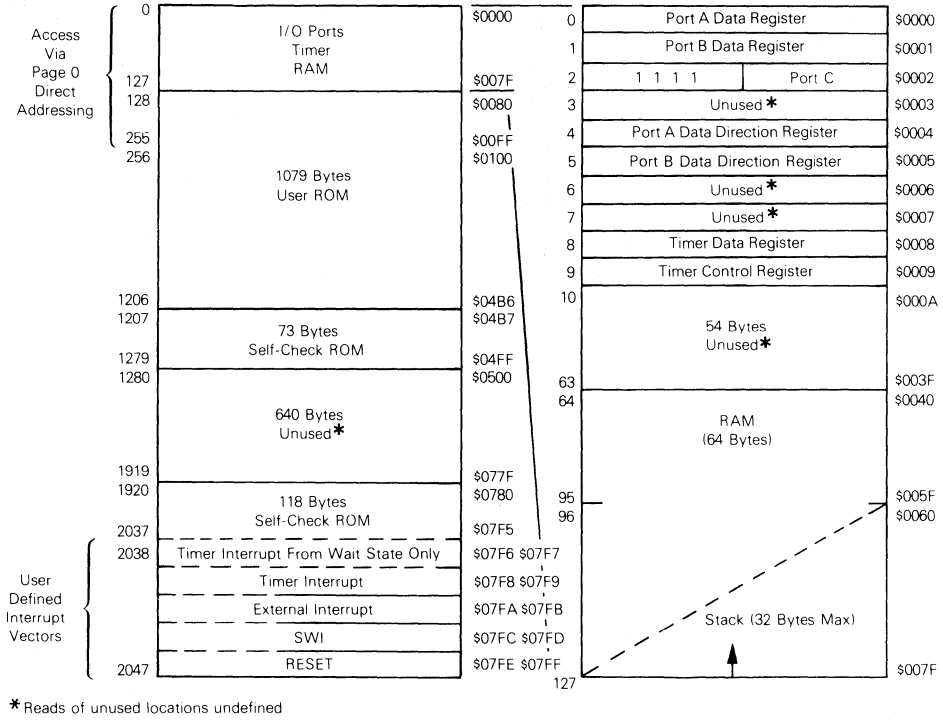
MEMORY

The MC146805F2 has a total address space of 2048 bytes of memory and I/O registers. The address space is shown in Figure 11.

The first 128 bytes of memory (first half of page zero) is comprised of the I/O port locations, timer locations, and 64 bytes of RAM. The next 1079 bytes comprise the user ROM. The 10 highest address bytes contain the reset and interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $7F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 32 bytes of RAM are available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are available for program data storage.

FIGURE 11 — ADDRESS MAP



*Reads of unused locations undefined

REGISTERS

The MC146805F2 contains five registers as shown in the programming model (Figure 12). The interrupt stacking order is shown in Figure 13.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands and results of the arithmetic calculations and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides the 8-bit operand which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is an 11-bit register containing the address of the next free location on the stack. When accessing memory, the six most-significant bits are appended to the five least-significant register bits to produce an address within the range of $7F to $60. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a "reset stack pointer" instruction, the stack pointer is set to its upper limit ($7F). Nested interrupts and/or subroutines may use up to 32 (decimal) locations beyond which the stack pointer "wraps around" and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five bytes.

FIGURE 12 — PROGRAMMING MODEL

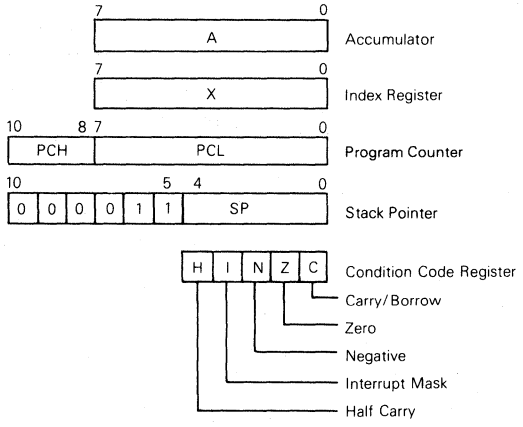
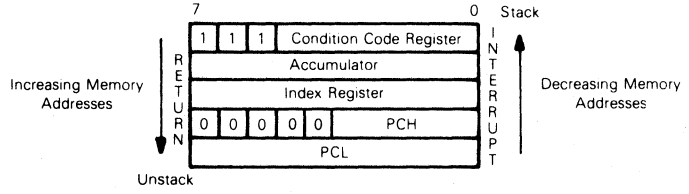


FIGURE 13 — STACKING ORDER



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BIT (H) — The H bit is set to a "1" when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed when the I bit is next cleared.

NEGATIVE (N) — Indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical "1").

ZERO (Z) — Indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) — Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The MC146805F2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{RL}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time of the first oscillator operation. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit (TCR7) is cleared to a "0".
- Timer control register interrupt mask bit (TCR6) is set to a "1".
- All data direction register bits are cleared to a "0". All ports are defined as inputs.
- Stack pointer is set to $7F.
- The internal address bus is forced to the reset vector ($7FE, $7FF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are reset.
- External interrupt latch is reset.

All other functions, such as other registers (including output ports), the timer, etc., are not cleared by the reset conditions.

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The MC146805F2 may be interrupted by one of three different methods, either one of two maskable interrupts (external input or timer) or a non-maskable software interrupt (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and return to normal processing. The stacking order is shown in Figure 13.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

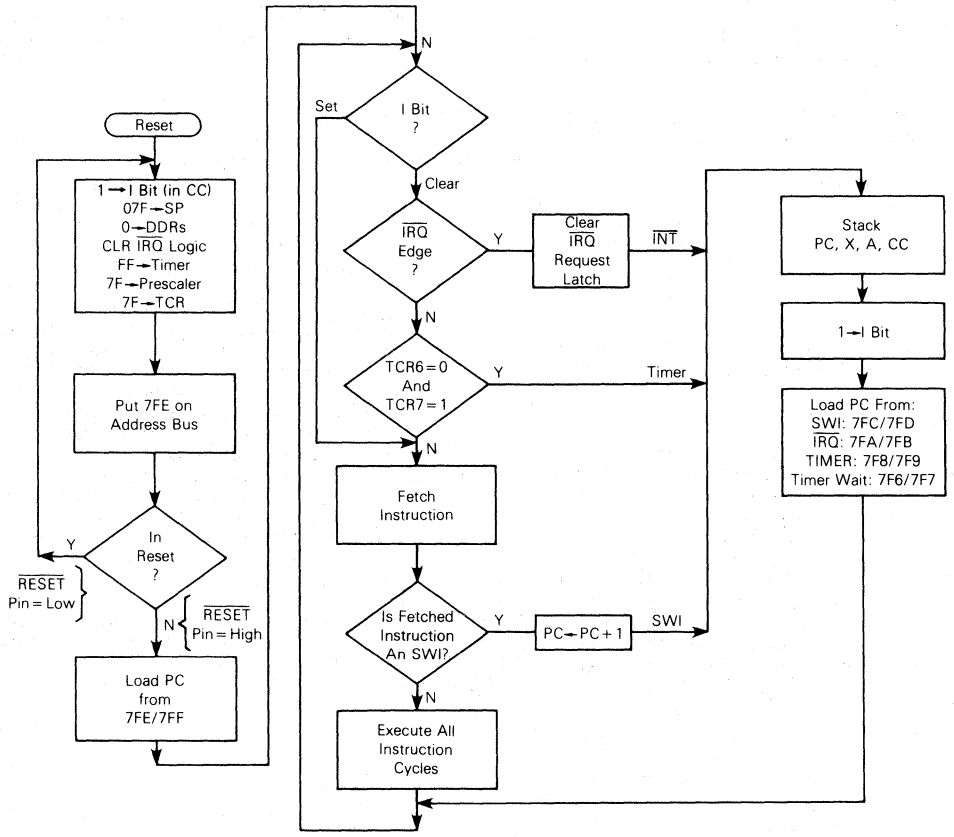
When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction. Refer to Figure 14 for the interrupt and instruction processing sequence.

TIMER INTERRUPT

Each time the timer decrements to zero (transitions from $01 to $00), the timer interrupt request bit (TCR7) is set. The processor is interrupted only if the timer mask bit (TCR6) and interrupt mask bit (I bit) are both cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This mask prevents further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $7F8 and $7F9 unless the processor is in a WAIT mode, in which case the contents of $7F6 and $7F7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

FIGURE 14 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART



EXTERNAL INTERRUPT

Either level- and edge-sensitive or edge-sensitive only inputs are available as mask options. If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{IRQ}$) is "low" or a negative edge has set the internal interrupt flip-flop, then the external interrupt occurs. The action of the external interrupt is identical to the timer except that the service routine address is specified by the contents of $7FA and $7FB. Figure 15 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 15. The second configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $7FC and $7FD.

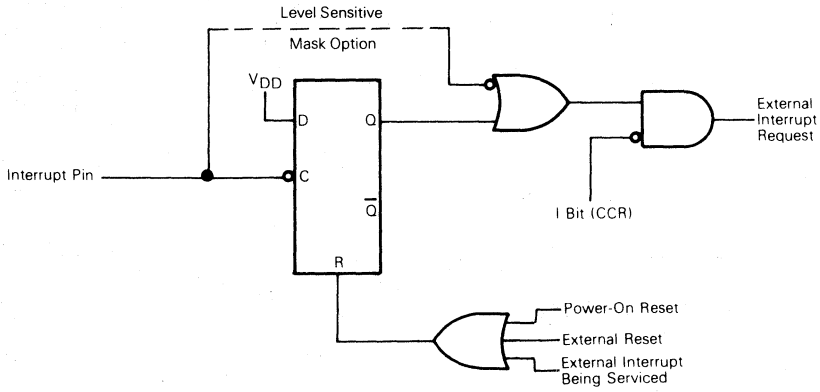
The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are $\overline{RESET}$, STOP, and WAIT.

$\overline{RESET}$ – The $\overline{RESET}$ input pin and the internal power-on reset function each cause the program to vector to an initialization program. This vector is specified by the contents

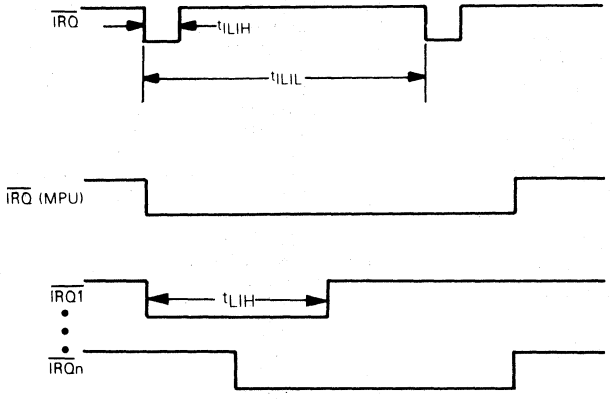
3

FIGURE 15 – EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Mask Optional Level Sensitive

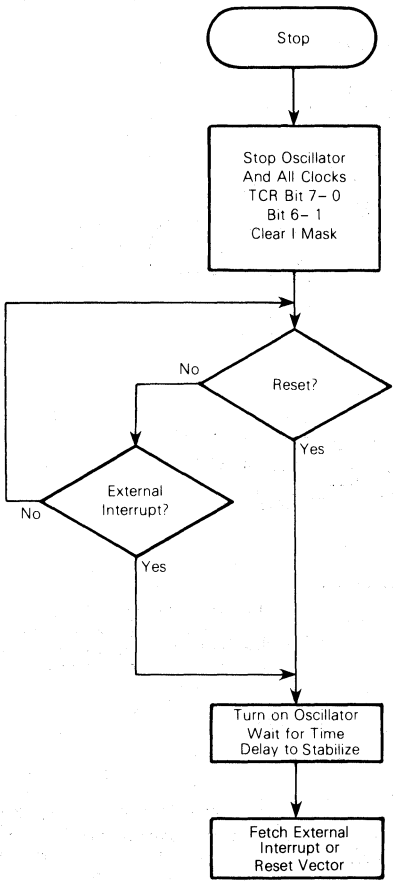
If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

of memory locations $7FE and $7FF. The interrupt mask of the condition code register is also set. See preceding section on Reset for details.

STOP — The STOP instruction places the MC146805F2 in its lowest power consumption mode. In the STOP function, the internal oscillator is turned off causing all internal processing and the timer to be halted; refer to Figure 16.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timing interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged. The processor can only be brought out of the STOP mode by an external $\overline{IRQ}$ or $\overline{RESET}$.

FIGURE 16 — STOP FUNCTION FLOWCHART



WAIT — The WAIT instruction places the MC146805F2 in a low-power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 17. Thus, all internal processing is halted, however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled by software prior to entering the WAIT mode to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MCU is no longer in the WAIT mode.

TIMER

The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 18 contains a block diagram of the timer. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register (TCR)) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack and then fetches the timer vector address from locations $7F8 and $7F9 (or $7F6 and $7F7 if in the WAIT mode) in order to begin servicing.

The counter continues to count after it reaches zero allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable, prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit remains set until cleared by the software. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

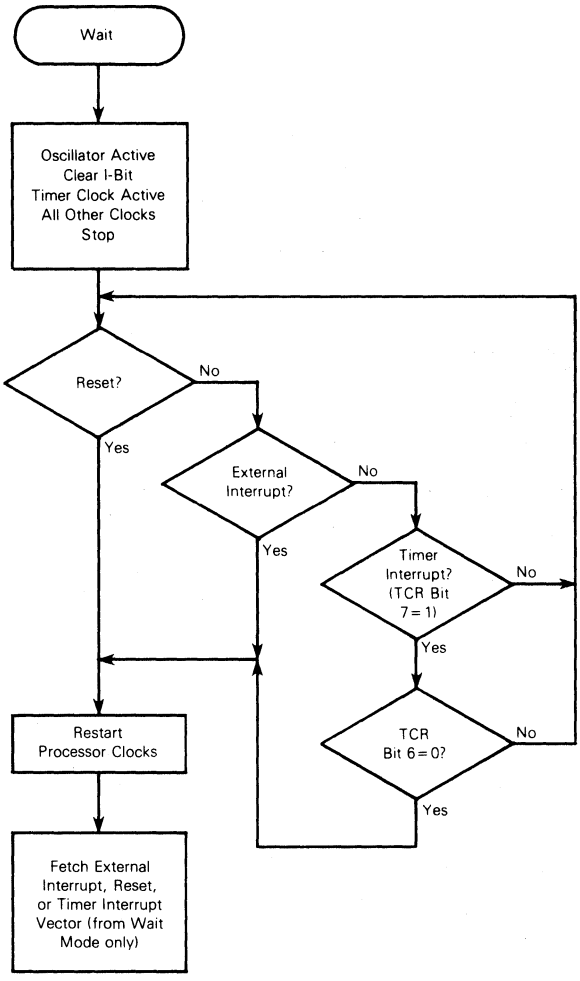
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output within the range of ÷1 to ÷128 which is used as the counter input. The processor cannot write into or read from the prescaler, however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode depending on the value written to the TCR4 and TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR5 and TCR4 are both programmed to a "0", the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for

FIGURE 17 — WAIT FUNCTION FLOWCHART



periodic interrupt generation as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR5=0 and TCR4=1, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±one internal clock and therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

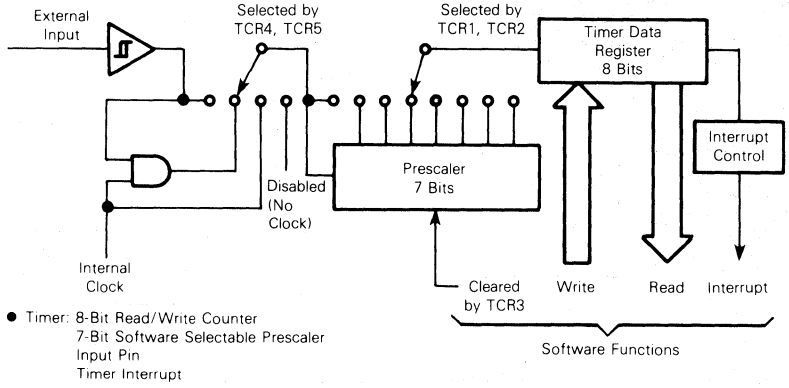
If TCR5=1 and TCR4=0, all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR5=1 and TCR4=1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 18 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction invalidate the contents of the counter.

FIGURE 18 — PROGRAMMABLE TIMER/COUNTER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register are clocked on the falling edge of the internal clocks or external input.
2. The timer data register counts down continuously.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are read/write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic "1".
1 — Set whenever the counter decrements to zero or under program control.
0 — Cleared on external $\overline{\text{RESET}}$, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic "1", it inhibits the timer interrupt to the processor.
1 — Set on external $\overline{\text{RESET}}$, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external timer pin or the internal clock. (Unaffected by $\overline{\text{RESET}}$.)
1 — Select external clock source.
0 — Select internal clock source.

TCR4 — External enable bit: control bit used to enable the external TIMER pin. (Unaffected by $\overline{\text{RESET}}$.)
1 — Enable external TIMER pin.
0 — Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal Clock to Timer
0	1	AND of Internal Clock and TIMER Pin to Timer
1	0	Inputs to Timer Disabled
1	1	TIMER Pin to Timer

TCR3 — Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates "0". (Unaffected by $\overline{\text{RESET}}$.)

TCR2, TCR1, TCR0 — Prescaler select bits: decoded to select one of eight outputs on the prescaler. (Unaffected by $\overline{\text{RESET}}$.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 basic instructions. 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 4.

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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and, if certain criteria are met, a branch is executed. This adds an offset between − 127 and + 128 to the current program counter. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

OPCODE MAP

Table 9 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 10.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions while the longest instructions (three bytes) permit tables throughout memory. Short and long absolute addressing is also included. Two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 10 shows the addressing modes for each instruction with the effects each instruction has on the condition code register. An opcode map is shown in Table 9.

The term "Effective Address" (EA) is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by," and a colon indicates "concatenation of two bytes." For additional details and graphical illustrations, refer to the M6805 Family User Manual.

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index registers or accumulator and no other arguments are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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 lowest 256 bytes in memory with a single two-byte instruction. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC + PC + 2$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$

$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

INDEXED, NO-OFFSET

In the indexed, no-offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register, therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in an n element table. All instructions are two bytes. The content of the index register

(X) is not changed. The content of (PC + 1) is an unsigned 8-bit integer. One-byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2

Address Bus High ← K; Address Bus Low ← X + (PC + 1)

where K = The carry from the addition of X + (PC + 1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three-byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset — 8 or 16 bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3

Address Bus High ← (PC + 1) + K;

Address Bus Low ← X + (PC + 2)

where K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;

otherwise, PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 128 addressable locations are thus accessed. The bit to be modified within that byte is specified with three bits of the opcode. The bit set and clear instructions occupy two bytes: one for the opcode (including the bit number) and the second for addressing the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2

Address Bus High ← 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit addressing, and relative addressing. The bit address and condition (set or clear) to be tested is part of the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)

Address Bus High ← 0; Address Bus Low ← (PC + 1)

EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;

otherwise, PC ← PC + 3

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	—	—	—	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	—	—	—	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	—	—	—

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9– INSTRUCTION SET OPCODE MAP

Hi / Low	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						Hi / Low
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

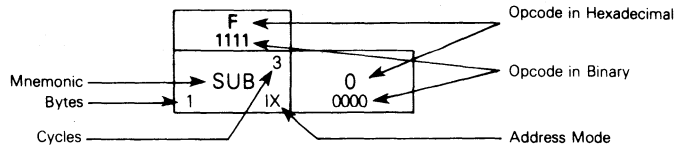


TABLE 10 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow

Λ Test and Set if True, Cleared Otherwise.
● Not Affected
? Load CC Register From Stack
0 Cleared
1 Set

ORDERING INFORMATION

The following information is required when ordering a custom MCU. This information may be transmitted to Motorola in the following media:

EPROM MCM2716

MDOS disk file

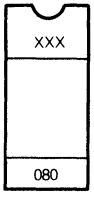
To initiate a ROM pattern for the MCU it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

EPROMs

The MCM2716 type EPROM, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The customer program should begin at address $080 (the address at which customer ROM begins on the MC146805F2) so that the EPROM maps directly into the MC146805F2. If the customer program starts at any other address, please mark the EPROM accordingly. See Figure 19 for recommended marking procedure.

After the EPROM is marked, it should be placed in a conductive IC carrier and securely packed. Do not use styrofoam.

FIGURE 19 — EPROM MARKING



XXX = Customer I.D.

VERIFICATION MEDIA

All original pattern media (EPROM 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 EPROM (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATON UNITS

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 5 volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS-compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The floppies are not to be returned by Motorola as they are used for archival storage. The minimum MDOS system files as well as the absolute binary object file (file name .LO type of file) from the M6805 cross assembler 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 the following files: filename, LX (EXORciser loadable format) and filename .SA (ASCII source code). These files will be kept confidential and 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 representatives.

MDOS is Motorola's Disk Operating System available on development systems such as EXORciser, EXORsets, etc.

OPTION LIST

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Internal Divide
- ☐ ÷4
- ☐ ÷2

Interrupt
- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Customer Name _____

Address _____

City _____ State _____ Zip _____

Phone (_____) _____ Extension _____

Contact Ms/Mr _____

Customer Part Number _____

Pattern Media

- ☐ 2716 EPROM
- ☐ MDOS Disk File
- ☐ Silent 700 Cassette
- ☐ Card Deck
- ☐ Tape of Card Deck
- ☐ (Note 1) _____

NOTE 1. Other media require prior factory approval.

Signature _____

Title _____

Silent 700 Cassette is a trademark of Texas Instruments Incorporated

MOTOROLA

MC146805G2

Advance Information

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT MICROCOMPUTER

8-BIT MICROCOMPUTER UNIT

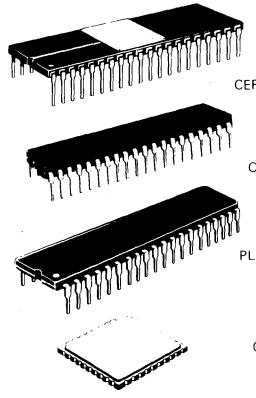
The MC146805G2 Microcomputer Unit (MCU) belongs to the M146805 CMOS Family of Microcomputers. This 8-bit MCU contains on-chip oscillator, CPU, RAM, ROM, I/O, and TIMER. The fully static design allows operation at frequencies down to dc, further reducing its already low-power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the MC146805G2 MCU.

HARDWARE FEATURES

- Typical Full Speed Operating Power of 12 mW at 5 V
- Typical WAIT Mode Power of 4 mW
- Typical STOP Mode Power of 5 μW
- Fully Static Operation
- 112 Bytes of On-Chip RAM
- 2106 Bytes of On-Chip ROM
- 32 Bidirectional I/O Lines
- High Current Drive
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Self-Check Mode
- Master Reset and Power-On Reset
- Single 3 to 6 Volt Supply
- On-Chip Oscillator with RC or Crystal Mask Options
- 40-Pin Dual-In-Line Package
- Chip Carrier Also Available

SOFTWARE FEATURES

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Most Self-Check Routines User Callable
- Two Power Saving Standby Modes



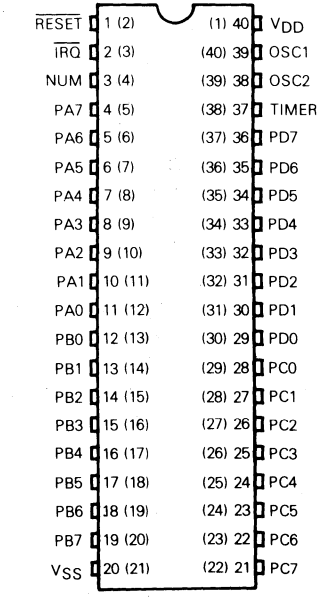
L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins.

GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic	1.0	0°C to 70°C	MC146805G2L
L Suffix	1.0	−40°C to 85°C	MC146805G2CL
Cerdip	1.0	0°C to 70°C	MC146805G2S
S Suffix	1.0	−40°C to 85°C	MC146805G2CS
Plastic	1.0	0°C to 70°C	MC146805G2P
P Suffix	1.0	−40°C to 85°C	MC146805G2CP
Leadless Chip Carrier			
Z Suffix	1.0	0°C to 70°C	MC 146805G2Z
	1.0	−40°C to 85°C	MC146805G2CZ

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

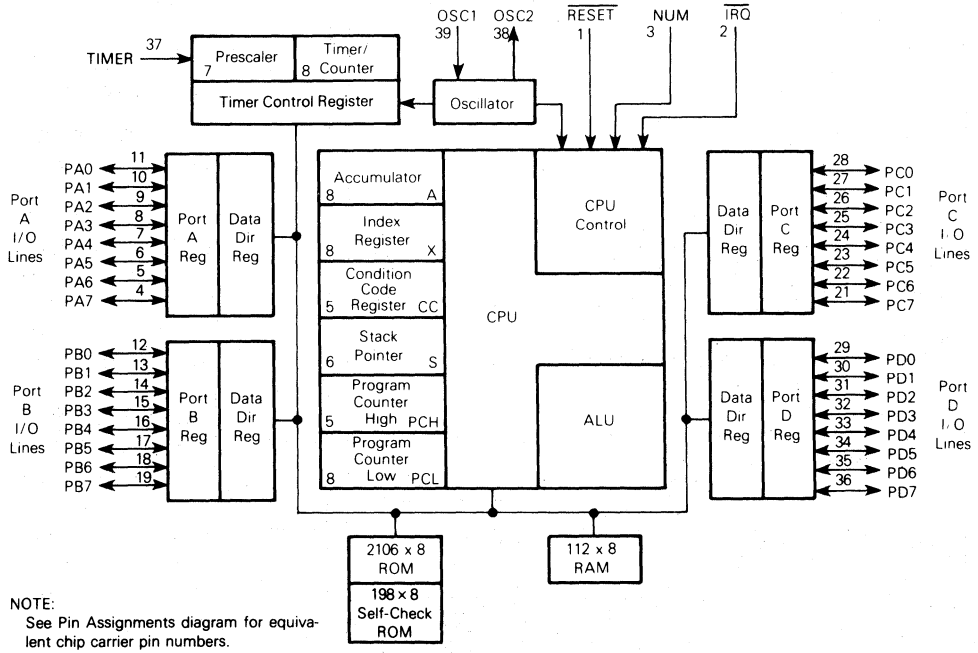
Ratings	Symbol	Value	Unit
Supply Voltage	V_D	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805G2 MC146805G2C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain Total (PD4-PD7 only)	I_{OH}	40	mA

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic Chip Carrier	θ_{JA}	 100 60 50 100	°C/W

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 $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and NUM are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

FIGURE 1 — MC146805G2 CMOS MICROCOMPUTER



NOTE:
See Pin Assignments diagram for equivalent chip carrier pin numbers.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V V
Output High Voltage (I_{Load} = −100 μA) PB0-PB7, PC0-PC7	V_{OH}	2.4	–	V
(I_{Load} = −2 mA) PA0-PA7, PD0-PD3	V_{OH}	2.4	–	V
(I_{Load} = −8 mA) PD4-PD7	V_{OH}	2.4	–	V
Output Low Voltage (I_{Load} = 800 μA) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	$V_{DD}-2.0$	V_{DD}	V
$\overline{\text{TIMER}}$, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$V_{DD}-0.8$	V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.8	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1 μs) RUN (V_{IL} = 0.2 V, V_{IH} = V_{DD} − 0.2 V)	I_{DD}	–	4	mA
$\overline{\text{WAIT}}$ (See Note)	I_{DD}	–	1.5	mA
$\overline{\text{STOP}}$ (See Note)	I_{DD}	–	150	μA
I/O Ports Input Leakage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	±10	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	I_{in}	–	±1	μA
Capacitance Ports	C_{out}	–	12	pF
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	C_{in}	–	8	pF

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.0 Vdc, V_{SS} = 0 Vdc, T_A = 0° to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \leq 1.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V V
Output High Voltage (I_{Load} = −50 μA) PB0-PB7, PC0-PC7	V_{OH}	1.4	–	V
(I_{Load} = −0.5 mA) PA0-PA7, PD0-PD3	V_{OH}	1.4	–	V
(I_{Load} = −2 mA) PD4-PD7	V_{OH}	1.4	–	V
Output Low Voltage (I_{Load} = 300 μA) All Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.3	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{IH}	2.7	V_{DD}	V
$\overline{\text{TIMER}}$, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	2.7	V_{DD}	V
Input Low Voltage All Inputs	V_{IL}	V_{SS}	0.3	V
Total Supply Current (no dc Loads, t_{cyc} = 5 μs) RUN (V_{IL} = 0.1 V, V_{IH} = V_{DD} − 0.1 V)	I_{DD}	–	0.5	mA
$\overline{\text{WAIT}}$ (See Note)	I_{DD}	–	200	μA
$\overline{\text{STOP}}$ (See Note)	I_{DD}	–	100	μA
I/O Ports Input Leakage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	±5	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	I_{in}	–	±1	μA
Capacitance Ports	C_{out}	–	12	pF
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	C_{in}	–	8	pF

NOTE: Test conditions for I_{DD} are as follows:
All ports programmed as inputs
V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)
V_{IH} = V_{DD} − 0.2 V for $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER
OSC1 input is a squarewave from 0.2 V to V_{DD} − 0.2 V
OSC2 output load = 20 pF (wait I_{DD} is affected linearly by the OSC2 capacitance).

TABLE 1 — CONTROL TIMING
(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0, T_A = 0° to 70°C, f_{osc} = 4 MHz)

Characteristics	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (Figure 11)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (Figure 12)	t_{ILCH}	–	100	ms
Timer Pulse Width (Figure 10)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (Figure 11)	t_{RL}	1.5	–	t_{cyc}
Timer Period (Figure 10)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width Low (Figure 21)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (Figure 21)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	100	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal	f_{osc}	–	4.0	MHz
External Clock	f_{osc}	DC	4.0	MHz

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

3

FIGURE 2 — EQUIVALENT TEST LOAD

Port	R_1	R_2
B and C	24.3 kΩ	4.32 kΩ
A, PD0-PD3	1.21 kΩ	3.1 kΩ
PD4-PD7	300 Ω	1.64 kΩ

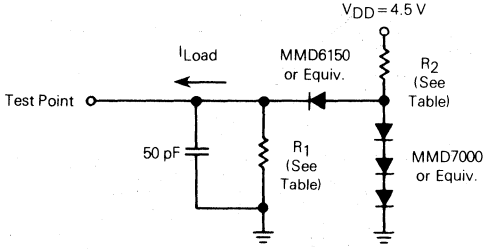


FIGURE 3 — TYPICAL OPERATING CURRENT vs INTERNAL FREQUENCY

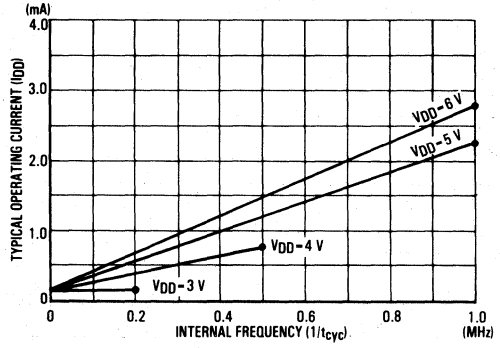


FIGURE 4 — MAXIMUM I_{DD} vs FREQUENCY

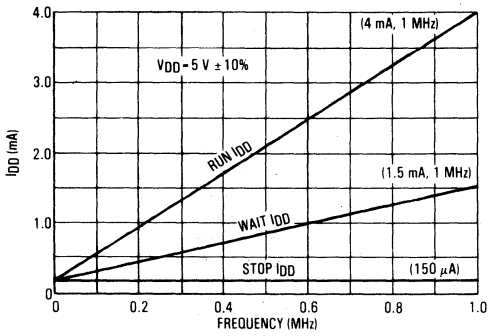


FIGURE 5 — MAXIMUM I_{DD} vs FREQUENCY

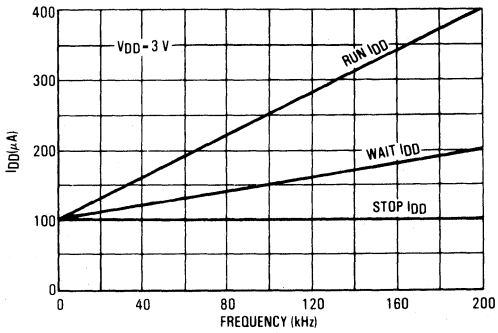


FIGURE 6 — MINIMUM I_{OH}, PORT D PINS 33-36

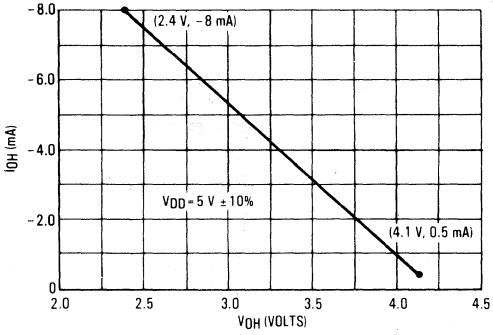


FIGURE 7 — MINIMUM I_{OH}, PORT C

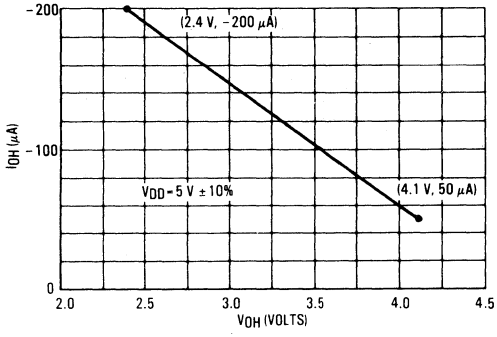


FIGURE 8 — MINIMUM I_{OH}, PORT A AND B

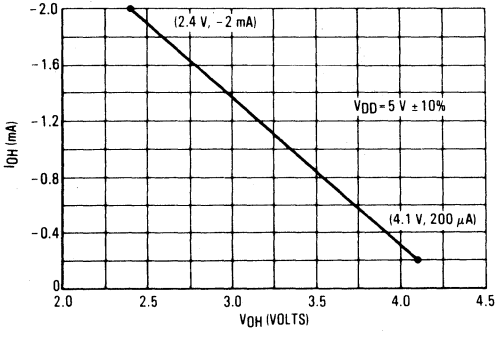


FIGURE 9 — MINIMUM I_{OL}, ALL PORTS

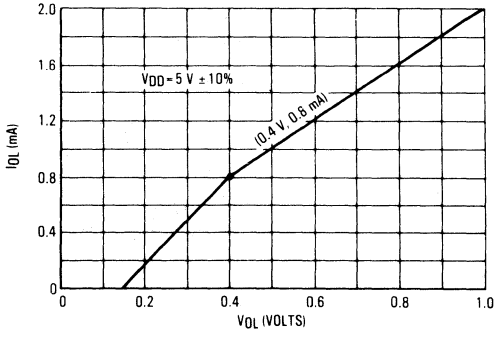


FIGURE 10 — TIMER RELATIONSHIPS

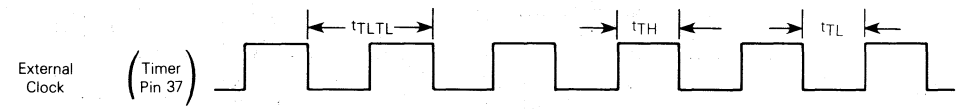
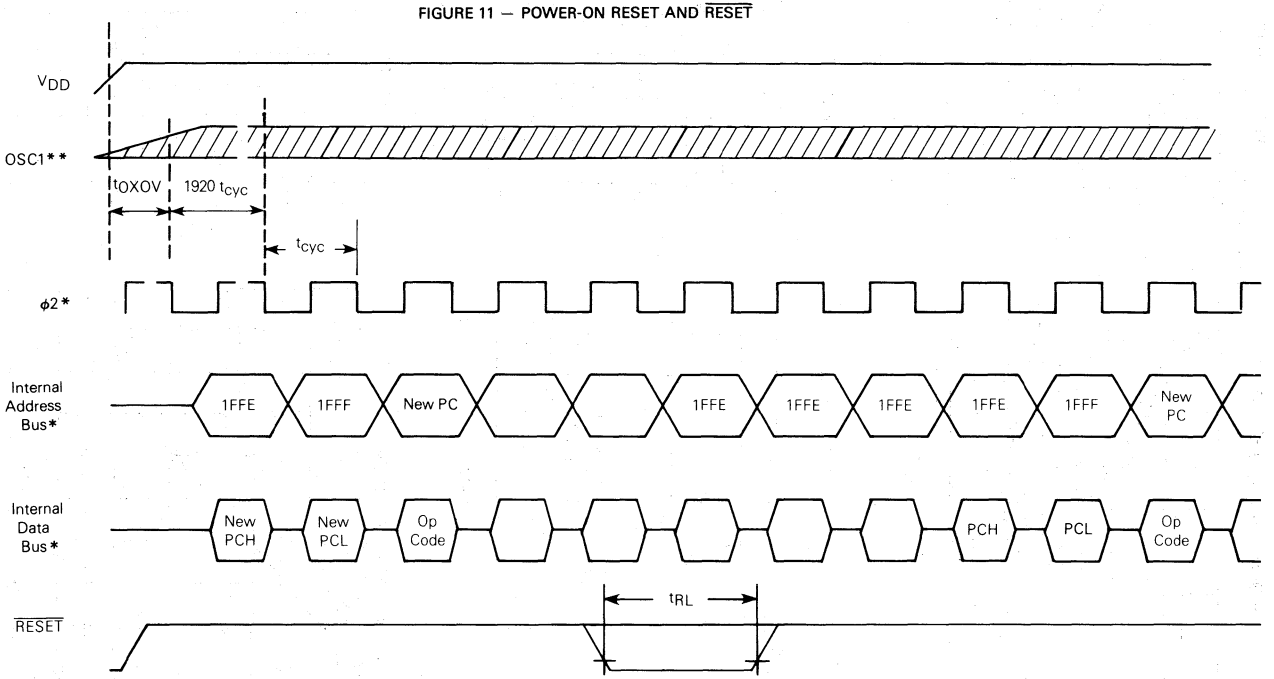
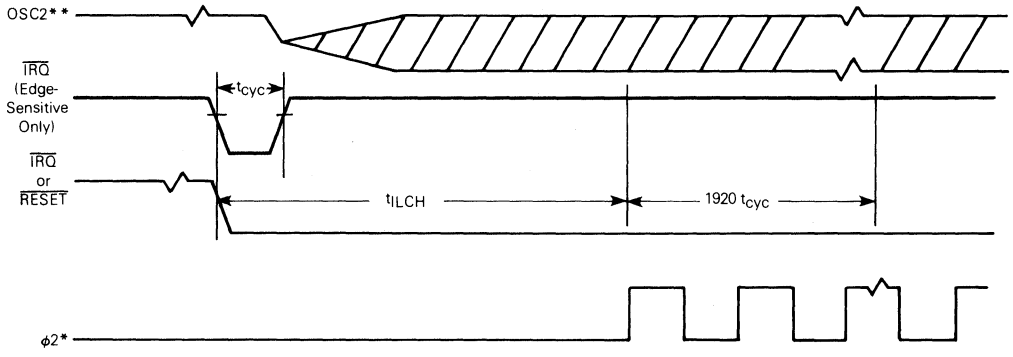


FIGURE 11 — POWER-ON RESET AND $\overline{\text{RESET}}$



*Internal timing signal and bus information not available externally.
**OSC1 line is not meant to represent frequency. It is only used to represent time.

FIGURE 12 — STOP RECOVERY AND POWER-ON RESET



*Internal timing signals not available externally.
**Represents the internal gating of the OSC1 input pin.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

IRQ (MASKABLE INTERRUPT REQUEST)

IRQ is mask option selectable with the choice of interrupt sensitivity being both level-sensitive, and negative edge-sensitive or negative edge-sensitive only. The MCU completes the current instruction before it responds to the request. If IRQ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

If the mask option is selected to include level sensitivity, then the IRQ input requires an external resistor to V_{DD} for "wire-OR" operation. See INTERRUPTS for more detail.

RESET

The RESET input is not required for start-up but can be used to reset the MCU's internal state and provide an orderly software start-up procedure. Refer to RESETS for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to TIMER for additional information about the timer circuitry.

NUM — NON-USER MODE

This pin is intended for use in self-check only. In user applications, connect this pin to ground through a 10 kΩ resistor.

OSC1, OSC2

The MC146805G2 can be configured to accept either a crystal input or an RC network to control the internal oscillator. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the internal oscillator output frequency (f_{OSC}). Both of these options are mask selectable.

RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 13(d). The relationship between R and f_{OSC} is shown in Figure 14.

CRYSTAL — The circuit shown in Figure 13(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in the electrical characteristics table. Using an external CMOS oscillator is suggested when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. Crystal frequency limits are also affected by V_{DD}. Refer to Control Timing Characteristics for limits. See Table 1.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 13(c). An external clock should be used with the crystal oscillator mask option only. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

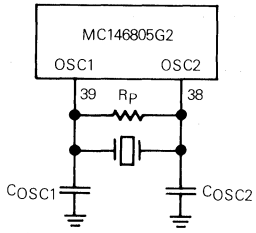
PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

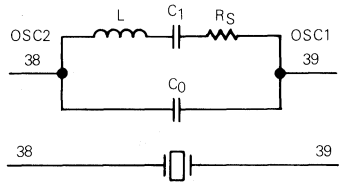
FIGURE 13 — OSCILLATOR CONNECTIONS

	1 MHz	4 MHz	Units
R_{SMAX}	400	75	Ω
C_0	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P	10	10	MΩ
Q	30	40	K

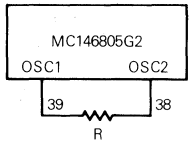
(a) Crystal Parameters



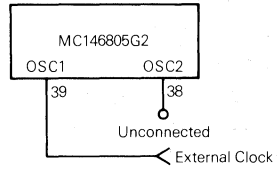
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit

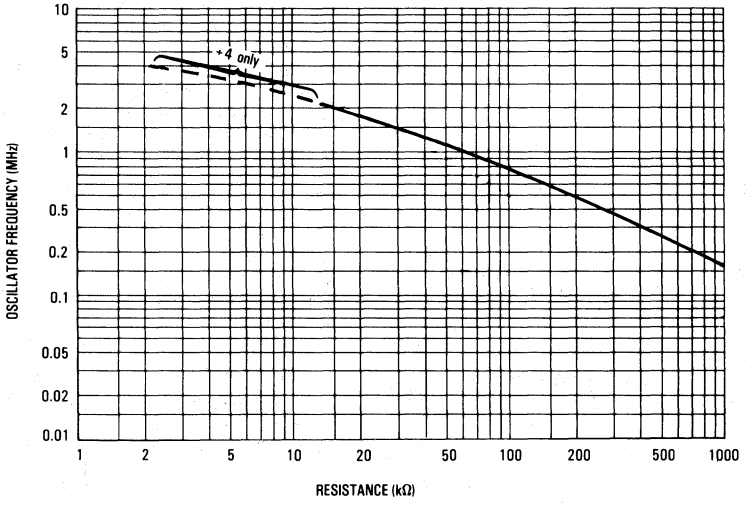


(d) RC Oscillator Connection



(e) External Clock Source Connections

FIGURE 14 — TYPICAL FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

PC0-PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

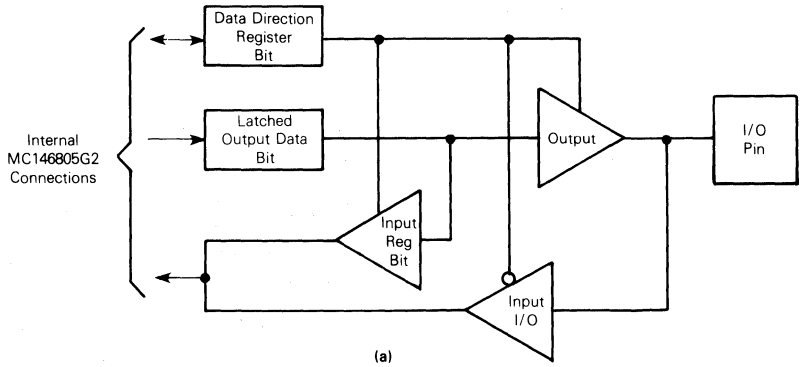
PD0-PD7

These eight lines comprise Port D. PD4-PD7 also are capable of driving LEDs directly. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a description of I/O programming.

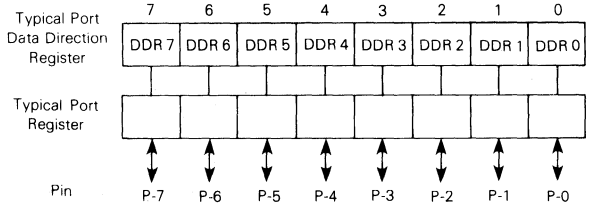
INPUT/OUTPUT PROGRAMMING

Any port pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0. At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 15 and Table 2.

FIGURE 15 — TYPICAL PORT I/O CIRCUITRY



(a)



(b)

TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

SELF-CHECK

The MC146805G2 self-check is performed using the circuit in Figure 16. Self-check is initiated by connecting NUM and TIMER pins to a logic 1 then executing a reset. After reset, five subroutines are called that execute the following tests:

I/O — Functionally exercise ports A, B, C, D

RAM — Walking bit test

ROM — Exclusive OR with odd 1s parity result

Timer — Functionally exercise timer

Interrupts — Functionally exercise external and timer interrupts

Self-check results are shown in Table 3. The following subroutines are available to user programs and do not require any external hardware.

RAM SELF-CHECK SUBROUTINES

Returns with the Z bit clear if any error is detected; otherwise the Z bit is set.

The RAM test must be called with the stack pointer at $007F. When run, the test checks every RAM cell except for $007F and $007E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top 2 are modified. (Enter at location $1F80.)

ROM CHECKSUM SUBROUTINE

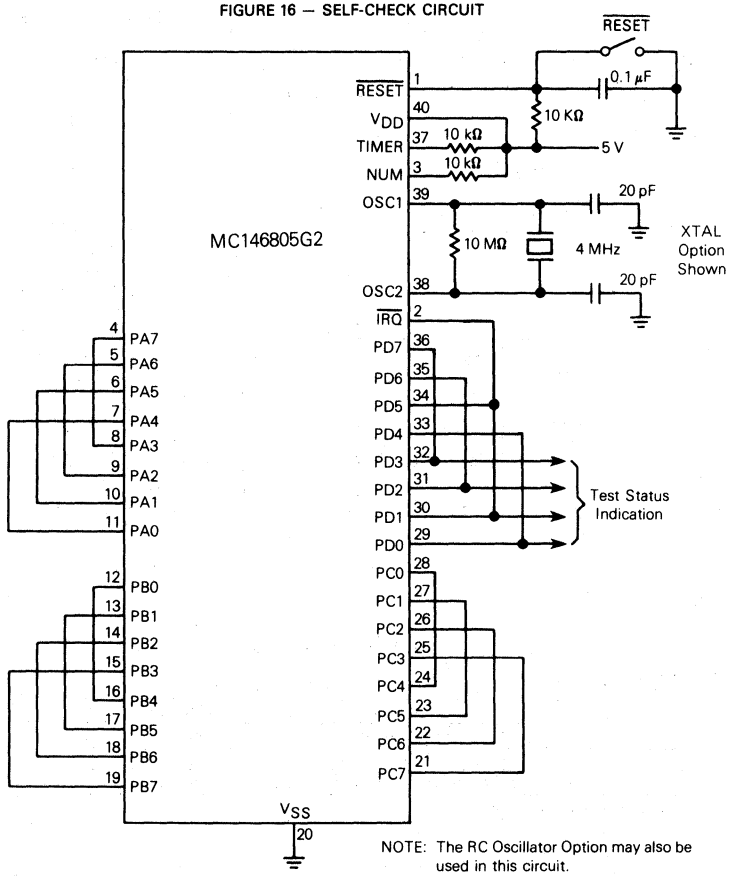
Returns with Z bit cleared if any error was found, otherwise Z = 1. X = 0 on return, and A is zero if the test passed. RAM locations $0040-$0043 are overwritten. (Enter at location $1F9B.)

TIMER TEST SUBROUTINE

Return with Z bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask not set so the caller must protect himself from interrupts if necessary.

FIGURE 16 — SELF-CHECK CIRCUIT

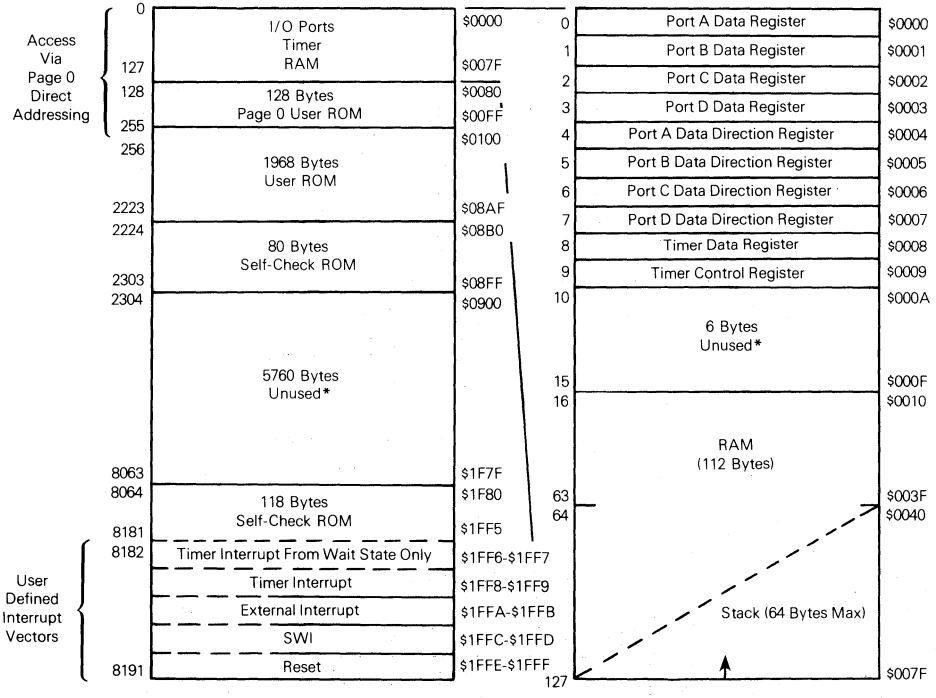


NOTE: The RC Oscillator Option may also be used in this circuit.

TABLE 3 — SELF-CHECK RESULTS

PD3	PD2	PD1	PD0	Remarks
1	0	1	0	Bad I/O
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
Cycling				Good Part
All Others				Bad Part

FIGURE 17 — ADDRESS MAP



*Reads of unused locations undefined.

A and X register contents are lost; this routine counts how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all. (Enter at location $1FB5.)

MEMORY

The MC146805G2 has a total address space of 8192 bytes of memory and I/O registers. The address space is shown in Figure 17.

The first 128 bytes of memory (first half of page zero) are comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The next 2096 bytes (including the 128 bytes of the second half of page zero) comprise the user ROM. The 10 highest address bytes contain the reset and the interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power-up, the stack pointer is set to $007F and it is decremented as data is pushed on the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

REGISTERS

The MC146805G2 contains five registers, as shown in the programming model in Figure 18. The interrupt stacking order is shown in Figure 19.

FIGURE 18 — PROGRAMMING MODEL

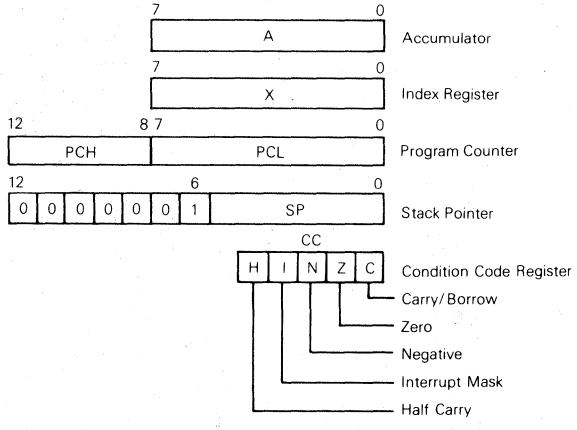
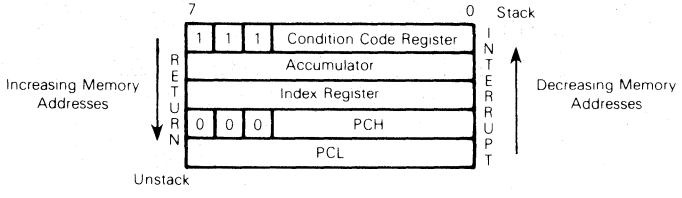


FIGURE 19 — STACKING ORDER



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

ACCUMULATOR (A)

This accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently configured to 0000001. These seven bits are appended to the six least significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($007F). Nested interrupts and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer wraps around and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BITS (H) — The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed when the I bit is next cleared.

NEGATIVE (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logical 1).

ZERO (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulations is 0.

CARRY/BORROW (C) — When set, this bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The MC146805G2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 11.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{CYC}. The $\overline{\text{RESET}}$ pin is provided with a Schmitt Trigger input (internally) to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{CYC} delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 t_{CYC} time out, the processor remains in the reset condition.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit TCR7 is cleared to a "0".
- Timer control register interrupt mask bit TCR6 is set to a "1".
- All data direction register bits are cleared to logical zeros. All ports are defined as inputs.
- Stack pointer is preset to $007F.
- The internal address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a "1".
- STOP and WAIT latches are cleared.
- External interrupt latch is cleared.

All other functions, such as other registers (including output ports), the timer, etc. are not cleared by the reset conditions.

INTERRUPTS

The MC146805G2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a nonmaskable software interrupt (SWI). Systems often require that normal processing be interrupted so that some external event may be serviced.

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 19.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is complete.

NOTE

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked (I bit

clear), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction and as such takes precedence over hardware interrupts only if the I bit is set (hardware interrupts masked). Refer to Figure 20 for the interrupt and instruction processing sequence.

Table 4 shows the execution priority of the $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$ and timer interrupts, and the software interrupt, SWI. Two conditions are shown, one with the I bit set and the other with I bit clear; however, in either case $\overline{\text{RESET}}$ has the highest priority of execution. If the I bit is set as per Table 4(a), the second highest priority is assigned to SWI. This is illustrated in Figure 20 which shows that the $\overline{\text{IRQ}}$ or Timer interrupts are not executed when the I bit is set and the next instruction (including SWI) is fetched. If the I bit is cleared as per Table 4(b), the priorities change in that the next instruction (including SWI) is not fetched until after the $\overline{\text{IRQ}}$ and Timer interrupts have been recognized (and serviced). Also, when the I bit is clear, if both $\overline{\text{IRQ}}$ and Timer interrupts are pending, the $\overline{\text{IRQ}}$ interrupt is always serviced before the Timer interrupt.

FIGURE 20 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

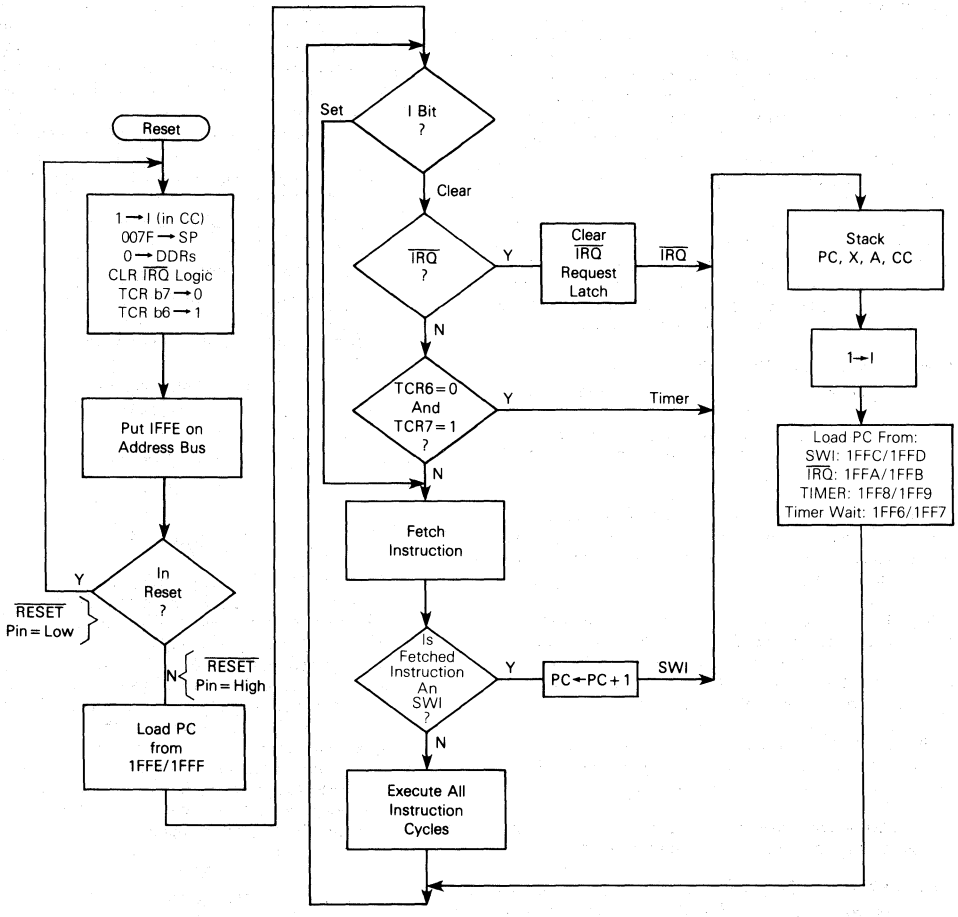


TABLE 4 — INTERRUPT/INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
SWI	2	$1FFC-$1FFD

NOTE: $\overline{\text{IRQ}}$ and Timer Interrupts are not executed when the I bit is set; therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
$\overline{\text{IRQ}}$	2	$1FFA-$1FFB
Timer	3	$1FF8-$1FF9
		$1FF6-$1FF7*
SWI	4	$1FFC-$1FFD

*The Timer vector address from the WAIT mode is $1FF6-$1FF7.

NOTE

Processing is such that at the end of the current instruction execution, the I bit is tested and if set the next instruction (including SWI) is fetched. If the I bit is cleared, the hardware interrupt latches are tested, and if no hardware interrupt is pending, the program falls through and the next instruction is fetched.

TIMER INTERRUPT

If the timer interrupt mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode in which case the contents of $1FF6 and $1FF7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

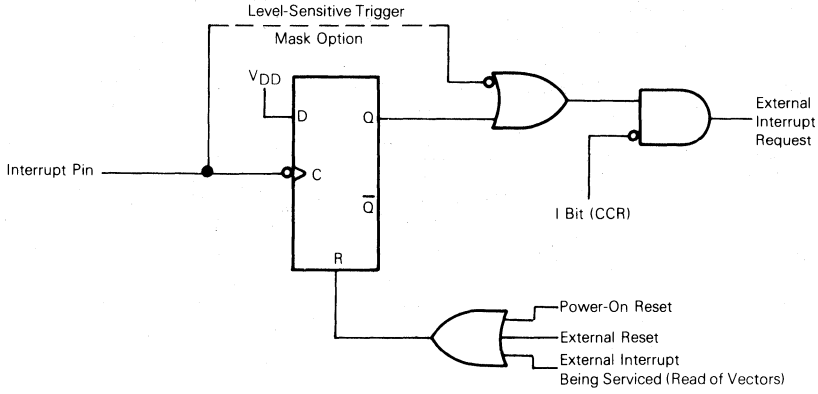
If the interrupt mask bit of the condition code register is cleared and the external interrupt pin ($\overline{\text{IRQ}}$) is low, then the external interrupt occurs. The action of the external interrupt is identical to the timer interrupt with the exception that the service routine address is specified by the contents of $1FFA and $1FFB. Either a level- and edge-sensitive trigger (or edge-sensitive only) are available as mask options. Figure 21 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first method is single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 21. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

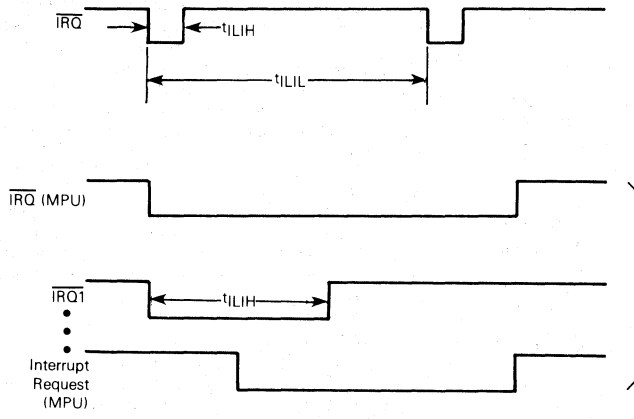
The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 20 for interrupt and instruction processing flowchart.

FIGURE 21 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Normally used with Wire-ORed Connection

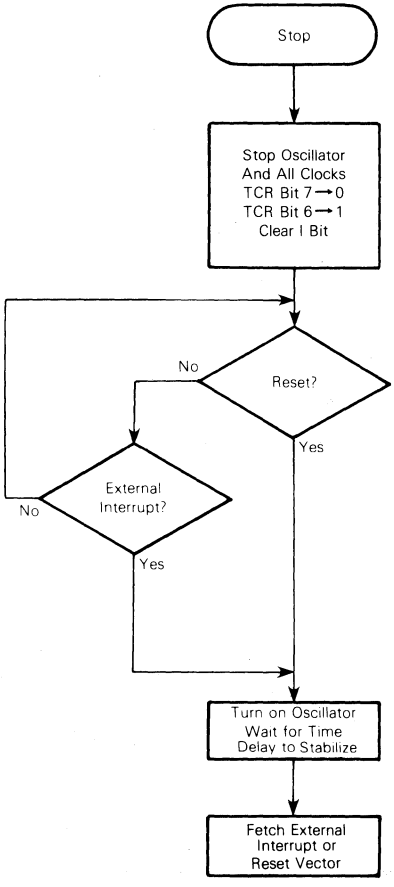
LOW-POWER MODES

STOP

The STOP instruction places the MC146805G2 in its lowest power consumption mode. In the STOP mode the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 22.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

FIGURE 22 — STOP FUNCTION FLOWCHART



WAIT

The WAIT instruction places the MC146805G2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 23. Thus, all internal processing is halted; however, the timer continues to count normally.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

3

TIMER

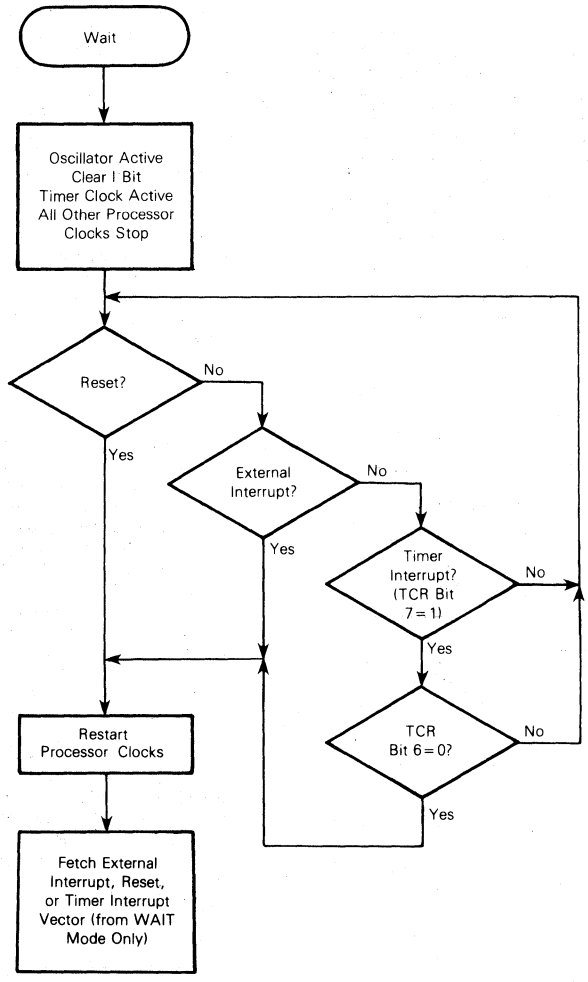
The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 24 contains a block diagram of the timer. The counter may be loaded under program control and is decremented towards zero by the clock input (prescaler output). When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register TCR) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 (or $1FF6 and $1FF7 if in the WAIT mode) in order to begin servicing.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit (TCR7) remains set until cleared by the software. If the timer interrupt request bit (TCR7) is cleared before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals a logic one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode, depending on the value written to the TCR4 and TCR5 control register bits. Refer to TIMER CONTROL REGISTER.

FIGURE 23 — WAIT FUNCTION FLOWCHART



TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a zero, the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ±1 clock; therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

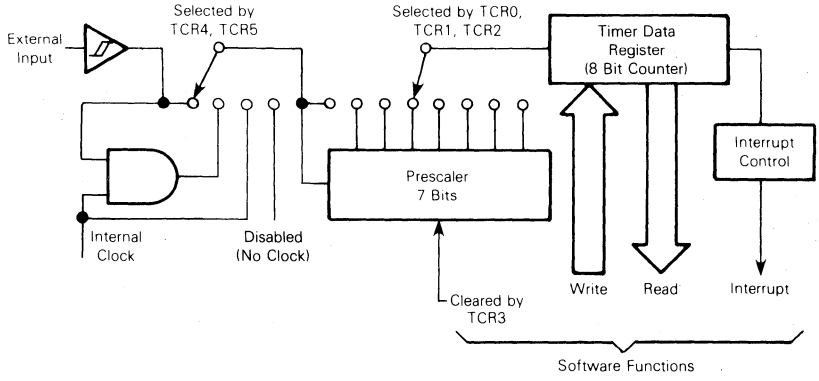
If TCR4 = 0 and TCR5 = 1, then all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR4 = 1 and TCR5 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 24 shows a block diagram of the timer subsystem. Power-on Reset and the STOP instruction cause the counter to be set to $FO.

FIGURE 24 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register (8-bit counter) are clocked on the falling edge of the internal clock or external input.
2. The timer data register counts down continuously.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are Read/Write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic one.
1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic one it inhibits the timer interrupt to the processor.
1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock. (Unaffected by reset.)
1 — Select external clock source.
0 — Select internal clock source (period = t_{cyc}).

TCR4 — External enable bit: control bit used to enable the external TIMER pin. (Unaffected by reset.)
1 — Enable external TIMER pin.
0 — Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal clock to timer
0	1	AND of internal clock and TIMER pin to timer
1	0	Inputs to timer disabled
1	1	TIMER pin to timer

TCR3 — Timer prescaler reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0". (Unaffected by reset.)

TCR2, TCR1, TCR0 — Prescaler select bits: decoded to select one of eight outputs of the prescaler. (Unaffected by reset.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MCU has a set of 61 basic instructions. 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 5.

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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 6.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between −127 and +128 to the current program counter. Refer to Table 7.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 8.

NOTE

The MCU is actually capable of operating on the bit set and bit clear instructions anywhere in the first 256 bytes; however, since only ROM resides in the upper 128 bytes the bit set/clear instructions have no effect on the upper 128 bytes.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 9.

OPCODE MAP

Table 10 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 11.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit tables throughout memory. Short absolute (direct) and long absolute addressing are also included. Table 11 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 10.

The term "Effective Address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the M6805 Family User's Manual.

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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 lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient address mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

EA = X; PC ← PC + 1
Address Bus High ← 0; Address Bus Low ← X

INDEXED, 8-BIT OFFSET

Here, the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the m-th element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2)
Address Bus High ← K; Address Bus Low ← X + (PC + 1)
Where: K = The carry from the addition of X + (PC + 1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8 or 16-bit. The content of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3
Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)
Where: K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise, EA = PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified within the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2
Address Bus High ← 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit set or bit clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location immediately following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or clear in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC ← PC + 3

TABLE 5 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 6 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 7 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 8 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	—	—	—

TABLE 9 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 10 — M146805 CMOS FAMILY INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

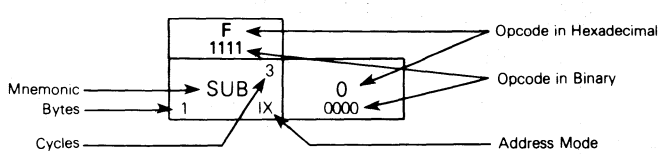


TABLE 11 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
ROR	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- Λ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

ORDERING INFORMATION

The following information is required when ordering a custom MCU. This information may be transmitted to Motorola in the following media:

EPROM(s) MCM2716s or MCM2532s

MDOS disk file

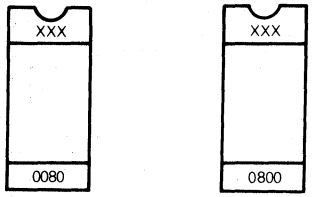
To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation. The EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure A-1 illustrates the recommended marking procedure for two MCM2716 EPROMs.

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE A-1 — EPROM MARKING



xxx = Customer ID

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 EPROM (supplied by the customer) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATION UNITS

The 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 5 volts. These RVUs are included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

FLEXIBLE DISKS

The disk media submitted must be single-sided, single-density, 8-inch, MDOS compatible floppies. The customer must write the binary file name and company name on the disk with a felt-tip pen. The floppies are not to be returned by Motorola as they are used for archival storage. The minimum MDOS system files as well as the absolute binary object file (filename.LO type of file) from the M6805 cross assembler 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 the following files: filename.LX (EXORciser loadable format) and filename.SA (ASCII Source Code). These files will of course be kept confidential and are 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 representatives.

MDOS is Motorola's Disk Operating System available on development systems such as EXORciser, EXORset, etc.

Option List

Select the options for your MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Internal Divide
- ☐ ÷4
- ☐ ÷2

Interrupt Trigger
- ☐ Edge-Sensitive Only
- ☐ Level-Sensitive and Edge-Sensitive

Customer Name _____

Address _____

City _____ State _____ Zip _____

Phone (_____) _____ Extension _____

Contact Ms/Mr _____

Customer Part Number _____

Pattern Media

2708 EPROM

2716 EPROM

MDOS Disk File

Silent 700 Cassette

Card Deck

Tape of Card Deck

(Note 2) _____

Notes: (2) Other media require prior factory approval.

Signature _____

Title _____

Silent 700 Cassette is a trademark of Texas Instruments Incorporated

MOTOROLA

MC146805H2

Advance Information

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT
MICROCOMPUTER

3

8-BIT MICROCOMPUTER UNIT

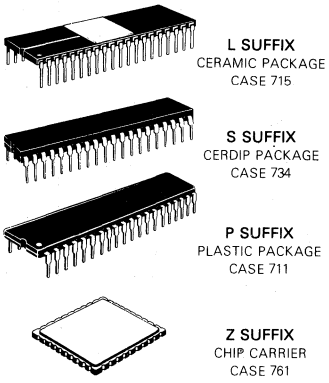
The MC146805H2 Microcomputer Unit (MCU) belongs to the M146805 CMOS Family of low-cost, single-chip microcomputers. This 8-bit MCU contains an on-chip oscillator, CPU, RAM, I/O, and a timer. The fully static design allows operation at two software selectable frequencies, further reducing its already low power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications market where very low power consumption constitutes an important factor. The following are the major features of the MC146805H2 MCU.

HARDWARE FEATURES

- Typical Full Speed Operating Power of 20 mW at 5 V
- Typical WAIT Mode Power of 4 mW
- Typical STOP Mode Power of 5 μW
- 8-Bit Architecture
- Fully Static Operation
- 112 Bytes of On-Chip RAM
- 2048 Bytes on On-Chip ROM
- 24 Bidirectional I/O Lines Plus Four Input-Only Lines
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- Watchdog Timer
- External Timer Input
- External and Timer Interrupts
- Self-Check Mode
- Master Reset and Power-On Reset
- Single 3- to 6-Volt Supply
- On-Chip Oscillator
- 40-Pin Dual-in-Line Package
- Chip Carrier Also Available
- Alert Tone Generator
- Frequency Synthesizer

SOFTWARE FEATURES

- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Most Self-Check Routines User Callable



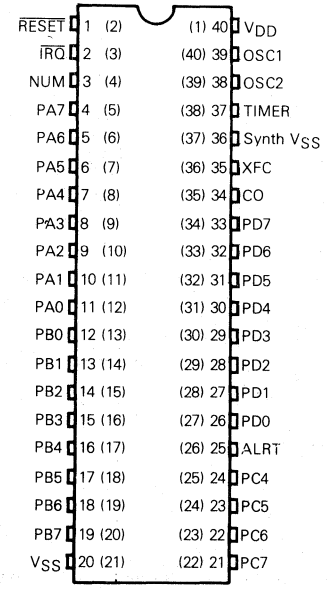
L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

Z SUFFIX
CHIP CARRIER
CASE 761

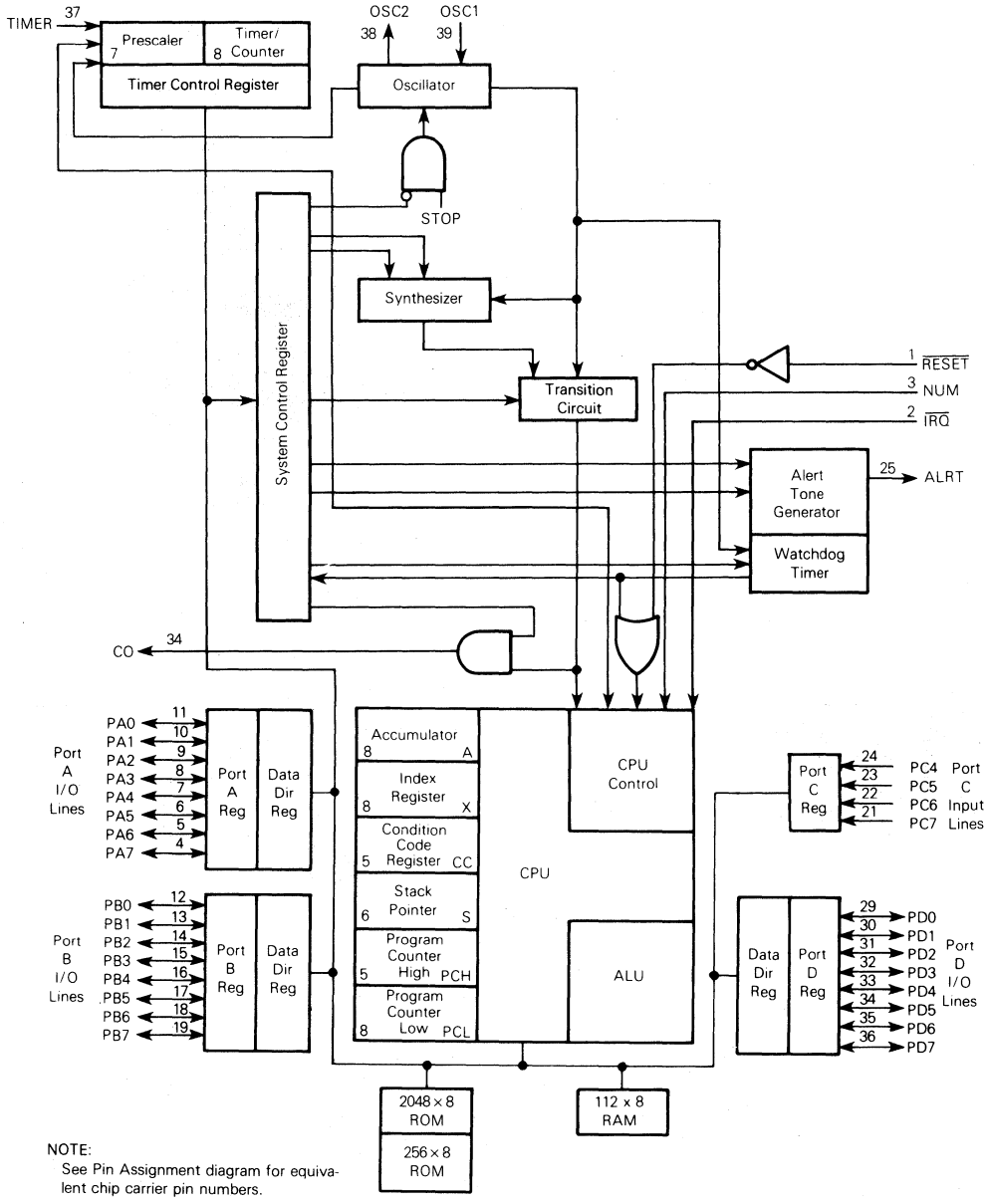
PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — BLOCK DIAGRAM



NOTE:
See Pin Assignment diagram for equivalent chip carrier pin numbers.

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages 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 $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	100	°C/W
Cerdip		60	
Ceramic		50	
Chip Carrier		TBD	

3

DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=0°C$ to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \le 10.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Output High Voltage				
Alert Tone Generator ($I_{Load}=-2$ mA)	V_{OH}	2.4	–	V
CO ($I_{Load}=-4$ mA)		2.4	–	
PA0-PA7, PD0-PD7 ($I_{Load}=-2$ mA)		2.4	–	
PB0-PB7 ($I_{Load}=-100\ \mu A$)		2.4	–	
Output Low Voltage				
Alert Tone Generator ($I_{Load}=900\ \mu A$)	V_{OL}	–	0.4	V
CO ($I_{Load}=800\ \mu A$)		–	0.4	
PA0-PA7, PB0-PB7, PD0-PD7 ($I_{Load}=800\ \mu A$)		–	0.4	
Input High Voltage				
PA0-PA7, PB0-PB7, PC4-PC7, PD0-PD7	V_{IH}	$V_{DD}-2.0$	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$		$V_{DD}-0.8$	–	
OSC1		$V_{DD}-0.8$	–	
Input Low Voltage All Inputs	V_{IL}	–	0.8	V
Total Supply Current ($C_L=50$ pF on Ports, no dc Loads, $t_{cyc}=1\mu s$)				
RUN ($V_{IL}=0.2$ V, $V_{IH}=V_{DD}-0.2$ V)	I_{DD}	–	TBD	mA
WAIT		–	TBD	mA
STOP		–	TBD	μA
I/O Port Input Leakage	I_{IL}	–	±10	μA
Input Curent				
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	μA
Capacitance				
Ports	C_{out}	–	12	pF
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{in}	–	8	pF

TBD = To be determined.

NOTE:
Test conditions for I_{DD} are as follows:
All ports programmed as inputs
$V_{IL}=0.2$ V (PA0-PA7, PB0-PB7, PC4-PC7, PD0-PD7)
$V_{IH}=V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
OSC1 input as a squarewave from 0.2 V to $V_{DD}-0.2$ V
OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance. STOP I_{DD} is also affected linearly by this capacitance if the oscillator is not killed.)

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3.0 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V V
Output High Voltage Alert Tone Generator ($I_{Load} = -0.5$ mA) CO ($I_{Load} = -1.0$ mA) PA0-PA7, PD0-PD7 ($I_{Load} = -0.5$ mA) PB0-PB7 ($I_{Load} = -50\ \mu A$)	V_{OH}	 1.4 1.4 1.4 1.4	 – – – –	 V
Output Low Voltage Alert Tone Generator ($I_{Load} = 900\ \mu A$) CO ($I_{Load} = 800\ \mu A$) PA0-PA7, PB0-PB7, PD0-PD7 ($I_{Load} = 800\ \mu A$)	V_{OL}	 – – –	 0.3 0.3 0.3	 V
Input High Voltage PA0-PA7, PB0-PB7, PC4-PC7, PD0-PD7 TIMER, $\overline{IRQ}$, $\overline{RESET}$ OSC1	V_{IH}	 $V_{DD}-0.3$ $V_{DD}-0.3$ $V_{DD}-0.3$	 – – –	 V
Input Low Voltage All Inputs	V_{IL}	–	0.3	V
Total Supply Current (C_L = 50 pF on Ports no dc Loads, t_{cyc} = 5 µs) RUN (V_{IL} = 0.2 V, $V_{IH} = V_{DD}-0.2$ V) WAIT STOP	I_{DD}	 – – –	 TBD TBD TBD	 mA mA µA
I/O Ports Input Leakage Current	I_{IL}	–	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	µA
Capacitance Ports $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	 C_{out} C_{in}	 – –	 12 8	 pF pF

TBD = To be determined.

NOTE:

Test conditions for I_{DD} are as follows:

All ports programmed as inputs

V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC4-PC7, PD0-PD7)

$V_{IH} = V_{DD}-0.2$ V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER

OSC1 input is a squarewave from 0.2 V to $V_{DD}-0.2$ V

OSC2 output load = 20 pF (WAIT and STOP I_{DD} are affected linearly by the OSC2 capacitance.)

TABLE 1 — CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C)

Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	TBD	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 6)	t_{ILCH}	–	TBD	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
$\overline{RESET}$ Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width Low (See Figure 15)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (See Figure 15)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	TBD	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal Synthesizer	 f_{osc} f_{synth}	 30 0.5	 50 2.0	 kHz MHz

* The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

TABLE 2 — CONTROL TIMING
(V_{DD} = 3.0 Vdc, V_{SS} = 0 Vdc, T_A = 0°C to 70°C)

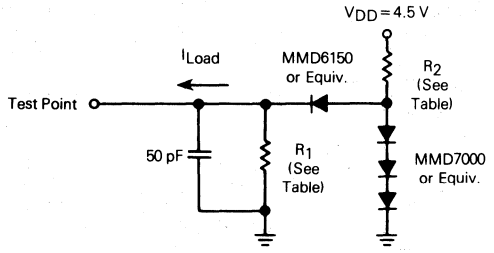
Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	TBD	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 6)	t_{ILCH}	–	TBD	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
$\overline{RESET}$ Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width Low (See Figure 14)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (See Figure 14)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	TBD	–	ns
Cycle Time	t_{cyc}	5000	–	ns
Frequency of Operation Crystal Synthesizer	 f_{osc} f_{synth}	 30 120	 50 600	 kHz kHz

*The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

3

FIGURE 2 — EQUIVALENT TEST LOAD

Port	R1	R2
B	24.3 kΩ	4.32 kΩ
A and D	1.21 kΩ	3.1 kΩ

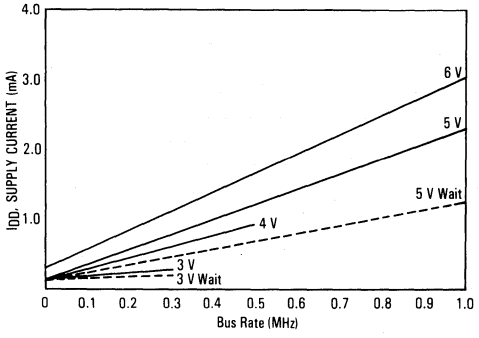


TYPICAL CURRENT CALCULATIONS

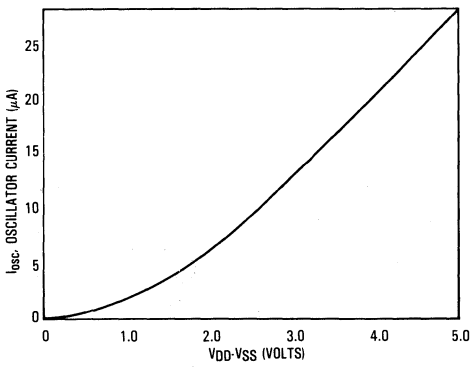
The operating current of the MCU (I_{DD}) is a function of supply voltage, bus rate, capacitive loading on any active pins (i.e., OSC1, OSC2, etc.), the processor state (RUN, WAIT, or STOP), the synthesizer state (ON or OFF), and the resistive loading on all outputs. Inputs, such as input ports can also cause significant increases in currents if they are placed in the active region of the input device. Because of this, inputs should never be allowed to simply "float". It is impossible to determine a "typical" I_{DD} for a particular application without first knowing all of the above conditions and their corresponding currents. Thus, some "typical" current curves are provided in Figure 3 (a, b, and c). It should be emphasized that these are only approximations and no minimums or maximums are implied.

FIGURE 3 — TYPICAL OPERATING CURRENT vs FREQUENCY/VOLTAGE

(a) — Processor Current vs Bus Frequency



(b) — Typical Crystal Oscillator Current vs Voltage



(c) — Typical Synthesizer Current vs Frequency

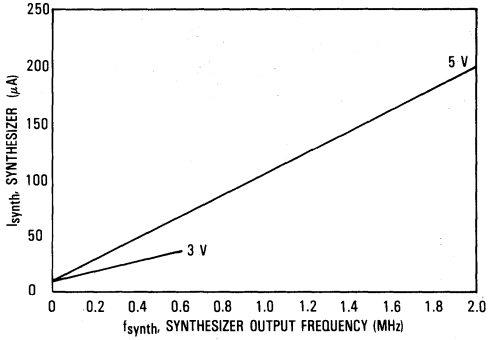


FIGURE 4 — TIMER RELATIONSHIPS

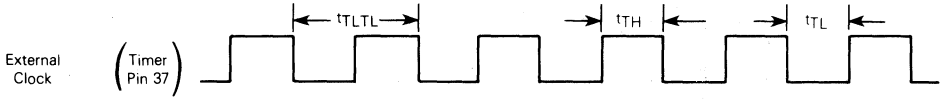
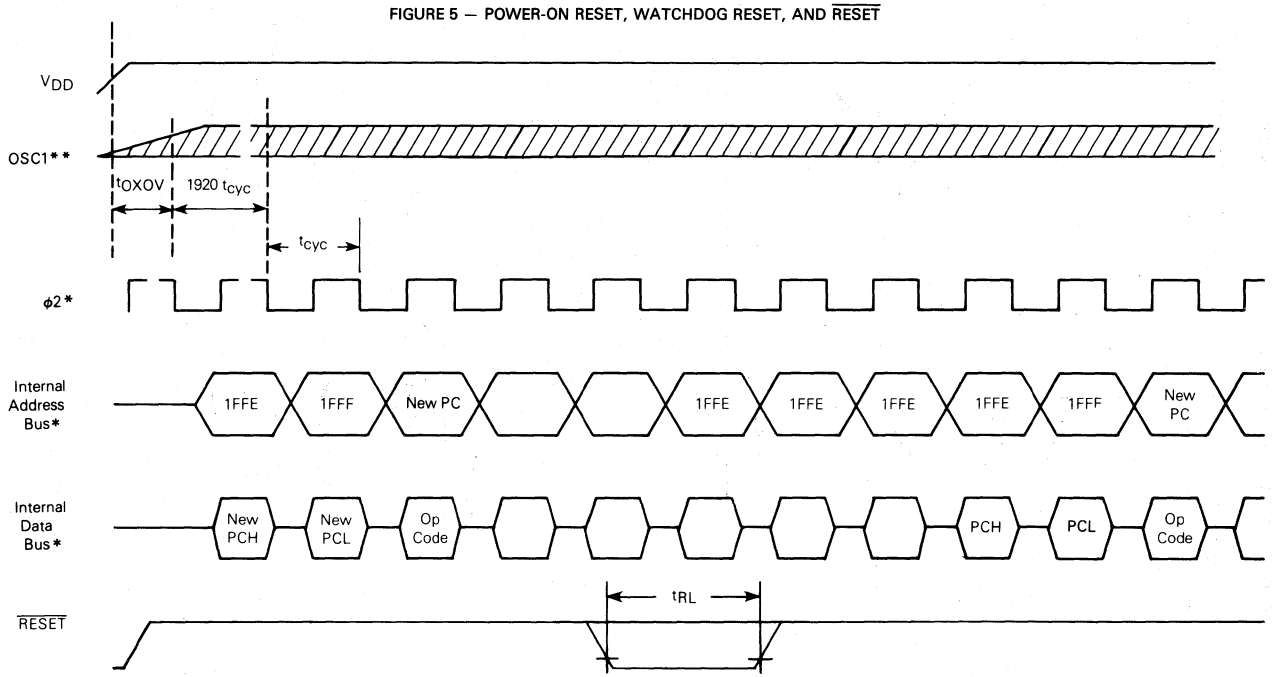
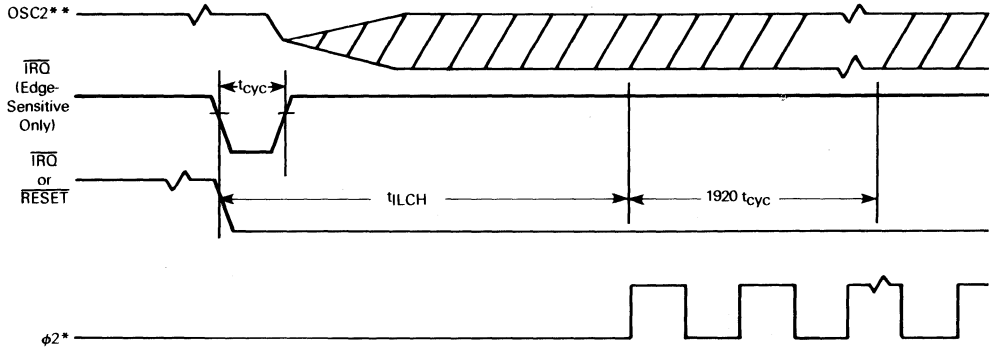


FIGURE 5 — POWER-ON RESET, WATCHDOG RESET, AND $\overline{\text{RESET}}$



*Internal timing signal and bus information not available externally.
**OSC1 line is not meant to represent frequency. It is only used to represent time.

FIGURE 6 — STOP RECOVERY AND POWER-ON RESET



*Internal timing signals not available externally.
**Represents the internal control of crystal oscillator.

FUNCTIONAL PIN DESCRIPTION

V_{DD}, V_{SS}, AND SYNTH V_{SS}

Power is supplied to the MCU using these pins. V_{DD} is power and V_{SS} is ground. A separate ground is provided for the synthesizer which must be at the same potential as V_{SS}. These grounds (synthesizer V_{SS} and V_{SS}) may be bypassed independently to minimize noise if necessary.

IRQ (MASKABLE INTERRUPT REQUEST)

IRQ is a mask programmable option which provides two different choices of interrupt triggering sensitivity. These options are: 1) negative edge-sensitive triggering only, or 2) both negative edge-sensitive and level-sensitive triggering. In the latter case, either type of input to the IRQ pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When the IRQ pin goes low for at least one t_{CYC}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

If the option is selected to include level-sensitive triggering, then the IRQ input requires an external resistor to V_{DD} for "wire-OR" operation. See **INTERRUPTS** for more detail.

RESET

The RESET input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure. Refer to **RESETS** for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to **TIMER** for additional information about the timer circuitry.

OSC1, OSC2

The MC146805H2 is configured to accept a crystal to control the internal oscillator. An external clock may also be used. These are discussed below.

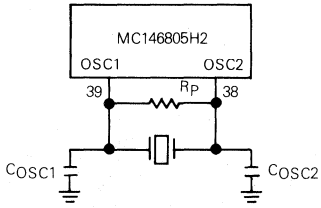
CRYSTAL — The circuit shown in Figure 7(b) is recommended when using a crystal. The internal oscillator is designed to interface with a parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC} in Table 1 and Table 2 control timing. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with OSC2 not connected, as shown in Figure 7(d). The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

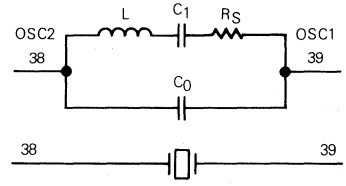
FIGURE 7 — OSCILLATOR CONNECTIONS

	Vendor A	Vendor B	Units
$R_{S(Max)}$	50	23	kΩ
C_0	0.8	1.5	pF
C_1	–	2.35	pF
C_{OSC1}	10	35	pF
C_{OSC2}	2.5-10	5-30	pF
Q	60	90	K
R_P	2-10	2-10	MΩ

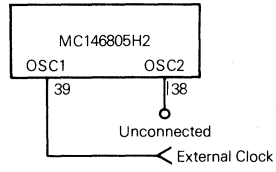
(a) Typical Crystal Parameters @ 32.768 kHz



(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit



(d) External Clock Source Connections

CO

This pin provides a clock output which represents the clock input to the CPU; however, it is twice the frequency of the bus rate since the CPU divides the clock by two to obtain the bus rate. It can be used to provide an external synchronizing clock or as a test point for checking the on-chip clock input to the CPU.

ALRT

This output provides one of three tone signals to drive an external amplifier whenever the tone generator is activated by the microcomputer program. Whenever the tone generator is turned off, this pin represents a high impedance. Refer to **AUDIO ALERT TONE GENERATOR** for more information.

XFC

This pin provides a means for connecting an external capacitor to the synthesizer phase lock loop filter. Refer to **PHASE LOCK LOOP** for additional information concerning this capacitor.

PA0-PA7

These eight I/O lines comprise port A. The state of any pin is software programmable. Refer to **INPUT/OUTPUT PROGRAMMING** for a description of I/O programming.

PB0-PB7

These eight lines comprise port B. The state of any pin is software programmable. Refer to **INPUT/OUTPUT PROGRAMMING** for a description of I/O programming.

PC4-PC7

These four lines comprise port C, a fixed input port. When port C is read, the four least significant bits on the data bus are zeros. There is no data direction register associated with port C.

PD0-PD7

These eight lines comprise port D. The state of any pin is software programmable. Refer to **INPUT/OUTPUT PROGRAMMING** for a description of I/O programming.

INPUT/OUTPUT PROGRAMMING

Any port A, B, or D pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A particular port A, B, or D pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At reset, all DDRs are cleared, which configures all port A, B, and D pins as inputs. Port C is input only. A particular port A, B, or D pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 8 and Table 3.

FIGURE 8 — TYPICAL PORT A, B, OR D I/O CIRCUITRY

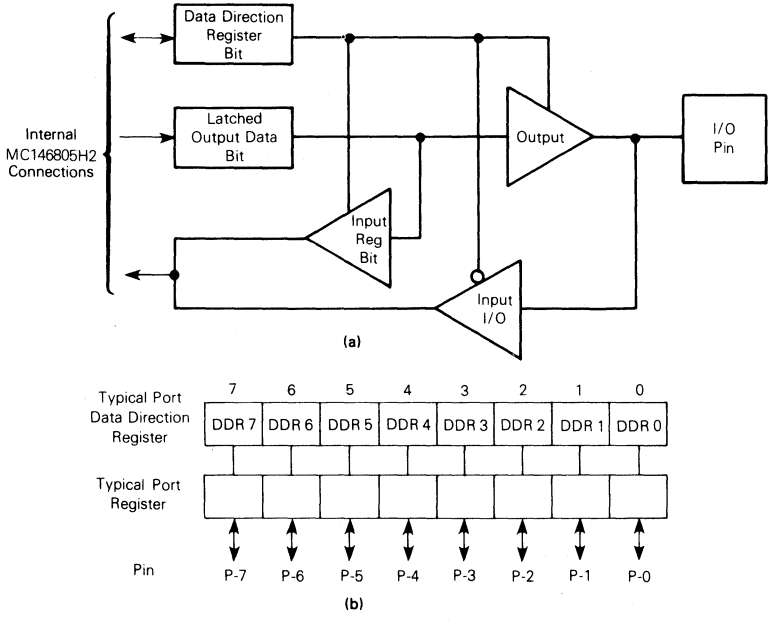


TABLE 3 — PORT A, B, OR D I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

SELF-CHECK

The MC146805H2 self-check is performed using the circuit in Figure 9. Self-check is initiated by connecting the NUM and TIMER pins to a logic one and then executing a reset. After reset, five subroutines are called that execute the following tests:

I/O — Functionally exercises ports A, B, C, D
RAM — Walking bit test
ROM — Exclusive OR with odd ones parity result
Timer — Functionally exercise timer
Interrupts — Functionally exercise external and timer interrupts

Self-check results are shown in Table 4. The following subroutines are available to user programs and do not require any external hardware.

FIGURE 9 — SELF-CHECK CIRCUIT

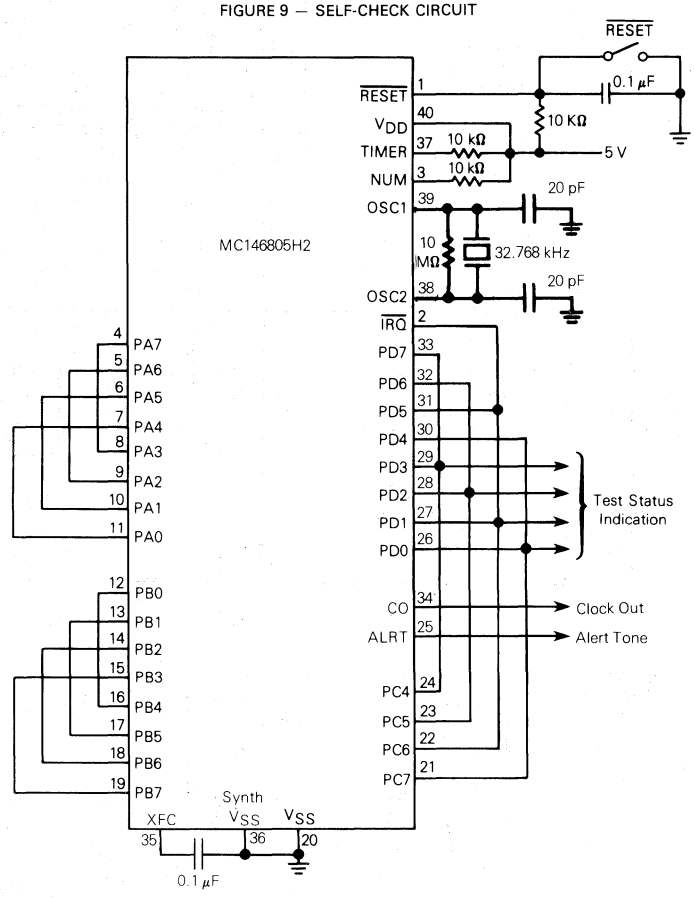


TABLE 4 — SELF-CHECK RESULTS

PD3	PD2	PD1	PD0	Remarks
1	0	1	0	Bad I/O
1	0	1	1	Bad Timer
1	1	0	0	Bad RAM
1	1	0	1	Bad ROM
1	1	1	0	Bad Interrupt or Request Flag
Cycling				Good Part
All Others				Bad Part

RAM SELF-CHECK SUBROUTINES

Returns with the Z bit clear if any error is detected; otherwise the Z bit is set.

The RAM test must be called with the stack pointer at $007F. When run, the test checks every RAM cell except for $007F and $007E which are assumed to contain the return address.

A and X are modified. All RAM locations except the top two are modified.

ROM CHECKSUM SUBROUTINE

Returns with Z bit cleared if any error was found; otherwise Z = 1. X = 0 on return, and A is zero if the test passed. RAM locations $0040-$0043 are overwritten.

TIMER TEST SUBROUTINE

Returns with Z bit cleared if any error was found; otherwise Z = 1.

This routine runs a simple test on the timer. In order to work correctly as a user subroutine, the internal clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock will be running and the interrupt mask not set so the caller must protect himself from interrupts if necessary.

The A and X register contents are lost since this routine uses them in determining how many times the clock counts in 128 cycles. The number of counts should be a power of two since the prescaler is a power of two. If not, the timer probably is not counting correctly. The routine also detects if the timer is running at all.

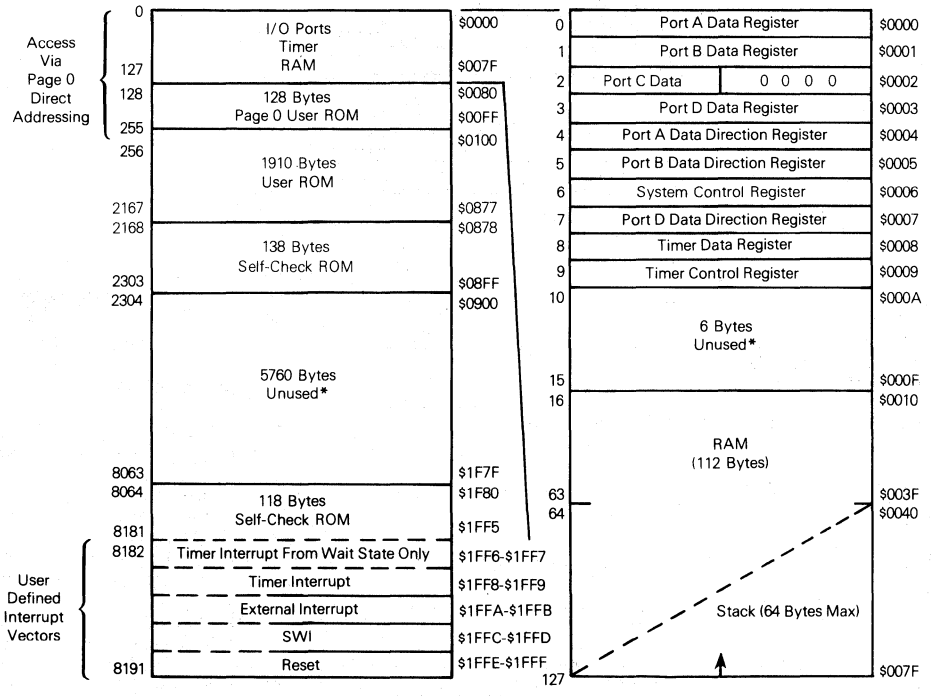
MEMORY

The MC146805H2 has a total address space of 8192 bytes of memory and I/O registers. The address space is shown in Figure 10.

The first 128 bytes of memory (first half of page zero) are comprised of the I/O port locations, timer locations, and 112 bytes of RAM. The next 2038 bytes (including the 128 bytes of the second half of page zero) comprise the user ROM. The 10 highest address bytes contain the reset and the interrupt vectors.

The stack pointer is used to address data stored on the stack. Data is stored on the stack during interrupts and subroutine calls. At power up, the stack pointer is set to $007F and it is decremented as data is pushed onto the stack. When data is removed from the stack, the stack pointer is incremented. A maximum of 64 bytes of RAM is available for stack usage. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

FIGURE 10 — ADDRESS MAP



*Reads of unused locations undefined.

REGISTERS

The MC146805H2 contains five registers, as shown in the programming model of Figure 11. The interrupt stacking order is shown in Figure 12.

ACCUMULATOR (A)

The accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently configured to 0000001. These seven bits are appended to the six least significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($007F). Nested interrupt and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer wraps around and points to its upper limit thereby losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each bit is explained in the following paragraphs.

FIGURE 11 — PROGRAMMING MODEL

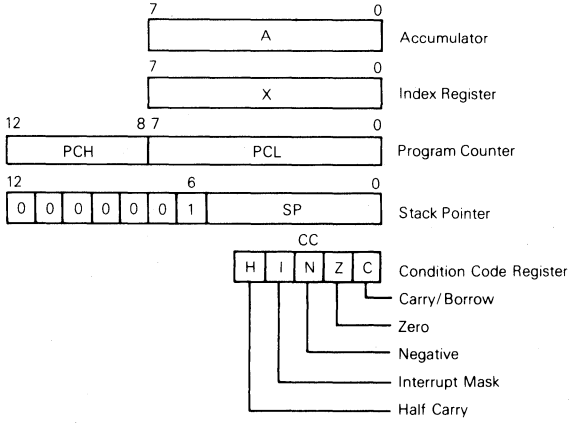
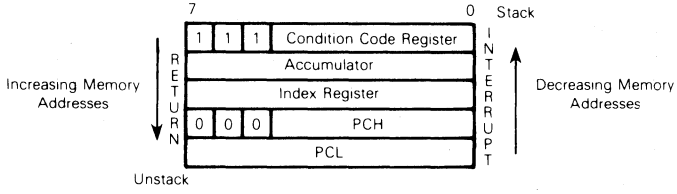


FIGURE 12 — STACKING ORDER



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

HALF CARRY BIT (H) — The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU and during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed after the I bit is next cleared.

NEGATIVE (N) — When set, this bit indicates that the results of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

ZERO (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) — When set, this bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The MC146805H2 has three reset modes: an active low external reset pin ($\overline{\text{RESET}}$), a power-on reset function, and a dead-man timer reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{CYC}. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input (internally) to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for

power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{cyc} delay from the time that the oscillator becomes active. If the external $\overline{RESET}$ pin is low at the end of the 1920 t_{cyc} time out, the processor remains in the reset condition until $\overline{RESET}$ goes high.

WATCHDOG TIMER

The watchdog timer contains an 18-stage divider which is clocked by the crystal oscillator output. A program controlled input (SCR4) from the system control register clears the watchdog timer as described in **SYSTEM CONTROL REGISTER.** Unless the watchdog timer is periodically cleared by the system program it will time out and reset the MCU.

Since the watchdog timer is connected directly to the crystal oscillator it is not affected by the oscillator gating as discussed in **SYSTEM CONTROL REGISTER.** If the oscillator is left running, the watchdog timer will time out and reset the MCU. This is a safety feature to preclude the MCU from becoming "lost", and inadvertently stopping the oscillator and thereby "killing" itself. As will be discussed in **SYSTEM CONTROL REGISTER,** the SCR3 bit is heavily protected to protect the system during a loss of MCU control.

RESET CONDITIONS

Either of the three types of reset conditions causes the following to occur:

- Timer control register interrupt request bit TCR7 is cleared to a logic zero to preclude premature timer interrupts.
- Timer control register interrupt mask bit TCR6 is set to a logic one to preclude timer interrupt processing.
- All data direction register bits are cleared to logic zeros to define all ports as input.
- Stack pointer is preset to its upper limit, $007F.
- The internal address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a logic one to mask any external interrupts.
- STOP and WAIT latches are cleared to place MCU in normal operation.
- External interrupt latch is cleared to ensure no external interrupt is processed.
- System control register bits SCR6, SCR7, and SCR2 are cleared; however, bits SCR5, SCR3, SCR1, and SCR0 are set. Bit SCR4 could be either set or cleared as discussed in the **SYSTEM CONTROL REGISTER.**

All other functions, such as other registers, the timer, etc. are not cleared by the reset conditions.

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The MC146805H2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a nonmaskable software interrupt (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 12.

Unlike $\overline{RESET}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is completed.

NOTE

The current instruction is considered to be the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if an interrupt is pending and is unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt servicing.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction. Refer to Figure 13 for the interrupt and instruction processing sequence.

Table 5 shows the execution priority of the $\overline{RESET}$, $\overline{IRQ}$, timer interrupts, and the software interrupt, SWI. Two conditions are shown, one with the I bit set and the other with the I bit clear; however, in either case $\overline{RESET}$ has the highest priority of execution. If the I bit is set as per Table 5(a), the second highest priority is assigned to SWI. This is illustrated in Figure 13 which shows that the $\overline{IRQ}$ or timer interrupts are not executed when the I bit is set. If the I bit is clear as per Table 5(b), the priorities change in that the next instruction (including SWI) is not fetched until after the $\overline{IRQ}$ and timer interrupts have been recognized (and serviced). Also, when the I bit is clear, if both $\overline{IRQ}$ and timer interrupts are pending, the $\overline{IRQ}$ interrupt is always serviced before the timer interrupt.

NOTE

The conditions for Table 5 assume that, except for $\overline{RESET}$, the current instruction is completed; thus the MCU is at an instruction boundary. Processing is such that at the end of the current instruction, the I bit is tested and if set the next instruction (including SWI) is fetched. If the I bit is cleared, the hardware interrupt latches are tested, and if no hardware interrupt is pending, the program falls through and the next instruction is fetched.

TABLE 5 — INTERRUPT INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
SWI	2	$1FFC-$1FFD

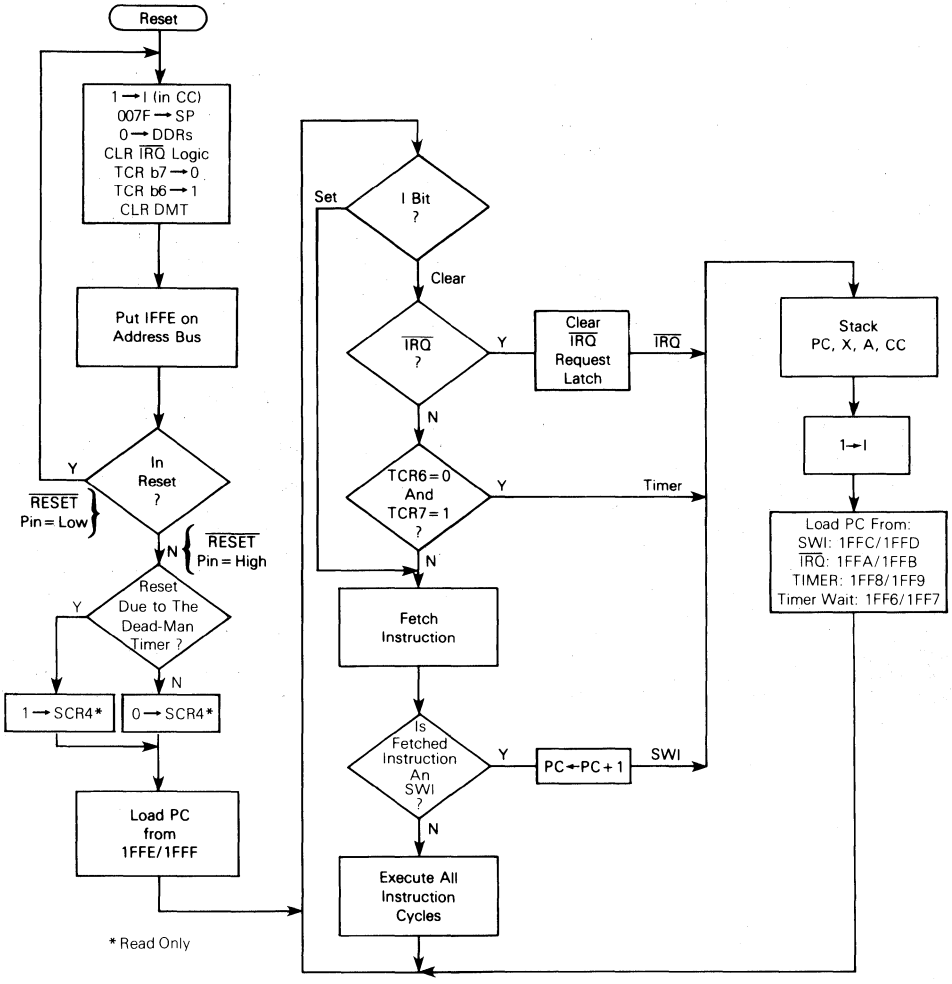
NOTE: $\overline{\text{IRQ}}$ and Timer Interrupts are not executed when the I bit is set; therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
$\overline{\text{IRQ}}$	2	$1FFA-$1FFB
Timer	3	$1FF8-$1FF9
		$1FF6-$1FF7*
SWI	4	$1FFC-$1FFD

*The timer vector address from the WAIT mode is $1FF6-$1FF7.

FIGURE 13 — RESET AND INTERRUPT PROCESSING FLOWCHART



TIMER INTERRUPT

If the timer interrupt mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00 to set TCR7) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit (in the condition code register) is cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode, in which case the contents of $1FF6 and $1FF7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

The actual timer interrupt request can be delayed by controlling TCR6 (interrupt mask bit). If TCR6 is programmed to a logic one, no interrupt is generated even if TCR7 (interrupt request bit) is set. Then TCR6 can be programmed (after a specific time) to a logic zero to generate the actual timer interrupt request.

EXTERNAL INTERRUPT

If the interrupt mask bit of the condition code register has been cleared and the external interrupt pin ($\overline{IRQ}$) has gone low, then the external interrupt is recognized. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{IRQ}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB. Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as a mask option. Figure 14 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{IRQ}$) to the MCU. The first method shows single pulses on the interrupt line space far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service routine. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 15. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT

The software interrupt (SWI) is an executable instruction. The action of the software interrupt instruction is similar to the hardware interrupts. The software interrupt is executed regardless of the state of the interrupt mask bit in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 13 for interrupt and instruction processing flowchart.

LOW-POWER MODES

STOP

The STOP instruction places the MC146805H2 in its lowest power consumption mode. In the STOP mode, all internal processing and the timer operation are halted; refer to Figure 15.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can only be brought out of the STOP mode by an external interrupt, reset, or dead-man timer timeout.

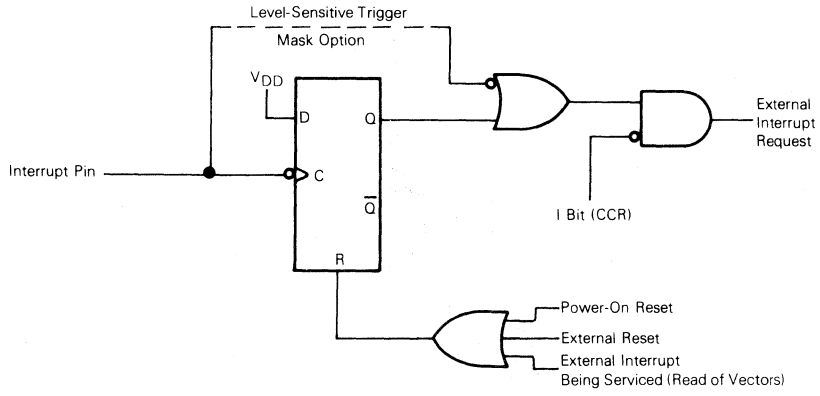
WAIT

The WAIT instruction places the MC146805H2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 16. Thus, all internal processing is halted; however, the timer continues to count normally.

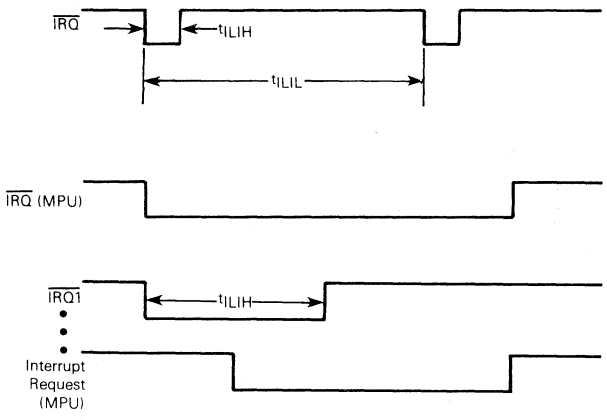
During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

FIGURE 14 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Normally used with Wire-ORed Connection

FIGURE 15 — STOP FUNCTION FLOWCHART

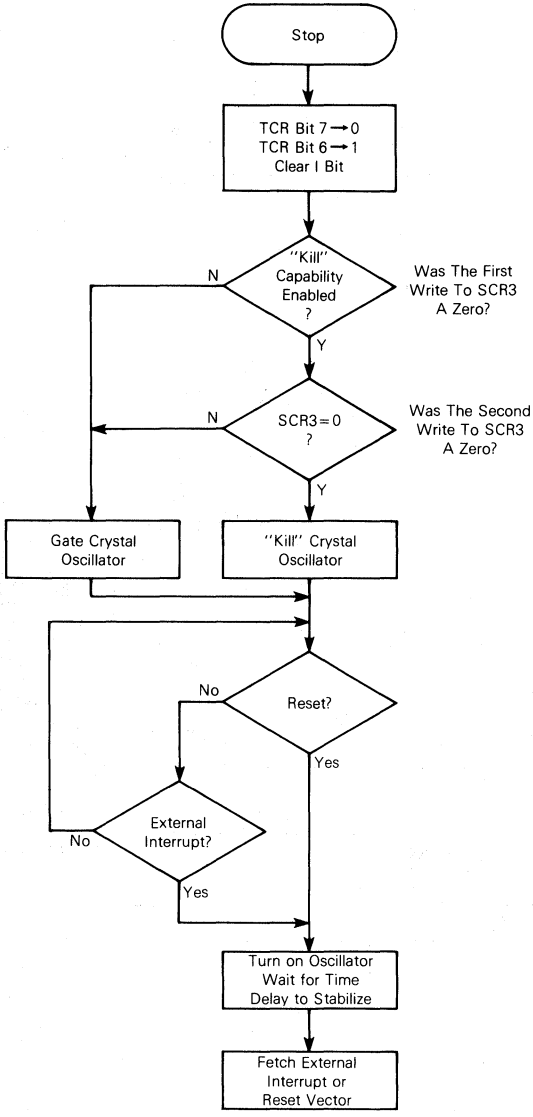
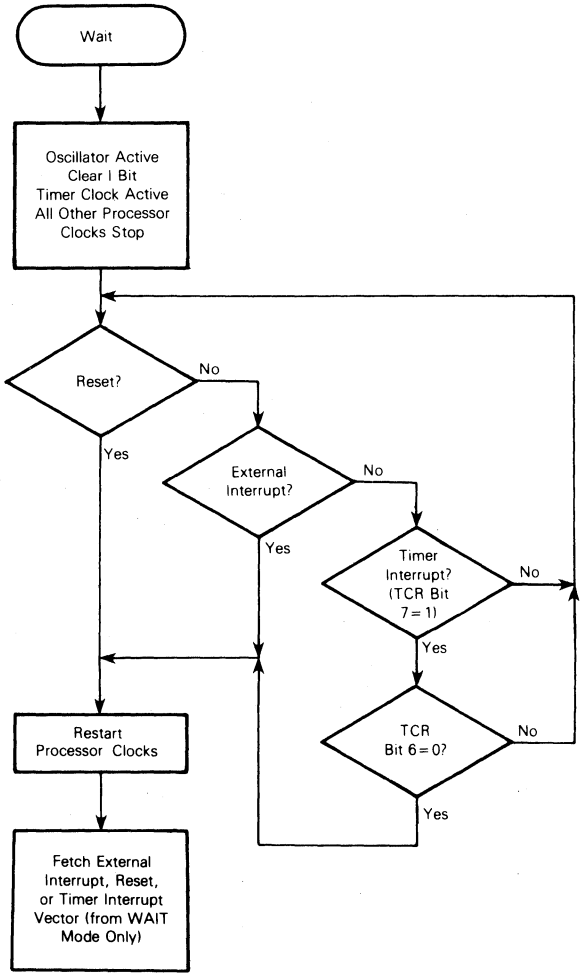


FIGURE 16 — WAIT FUNCTION FLOWCHART



TIMER

The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 17 contains a block diagram of the timer. The counter may be loaded under program control and is decremented towards zero by the clock input (prescaler output). When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register, TCR) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 (or $1FF6 and $1FF7 if in the WAIT mode) in order to begin servicing; refer to **INTERRUPTS.**

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter

become stable prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit (TCR7) remains set until cleared by the software. If the timer interrupt request bit (TCR7) is cleared before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals a logic one. This allows for truncation-free counting.

The timer input can be configured for four different operating modes, depending on the value written to the TCR4 and TCR5 timer control register bits. Refer to **TIMER CONTROL REGISTER.**

TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a zero, the input to the timer is from the internal processor clock and the TIMER input pin is disabled. The processor clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The processor clock is the instruction cycle clock. During a WAIT instruction, the processor clock input to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR4 = 1 and TCR5 = 0, the internal processor clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal processor clock for the duration of the pulse. The resolution of the count in this mode is plus or minus one clock cycle; therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

If TCR4 = 0 and TCR5 = 1, then the crystal oscillator is used to clock the timer. This clock source is independent of the internal processor clock and thus it is useful in keeping real time. It is particularly useful with a 32.768 kHz crystal where the maximum modulus of the timer results in a precise one second interrupt rate.

TIMER INPUT MODE 4

If TCR4 = 1 and TCR5 = 1, the internal processor clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

Figure 18 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction cause the counter to be set to $F0.

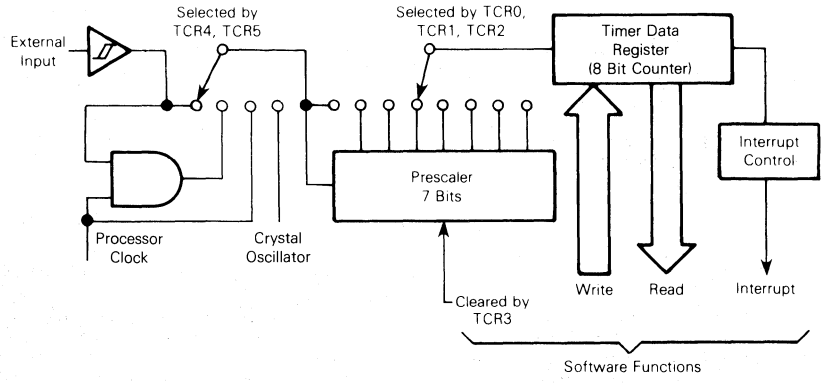
TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0	
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0	$0009

All bits in this register except bit 3 are read/write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic one.

FIGURE 17 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register (8-bit counter) are clocked on the falling edge of the internal processor clock, crystal oscillator, or external input.
2. The timer data register counts down continuously.

1 – Set whenever the counter decrements to zero, or under program control.
0 – Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 – Timer interrupt mask bit: when this bit is a logic one it inhibits the timer interrupt to the processor.
1 – Set on external reset, power-on reset, STOP instruction, or program control.
0 – Cleared under program control.

TCR5, TCR4 – Together, these two bits control the input to the timer. This is illustrated in the table below. (These two bits are unaffected by reset.)

TCR5	TCR4	
0	0	Processor clock to timer
0	1	AND of processor clock and TIMER pin
1	0	Crystal oscillator to timer
1	1	TIMER pin to timer

TCR3 – Timer prescaler reset bit: writing a one to this bit resets the prescaler to zero. A read of this location always indicates a zero. (Unaffected by reset.)

TCR2, TCR1, TCR0 – Prescaler select bits: decoded to select one of eight outputs of the prescaler. (Unaffected by reset.)

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

BUS RATES

The MC146805H2 MCU has the ability to change processor clock rate under program control. This is accomplished by utilizing the MCU program to select either the crystal oscillator or an internal synthesizer as the processor clock source. The entire circuit contains the crystal oscillator, synthesizer, gating and stop circuits, and a transition control circuit (refer to the block diagram of Figure 1). Rate selection is made by writing the system control register bit SCR2. Refer to **SYSTEM CONTROL REGISTER** for a discussion of this and all related controls.

CRYSTAL OSCILLATOR

The crystal oscillator is designed to operate in the 30 to 50 kHz range for use with low frequency crystals such as the 32.768 kHz watch crystal. The oscillator stop select circuit allows the STOP instruction to completely shut down MCU operation by "killing" the oscillator or by gating off its output. By controlling the gating off of the oscillator, the user is provided with an option of either very low current drain or rapid recovery from the STOP function; i.e., with the oscillator running but gated off during the STOP function, the start up time (t_{OXOV}) could be much shorter than if the oscillator were "killed."

FREQUENCY SYNTHESIZER

The frequency synthesizer uses a conventional phase lock loop which utilizes the crystal oscillator output as its reference frequency. The synthesizer output frequency is 16 f_{osc} for the 3-volt port and 64 f_{osc} for the 5-volt port (f_{osc} represents the crystal oscillator frequency). The synthesizer bandwidth (wide or narrow) is program controlled to provide two different damping factors. This bandwidth control is provided by the system control register SCR0 bit. An external filter capacitor must be connected to the XFC pin (35) as part of the frequency synthesizer loop filter.

TRANSITION CONTROL CIRCUIT

The transition control circuit provides a means for a smooth transition when switching the processor clock between the crystal oscillator and frequency synthesizer. Switching of the transition circuit is controlled by a bit in the system control register. A buffered output of the transition circuit is available at the CO pin whenever system control register SCR1 bit is set.

WATCHDOG TIMER

The watchdog timer is designed to periodically time out and reset the MCU unless it is periodically cleared by the system program (the watchdog timer is also cleared during any other MCU reset). The time-out period (t_{DM}), which for a 32.768 kHz crystal = 3.00 seconds, is calculated as:

$$t_{DM} = (98{,}296 \pm 16)\left(\frac{1}{f_{osc}}\right)$$

NOTE

The variation is the result of never clearing the first four stages of the timer which are used for other internal functions.

A reset from the watchdog timer affects the CPU in the same manner as an external reset ($\overline{RESET}$ pin goes low); however, the watchdog timer reset also sets the SCR4 bit in the system control register. Refer to **SYSTEM CONTROL REGISTER** for additional information.

AUDIO ALERT TONE GENERATOR

The alert tone generator provides a buffered tone output at the ALRT pin (25). This output provides drive for an external amplifier whenever the tone generator is activated by the microcomputer program; otherwise, the ALRT pin represents a high impedance. Three different tones can be generated by the alert tone generator and these tones are controlled by two bits (SCR6, SCR7) in the system control register. A table illustrating the SCR6-SCR7 bits status versus the alert tone generator output is shown below. Bits SCR6 and SCR7 are cleared by a reset.

SCR7	SCR6	
0	0	No output from alert generator (pin 25 high impedance)
0	1	Low frequency output from alert generator (crystal oscillator frequency ÷ 32)
1	0	Medium frequency output from alert generator (crystal oscillator frequency ÷ 16)
1	1	High frequency output from alert generator (crystal oscillator frequency ÷ 8)

SYSTEM CONTROL REGISTER

The system control register is an 8-bit register which provides control bits SCR0 through SCR7. These bits are used in determining whether the system clock is furnished by the crystal oscillator or synthesizer, plus controlling the alert generator and resetting the dead-man timer.

7	6	5	4	3	2	1	0	
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	$0006
				SCR3				

SCR6, SCR7 — Together, these two bits control the state and frequency of the alert generator as shown in **AUDIO ALERT TONE GENERATOR.** Both of these bits are cleared by reset.

SCR5 — This bit determines the on-off state of the synthesizer. This bit is set during reset. If the synthesizer output is already selected by SCR2 (SCR2 = 1), it cannot be turned off by SCR5 (SCR5 will remain at 1 to ensure that the selected synthesizer cannot be turned off).

1 = Synthesizer on
0 = Synthesizer off

SCR4 — This bit has two different functions. When it is read for the first time following a reset, it is a reset qualifier bit. When it is written to, it is used to clear the dead-man timer.

Read Cycle
1 = Indicates MCU was last reset by dead-man timer. This bit is cleared by a read of system control register.
0 = Indicates MCU was last reset by a power-on or external reset.

Write Cycle
1 = Clears the dead-man timer.
0 = No action taken.

SCR3 — This bit performs two functions. Together, these functions control the operation of the crystal oscillator in the STOP mode. The first write to this bit accesses a latch which can only be modified by the first write. After the first write, all other writes to SCR3 are written into a second latch. Only if both of these latches are zero will the oscillator be stopped (or "killed") when a STOP instruction is executed. See note and logic diagram below for further discussion.

1 = Gate crystal oscillator on.
0 = "Kill" crystal oscillator.

SCR2 — This bit is used to determine which clock (synthesizer or crystal oscillator) is passed to the CPU (via the transition control circuit). This bit is cleared during reset. If the synthesizer is off (SCR5 = 0), it cannot be selected (SCR2 will remain at zero to ensure that a turned off synthesizer cannot be selected).

1 = Select synthesizer output.
0 = Select crystal oscillator output.

SCR1 — This bit is used to either enable or disable the clock output (pin 34). This bit is set during reset.

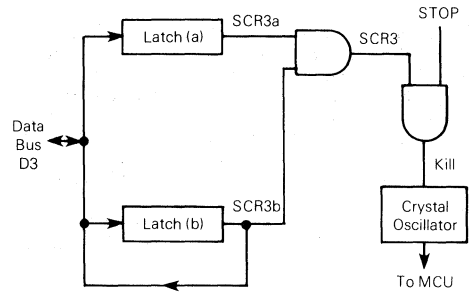
1 = CO output enabled.
0 = CO output disabled (pin 34 held low).

SCR0 — This bit controls the loop bandwidth of the phase lock loop frequency synthesizer. In the wide bandwidth mode it will lock on frequency quickly. Once it is locked, the narrow bandwidth should be used to maintain better frequency stability. This bit is set during reset.

1 = Wide bandwidth.
0 = Narrow bandwidth.

NOTE

The first write after reset (external, power on, or dead-man timer) will determine the crystal oscillator control function. That is, if a one is written to SCR3 as the first write after reset, then the crystal oscillator can never be stopped ("killed") but only gated. However, if a zero is written to SCR3 as the first write after reset, then the oscillator can be "killed" if SCR3 = 0 and a STOP instruction is executed. This method of controlling the oscillator is possible because the SCR3 bit is contained in two different latches (as illustrated logically below). Reads are always from SCR3b. The contents of latch (a) are only affected by the first write to the system control register following an MCU reset. Subsequent writes to this register will alter data in latch (b) only. Thus, all reads before the second write, will always be a one.



PHASE LOCK LOOP

The phase lock loop (PLL) consists of: a digital phase detector, a variable bandwidth loop filter, a voltage controlled oscillator (VCO), and a feedback frequency divider. A small external capacitor (typically 0.1 microfarads) is used by the loop filter. The synth V_{SS} pin is the ground for the PLL and may be bypassed to minimize noise.

The phase detector compares the frequency and phase of the feedback frequency (f_{FB}) and the crystal oscillator reference frequency (f_{REF}) and generates the output, $\phi COMP$, as shown in Figure 18. The output waveform is then integrated and amplified. The resultant dc voltage is applied to the voltage controlled oscillator. The output of the VCO is divided by a fixed frequency divider of 64 (in the 5-volt part) or 16 (in the 3-volt part) to provide the feedback frequency for the phase detector.

The startup time and frequency stability of the PLL can be changed via the variable bandwidth control in the loop filter. For the fastest startup, the low stability mode (SCR0 = 1) should be used. The high stability mode (SCR0 = 0) responds slowly and is normally used only after the PLL is at or near the operating frequency (see Figure 19).

The loop filter can source or sink only small currents in the high stability mode (approximately 1 microamp). Therefore, the external filter capacitor (XFC) should be selected for very low leakage. The printed circuit board must be clean and free from conductive material. The capacitor should be located as close to the microcomputer as possible to minimize noise.

FIGURE 18 — PHASE DETECTOR

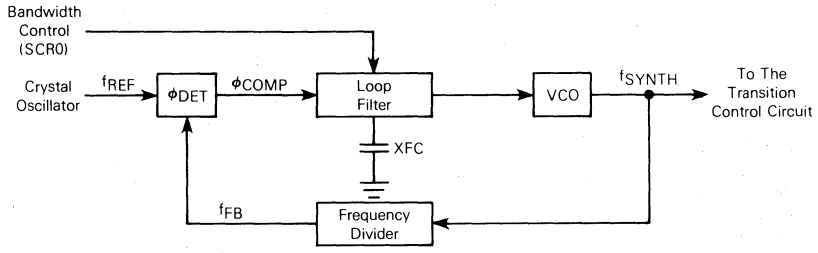
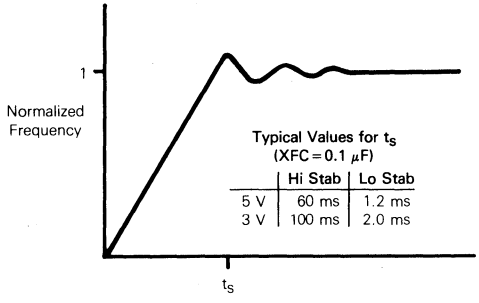


FIGURE 19 — TYPICAL STARTUP OF THE PLL



INSTRUCTION SET

The MCU has a set of 61 basic instructions. 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 6.

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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 7.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between − 127 and + 128 to the current program counter. Refer to Table 8.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space (where all port registers, port DDRs, timer, timer control, system control and on-chip RAM reside). Bit manipulation in the ROM mapped area will not affect data in the ROM. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to Table 9.

NOTE

The MCU is actually capable of operating on the bit set and bit clear instructions anywhere in the first 256 bytes; however, since only ROM resides in the upper 128 bytes the bit set/clear instructions have no affect on the upper 128 bytes.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 10.

OPCODE MAP

Table 11 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 12.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 12 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 11.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the *M6805 HMOS/M146805 CMOS Family Microcomputer/Microprocessor User's Manual.*

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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 lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers, and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

EA = (PC + 1):(PC + 2); PC ← PC + 3
Address Bus High ← (PC + 1); Address Bus Low ← (PC + 2)

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

EA = X; PC ← PC + 1
Address Bus High ← 0; Address Bus Low ← X

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2
Address Bus High ← K; Address Bus Low ← X + (PC + 1)
Where:
K = The carry from the addition of X + (PC + 1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8- or 16-bit. The contents of the index register is not changed.

EA = X + [(PC + 1):(PC + 2)]; PC ← PC + 3
Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)
Where:
K = The carry from the addition of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of −126 to +129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise, EA = PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified within the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2
Address Bus High − 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit set or bit clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High − 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC ← PC + 3

TABLE 6 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 7 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 8 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 9 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	—	—	—	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	—	—	—	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	—	—	—
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	—	—	—

TABLE 10 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 11 — MC146805 CMOS FAMILY INSTRUCTION SET OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
Hi / Low																	Hi / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

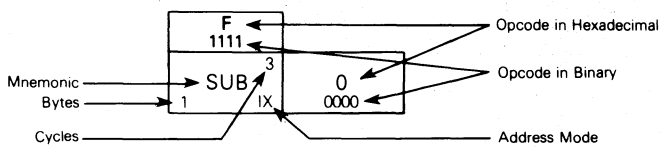


TABLE 12 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- Λ Test and Set if True, Cleared Otherwise
- ● Not Affected
- ? Load CC Register From Stack
- 0 Cleared
- 1 Set

ORDERING INFORMATION

The following information is required when ordering a custom MCU. This information may be transmitted to Motorola in the following media:

EPROM(s) MCM2716s or MCM2532s
MDOS disk file

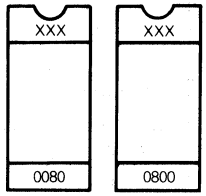
To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

EPROMs

The MCM2716 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and date), may be submitted for pattern generation. The EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure A-1 illustrates the marking for the two MCM2716 EPROMs required to emulate the MC146805H2.

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

FIGURE A-1 — EPROM MARKING



XXX = Customer ID

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 MCM2716 or MCM2532 EPROM (**supplied by the customer**) from the data file used to create the custom mask to aid in the verification process.

ROM VERIFICATION UNITS

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 included in the mask charge and are not production parts. These RVUs are not backed nor guaranteed by Motorola Quality Assurance.

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 and company name. The floppies are not returned by Motorola as they are used for archival storage. The minimum MDOS system files as well as the absolute binary object file (filename. LO type of file) from the M6805 cross assembler 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.

OPTION LIST

Select the options for the MCU from the following list. A manufacturing mask will be generated from this information. Select one in each section.

Operating Voltage
- ☐ 3 V (262 kHz bus with 32.768 kHz Crystal)
- ☐ 5 V (1.049 MHz bus with 32.768 kHz Crystal)

Interrupt Trigger
- ☐ Edge-Sensitive
- ☐ Level- and Edge-Sensitive

Customer Name_____

Address_____

City_____ State_____ Zip_____

Phone (___)_____ Extension_____

Contact Ms/Mr_____

Customer Part Number_____

Pattern Media
- ☐ 2532 EPROM
- ☐ 2716 EPROM
- ☐ MDOS Disk File
- ☐ (Note)_____

NOTE: Other media require prior factory approval.

Signature_____

Title_____

Advance Information

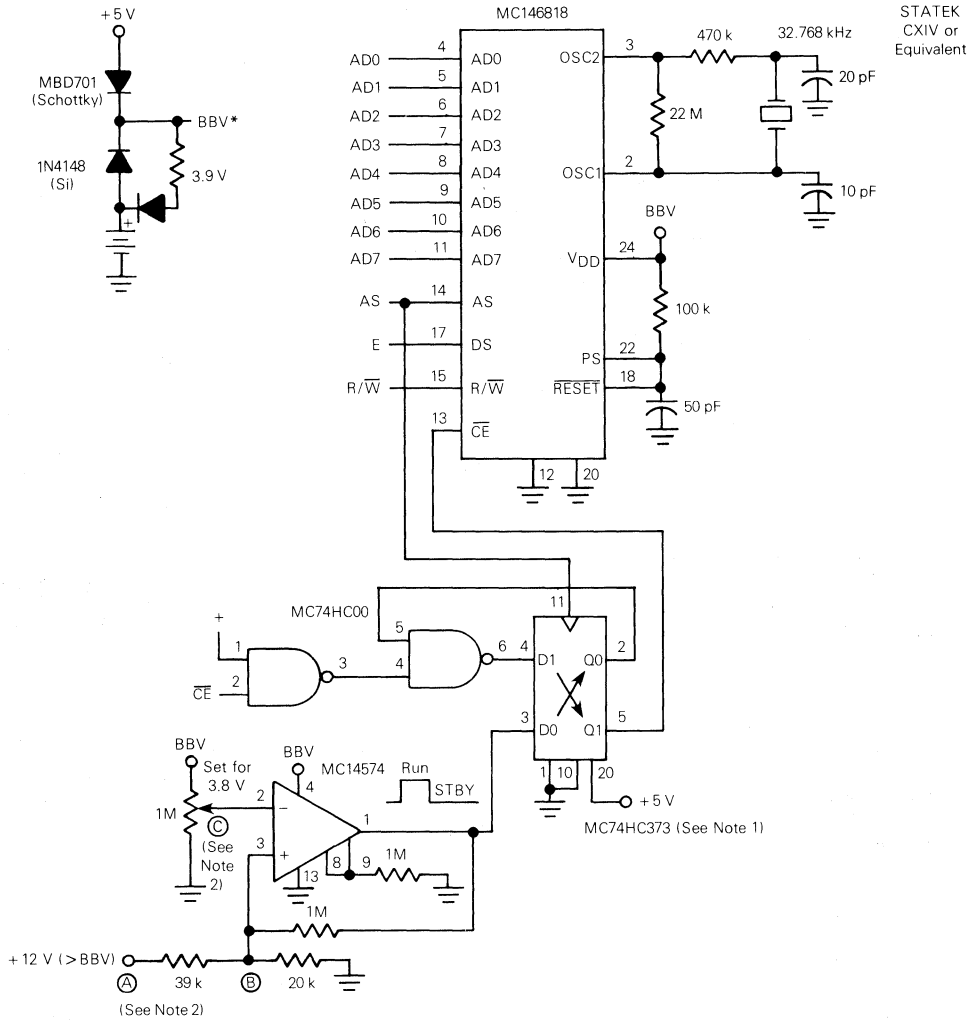
3

REAL-TIME CLOCK PLUS RAM (RTC)
Advance Information Data Sheet
ADI-856-R1

The following information is an addition to **POWER-DOWN CONSIDERATIONS** found on page 11 of the MC146818 Advance Information Data Sheet (ADI-856-R1).

MC146818s with the date code of 3N46XXXX and GC6XXXX require a synchronization of the $\overline{CE}$ pin with address strobe. The following circuit will satisfy that condition, and also show a typical application of power-down circuitry.

If $\overline{CE}$ is grounded at all times (no power down required) the following circuit need not be used.



*BBV = Battery Backup Voltage

NOTES:
1. All unused inputs of the MC74HC373 must be grounded.
2. If point (A) equals 12 V point (B) should be equal to 4.06 V. If point (A) equals 10 V point (B) should be equal to 3.38 V with (C) set for 3.18 V.

MOTOROLA

MC146818

Advance Information

CMOS

(HIGH-PERFORMANCE
SILICON-GATE COMPLEMENTARY MOS)

REAL-TIME CLOCK
PLUS RAM

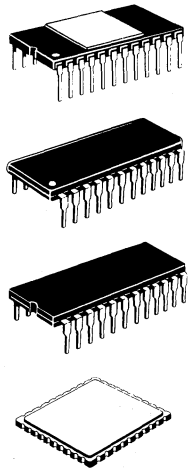
3

REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 μW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5 μs to 500 ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input
  - At Time Base Frequency ÷1 or ÷4
- 24-Pin Dual-In-Line Package
- Chip Carrier Also Available



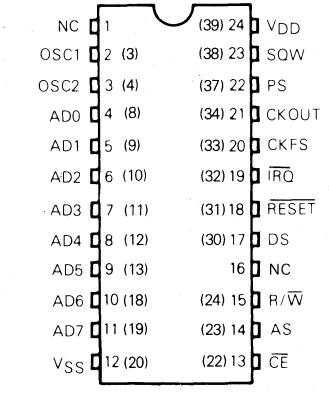
L SUFFIX
CERAMIC PACKAGE
CASE 716

P SUFFIX
PLASTIC PACKAGE
CASE 709

S SUFFIX
CERDIP PACKAGE
CASE 623

Z SUFFIX
CHIP CARRIER
CASE 761

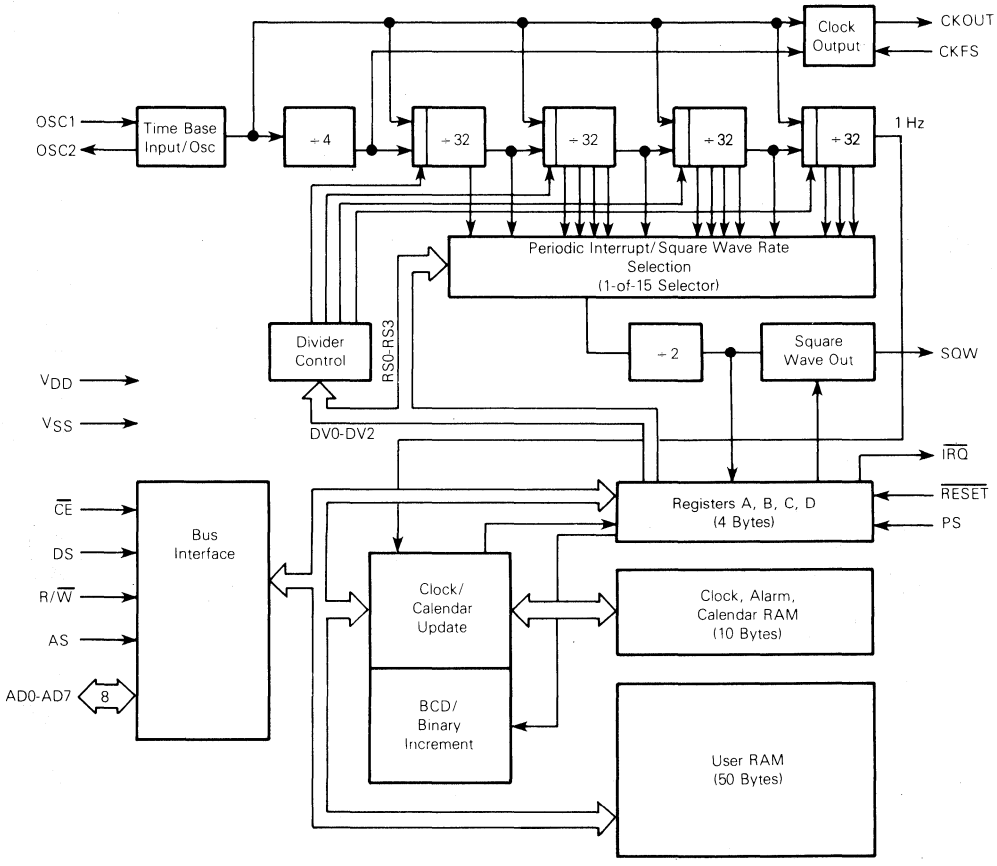
PIN ASSIGNMENT



Pin numbers in parentheses represent equivalent Z suffix chip carrier pins. Pins that have not been designated for the chip carrier are not connected.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — BLOCK DIAGRAM



MAXIMUM RATINGS (Voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	V_{SS} −0.5 to V_{DD} +0.5	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146818 MC146818C (V_{DD} = 3.0 to 5.5 V operation)	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic	θ_{JA}	 120 65 50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}	32.768	32.768	kHz
Output Voltage	V_{OL}	–	0.1	V
$I_{Load} < 10\ \mu A$	V_{OH}	$V_{DD} - 0.1$	–	
I_{DD} – Bus Idle CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{CE} = V_{DD} - 0.2$; C_L (OSC2) = 10 pF f_{osc} = 32.768 kHz	I_{DD3}	–	50	μA
I_{DD} – Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = $V_{DD} - 0.2$ V; No Clock	I_{DD4}	–	50	μA
Output High Voltage (I_{Load} = −0.25 mA, All Outputs)	V_{OH}	2.7	–	V
Output Low Voltage (I_{Load} = 0.25 mA, All Outputs)	V_{OL}	–	0.3	V
Input High Voltage AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, $\overline{RESET}$, CKFS, PS, OSC1	V_{IH}	2.1 2.5	V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.5	V
Input Current All Inputs	I_{in}	–	±1	μA
Three-State Leakage $\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	μA

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

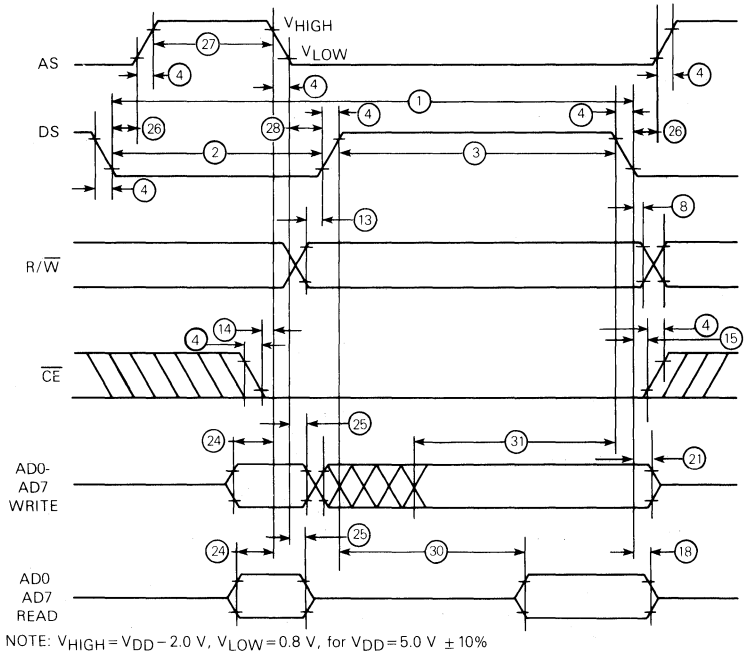
Characteristics	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}	32.768	4194.304	kHz
Output Voltage	V_{OL}	–	0.1	V
$I_{Load} < 10\ \mu A$	V_{OH}	$V_{DD} - 0.1$	–	
I_{DD} – Bus Idle (External Clock) CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{CE} = V_{DD} - 0.2$; C_L (OSC2) = 10 pF				
f_{osc} = 4.194304 MHz	I_{DD1}	–	3	mA
f_{osc} = 1.048516 MHz	I_{DD2}	–	800	μA
f_{osc} = 32.768 kHz	I_{DD3}	–	50	μA
I_{DD} – Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = $V_{DD} - 0.2$ V; No Clock	I_{DD4}	–	50	μA
Output High Voltage (I_{Load} = −1.6 mA, AD0-AD7, CKOUT) (I_{Load} = −1.0 mA, SQW)	V_{OH}	4.1	–	V
Output Low Voltage (I_{Load} = 1.6 mA, AD0-AD7, CKOUT) (I_{Load} = 1.0 mA, $\overline{IRQ}$ and SQW)	V_{OL}	–	0.4	V
Input High Voltage CKFS, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, PS	V_{IH}	$V_{DD} - 2.0$	V_{DD}	V
$\overline{RESET}$		$V_{DD} - 0.8$	V_{DD}	
OSC1		$V_{DD} - 1.0$	V_{DD}	
Input Low Voltage AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$	V_{IL}	V_{SS}	0.8	V
CKFS, PS, $\overline{RESET}$		V_{SS}	0.8	
OSC1		V_{SS}	0.8	
Input Current All Inputs	I_{in}	–	±1	μA
Three-State Leakage $\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	μA

BUS TIMING

Ident. Number	Characteristics	Symbol	V_{DD} = 3.0 V 50 pF Load		V_{DD} = 5.0 V ± 10% 2 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	–	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	1000	–	300	–	ns
3	Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	1500	–	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	10	–	ns
13	R/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	–	80	–	ns
14	Chip Enable Setup Time Before AS/ALE Fall	t_{CS}	200	*	55	*	ns
15	Chip Enable Hold Time	t_{CH}	10	–	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	–	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	–	50	–	ns
25	Muxed Address Hold Time	t_{AHL}	100	–	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	–	50	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	–	135	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	–	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	1300	–	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	1500	–	200	–	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.
*Refer to **IMPORTANT NOTICES** appearing on page 20 of this data sheet.

FIGURE 2 — MC146818 BUS TIMING



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

FIGURE 3 — BUS READ TIMING COMPETITOR MULTIPLEXED BUS

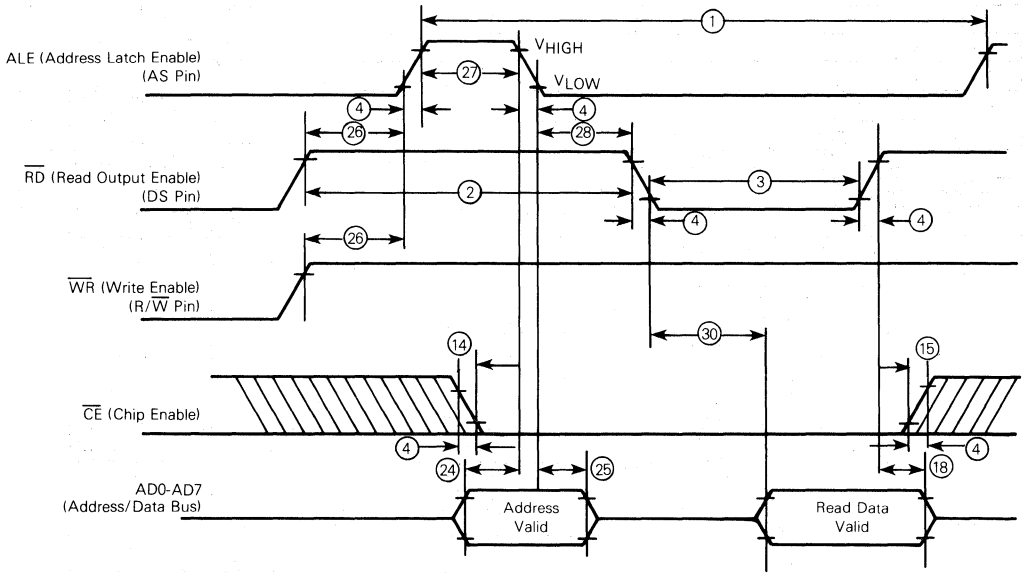
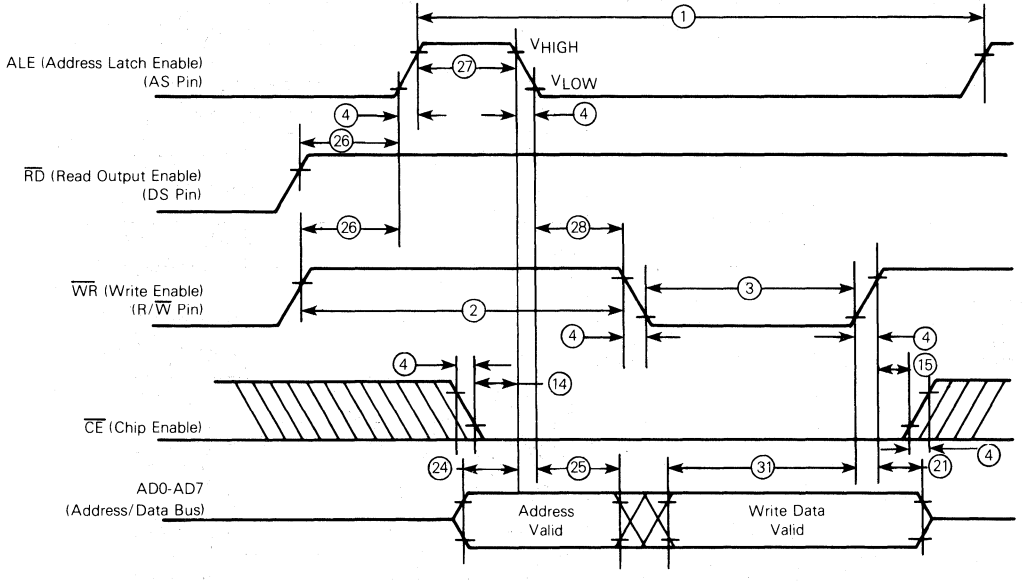


FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS



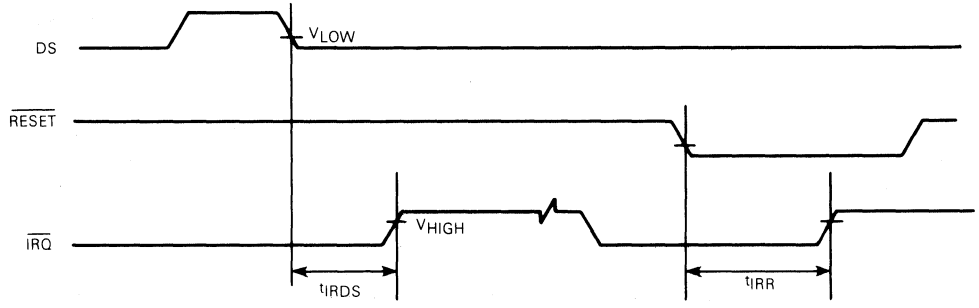
NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

TABLE 1 — SWITCHING CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Description	Symbol	Min	Max	Unit
Oscillator Startup	t_{RC}	—	100	ms
Reset Pulse Width	t_{RWL}	5	—	µs
Reset Delay Time	t_{RLH}	5	—	µs
Power Sense Pulse Width	t_{PWL}	5	—	µs
Power Sense Delay Time	t_{PLH}	5	—	µs
$\overline{IRQ}$ Release from DS	t_{IRDS}	—	2	µs
$\overline{IRQ}$ Release from $\overline{RESET}$	t_{IRR}	—	2	µs
VRT Bit Delay	t_{VRTD}	—	2	µs

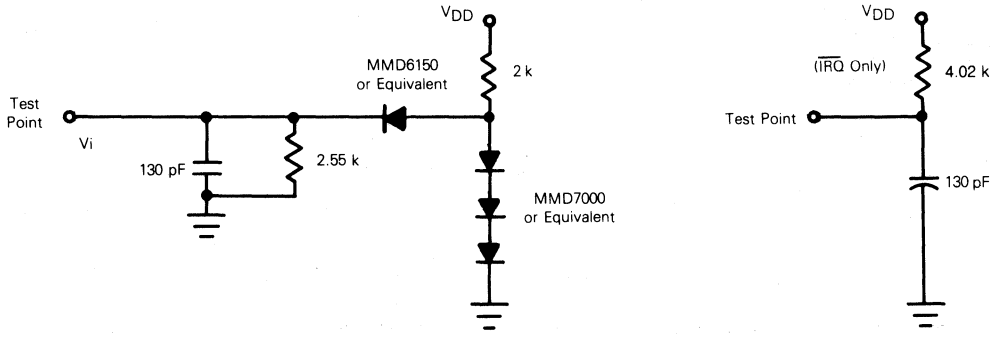
3

FIGURE 5 — $\overline{IRQ}$ RELEASE DELAY



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

FIGURE 6 — TTL EQUIVALENT TEST LOAD



All Outputs Except OSC2 (See Figure 10)

FIGURE 7 — POWER-UP

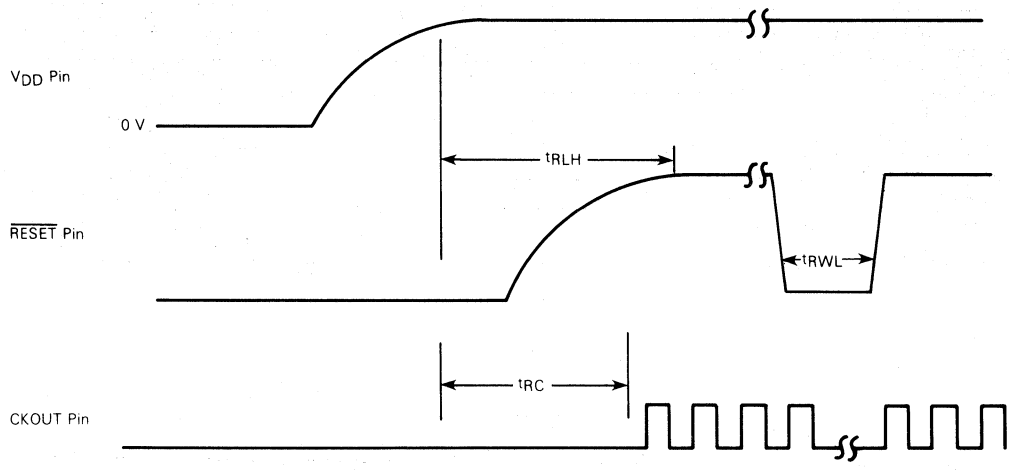
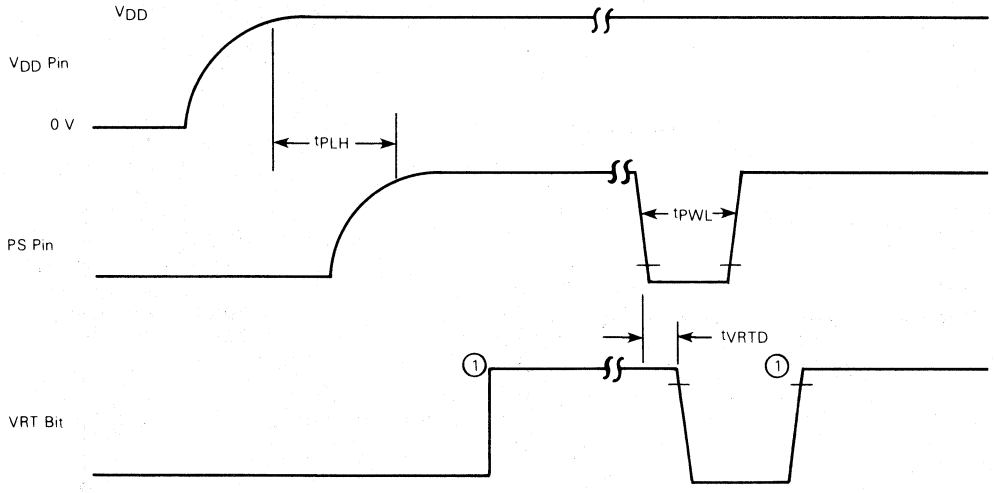


FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT



(1) The VRT bit is set to a "1" by reading Register d. The VRT bit can only be cleared by pulling the PS pin low (see **REGISTER D ($0D)**).

MOTEL

The MOTEL circuit is a new concept that permits the MC146818 to be directly interfaced with many types of microprocessors. No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

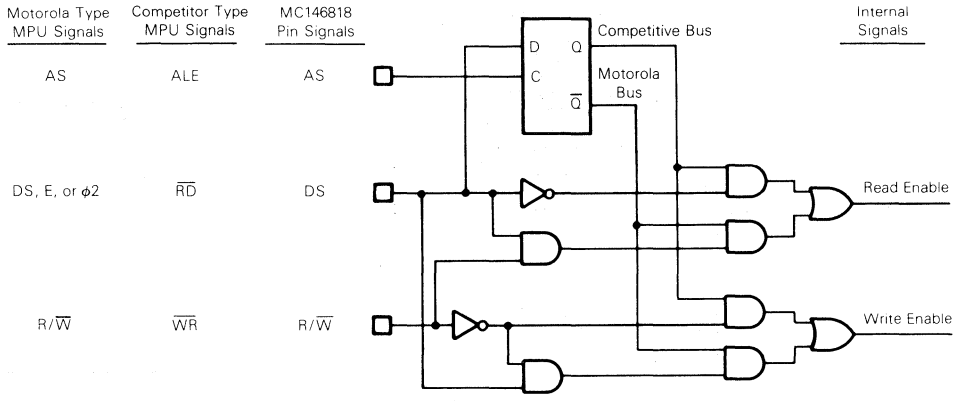
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated by the Motorola MC6800 and the other by the Intel 8080 and its companion part, the 8228.

The MOTEL circuit (for MOTorola and IntEL bus compatibility) is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry standard bus structure is now available. The MOTEL concept is shown logically in Figure 9.

MOTEL selects one of two interpretations of two pins. In the Motorola case, DS and R/$\overline{W}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of R/$\overline{W}$. With competitor buses, the inversion of $\overline{RD}$ and $\overline{WR}$ create functionally identical internal read and write enable signals.

The MC146818 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{RD}$ pin. Since DS is always low and $\overline{RD}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

FIGURE 9 – FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the MC146818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins, V_{DD} being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

OSC1, OSC2 – TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 10. The internal time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz or 1.048576 MHz frequencies. The crystal connections are shown in Figure 11 and the crystal characteristics in Figure 12.

CKOUT – CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS – CLOCK OUT FREQUENCY SELECT, INPUT

When the CKFS pin is tied to V_{DD} it causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is tied to V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

FIGURE 10 — EXTERNAL TIME-BASE CONNECTION

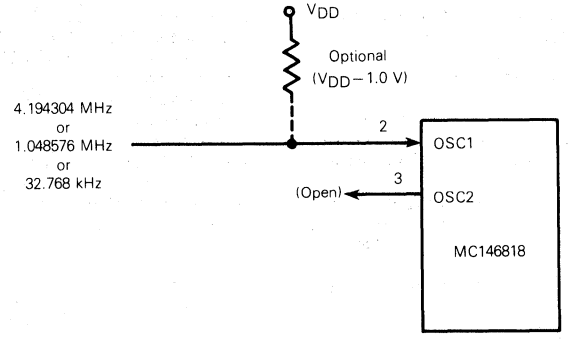
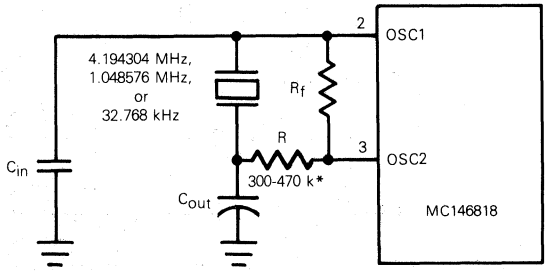
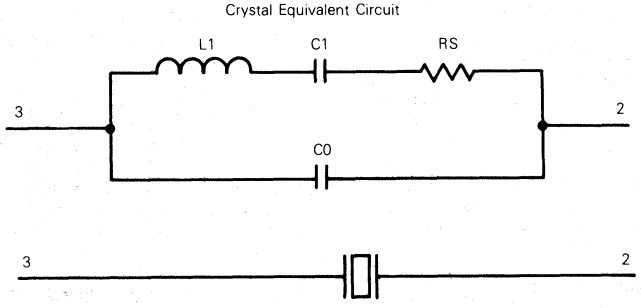


FIGURE 11 — CRYSTAL OSCILLATOR CONNECTION



*32.768 kHz Only — Consult Crystal Manufacturer's Specification

FIGURE 12 — CRYSTAL PARAMETERS



f_{osc}	4.194304 MHz	1.048576 MHz	32.768 kHz
RS (Maximum)	75 Ω	700 Ω	50 k
C0 (Maximum)	7 pF	5 pF	1.7 pF
C1	0.012 pF	0.008 pF	0.003 pF
Q	50 k	35 k	30 k
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
R	—	—	300-470 k
R_f	10 M	10 M	22 M

TABLE 2 — CLOCK OUTPUT FREQUENCIES

Time Base (OSC1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4.194304 MHz	High	4.194304 MHz
4.194304 MHz	Low	1.048576 MHz
1.048576 MHz	High	1.048576 MHz
1.048576 MHz	Low	262.144 kHz
32.768 kHz	High	32.768 kHz
32.768 kHz	Low	8.192 kHz

SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages. The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B.

AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the MC146818 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the MC146818 latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146818 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to the high-impedance state) when DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case.

AS — MULTIPLEXED ADDRESS STROBE, INPUT

A positive going multiplexed address strobe pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the address to be latched within the MC146818. The automatic MOTEL circuit in the MC146818 also latches the state of the DS pin with the falling edge of AS or ALE.

DS — DATA STROBE OR READ, INPUT

The DS pin has two interpretations via the MOTEL circuit. When emanating from a Motorola type processor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), and $\phi 2$ ($\phi 2$ clock). During read cycles, DS signifies the time that the RTC is to drive the bidirectional bus. In write cycles, the trailing edge of DS causes the Real-Time Clock plus RAM to latch the written data.

The second MOTEL interpretation of DS is that of $\overline{RD}$, $\overline{MEMR}$, or $\overline{I/OR}$ emanating from the competitor type processor. In this case, DS identifies the time period when the real-time clock plus RAM drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the MC146818, latches the state of the DS pin on the falling edge of AS/ALE. When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus processors. To ensure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

R/$\overline{W}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{W}$ pin in one of two ways. When a Motorola type processor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ during DS

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor type processors. The MOTEL circuit in this mode gives R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

$\overline{CE}$ — CHIP ENABLE, INPUT

The chip-enable ($\overline{CE}$) signal must be asserted (low) for a bus cycle in which the MC146818 is to be accessed. $\overline{CE}$ is not latched and must be stable during DS and AS (Motorola case of MOTEL) and during $\overline{RD}$ and $\overline{WR}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the MC146818. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is high, all address, data, DS, and R/$\overline{W}$ inputs from the processor are disconnected within the MC146818. This permits the MC146818 to be isolated from a powered-down processor. When $\overline{CE}$ is held high, an unpowered device cannot receive power through the input pins from the real-time clock power source. Battery power consumption can thus be reduced by using a pullup resistor or active clamp on $\overline{CE}$ when the main power is off. When $\overline{CE}$ is not used, it should be grounded.

$\overline{IRQ}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{IRQ}$ pin is an active low output of the MC146818 that may be used as an interrupt input to a processor. The $\overline{IRQ}$ output remains low as long as the status bit causing the interrupt is present and the corresponding interrupt-enable bit is set. To clear the $\overline{IRQ}$ pin, the processor program normally reads Register C. The $\overline{RESET}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{IRQ}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{IRQ}$ bus with one pullup at the processor.

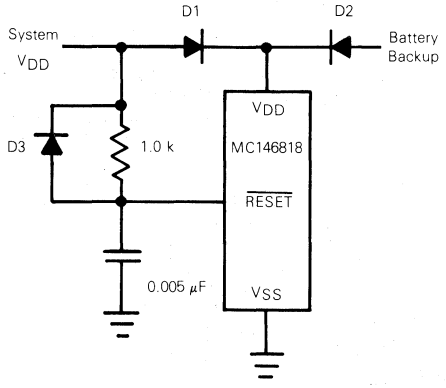
$\overline{RESET}$ — RESET, INPUT

The $\overline{RESET}$ pin does not affect the clock, calendar, or RAM functions. On powerup, the $\overline{RESET}$ pin must be held low for the specified time, t_{RLH}, in order to allow the power supply to stabilize. Figure 13 shows a typical representation of the $\overline{RESET}$ pin circuit.

When $\overline{RESET}$ is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Update ended Interrupt Enable (UIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible.

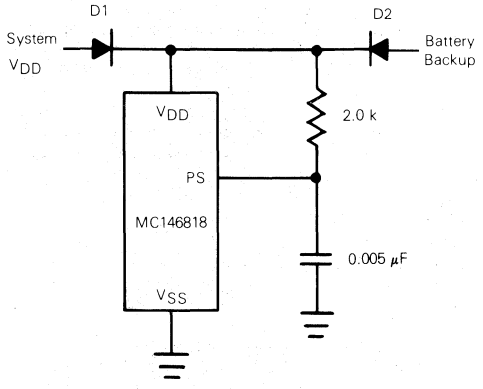
FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR $\overline{RESET}$



D1 = MBD701 (Schottky) or Equivalent
D2 = D3 = 1N4148 or Equivalent

Note: If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements.

FIGURE 14 — TYPICAL POWERUP DELAY CIRCUIT FOR POWER SENSE



D1 = MBD701 (Schottky) or Equivalent
D2 = 1N4148 or Equivalent

g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{IRQ}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to zero.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low the VRT bit is cleared to zero.

When using the VRT feature during powerup, the PS pin must be externally held low for the specified t_{PLH} time. As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. PS must go high after powerup to allow the VRT bit to be set by a read of register D.

POWER-DOWN CONSIDERATIONS

In most systems, the MC146818 must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The chip enable ($\overline{CE}$) pin controls all bus inputs (R/$\overline{W}$, DS, AS, AD0-AD7). $\overline{CE}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{CE}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded. Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.

ADDRESS MAP

Figure 15 shows the address map of the MC146818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except for the following: 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only. The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program. Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 μs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 μs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

FIGURE 15 — ADDRESS MAP

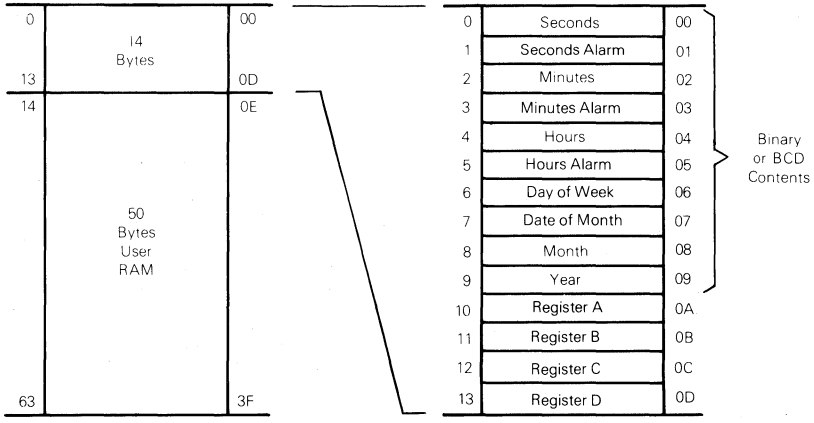


TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

Address Location	Function	Decimal Range	Range		Example*	
			Binary Data Mode	BCD Data Mode	Binary Data Mode	BCD Data Mode
0	Seconds	0-59	$00-$3B	$00-$59	15	21
1	Seconds Alarm	0-59	$00-$3B	$00-$59	15	21
2	Minutes	0-59	$00-$3B	$00-$59	3A	58
3	Minutes Alarm	0-59	$00-$3B	$00-$59	3A	58
4	Hours (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	$00-$17	$00-$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	$01-$0C (AM) and $81-$8C (PM)	$01-$12 (AM) and $81-$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	$00-$17	$00-23	05	05
6	Day of the Week Sunday = 1	1-7	$01-$07	$01-$07	05	05
7	Date of the Month	1-31	$01-$1F	$01-$31	0F	15
8	Month	1-12	$01-$0C	$01-$12	02	02
9	Year	0-99	$00-$63	$00-$99	4F	79

*Example: 5:58:21 Thursday 15 February 1979 (time is AM)

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the MC146818. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional MC146818s may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. The high-order bit of the seconds byte, bit 7 of Register A, and all bits of Registers C and D cannot effectively be used as general purpose RAM.

INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{IRQ}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle. One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{IRQ}$ pin is asserted low. $\overline{IRQ}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{IRQ}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

DIVIDER STAGES

The MC146818 has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controller by three divider bus (DV2, DV1, and DV0) in Register A.

DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held reset, which allows precision setting of the time. When the divider is changed from reset to an operating time base, the first update cycle is one-half second later. The divider-control bits are also used to facilitate testing the MC146818.

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A			Operation Mode	Divider Reset	Bypass First N-Divider Bits
	DV2	DV1	DV0			
4.194304 MHz	0	0	0	Yes	–	N = 0
1.048576 MHz	0	0	1	Yes	–	N = 2
32.768 kHz	0	1	0	Yes	–	N = 7
Any	1	1	0	No	Yes	–
Any	1	1	1	No	Yes	–

Note: Other combinations of divider bits are used for test purposes only.

SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B. Altering the divider, square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 μs. The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit in Register B. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial receive bits or bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Select Bits Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate tPI	SQW Output Frequency	Periodic Interrupt Rate tPI	SQW Output Frequency
0	0	0	0	None	None	None	None
0	0	0	1	30.517 μs	32.768 kHz	3.90625 ms	256 Hz
0	0	1	0	61.035 μs	16.384 kHz	7.8125 ms	128 Hz
0	0	1	1	122.070 μs	8.192 kHz	122.070 μs	8.192 kHz
0	1	0	0	244.141 μs	4.096 kHz	244.141 μs	4.096 kHz
0	1	0	1	488.281 μs	2.048 kHz	488.281 μs	2.048 kHz
0	1	1	0	976.562 μs	1.024 kHz	976.562 μs	1.024 kHz
0	1	1	1	1.953125 ms	512 Hz	1.953125 ms	512 Hz
1	0	0	0	3.90625 ms	256 Hz	3.90625 ms	256 Hz
1	0	0	1	7.8125 ms	128 Hz	7.8125 ms	128 Hz
1	0	1	0	15.625 ms	64 Hz	15.625 ms	64 Hz
1	0	1	1	31.25 ms	32 Hz	31.25 ms	32 Hz
1	1	0	0	62.5 ms	16 Hz	62.5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

UPDATE CYCLE

The MC146818 executes an update cycle once-per-second, assuming one of the proper time bases is in place, the DV0-DV2 divider is not clear, and the SET bit in Register B is clear. The SET bit in the "1" state permits the program to initialize the time and calendar bytes by stopping an existing update and preventing a new one from occurring.

The primary function of the update cycle is to increment the seconds byte, check for overflow, increment the minutes byte when appropriate and so forth through to the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match or if a "don't care" code (11XXXXXX) is present in all three positions.

With a 4.194304 MHz or 1.048576 MHz time base the update cycle takes 248 μs while a 32.768 kHz time base update cycle takes 1984 μs. During the update cycle, the time, calendar, and alarm bytes are not accessable by the processor program. The MC146818 protects the program from reading transitional data. This protection is provided by switching the time, calendar, and alarm portion of the RAM off the microprocessor bus during the entire update cycle. If the processor reads these RAM locations before the update is complete the output will be undefined. The update in progress (UIP) status bit is set during the interval.

A program which randomly accesses the time and date information finds data unavailable statistically once every 4032 attempts. Three methods of accommodating nonavailability during update are usable by the program. In discussing the three methods it is assumed that at random points user programs are able to call a subroutine to obtain the time of day.

The first method of avoiding the update cycle uses the update-ended interrupt. If enabled, an interrupt occurs after every update cycle which indicates that over 999 ms are available to read valid time and date information. During this time a display could be updated or the information could be transfered to continuously available RAM. Before leaving the interrupt service routine, the IRQF bit in Register C should be cleared.

The second method uses the update-in-progress bit (UIP) in Register A to determine if the update cycle is in progress or not. The UIP bit will pulse once-per-second. Statistically, the UIP bit will indicate that time and date information is unavailable once every 2032 attempts. After the UIP bit goes high, the update cycle begins 244 μs later. Therefore, if a low is read on the UIP bit, the user has at least 244 μs before the time/calendar data will be changed. If a "1" is read in the UIP bit, the time/calendar data may not be valid. The user should avoid interrupt service routines that would cause the time needed to read valid time/calendar data to exceed 244 μs.

The third method uses a periodic interrupt to determine if an update cycle is in progress. The UIP bit in Register A is set high between the setting of the PF bit in Register C (see Figure 16). Periodic interrupts that occur at a rate of greater than $t_{BUC} + t_{UC}$ allow valid time and date information to be read at each occurrence of the periodic interrupt. The reads should be completed within $(T_{PI} \div 2) + t_{BUC}$ to ensure that data is not read during the update cycle.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

REGISTERS

The MC146818 has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

REGISTER A ($0A)

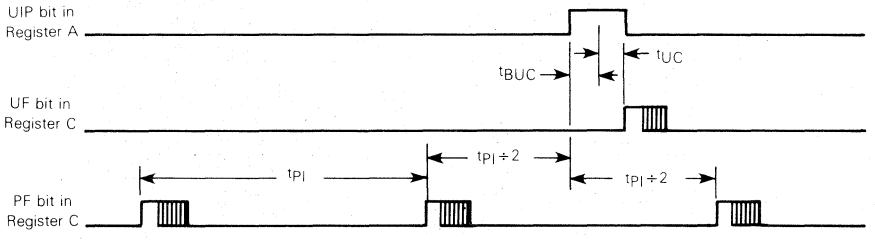
MSB							LSB	Read/Write
b7	b6	b5	b4	b3	b2	b1	b0	Register
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0	except UIP

UIP – The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin. When UIP is a "0" the update cycle is not in progress and will not be for at least 244 μs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero – it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibit any update cycle and then clear the UIP status bit.

TABLE 6 – UPDATE CYCLE TIMES

UIP Bit	Time Base (OSC1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4.194304 MHz	248 μs	–
1	1.048576 MHz	248 μs	–
1	32.768 kHz	1984 μs	–
0	4.194304 MHz	–	244 μs
0	1.048576 MHz	–	244 μs
0	32.768 kHz	–	244 μs

FIGURE 16 – UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIPS



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc. per Table 5)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

DV2, DV1, DV0 – Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

RS3, RS2, RS1, RS0 – The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{\text{RESET}}$.

REGISTER B ($0B)

MSB LSB — Read/Write Register

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

SET – When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is a read/write bit which is not modified by $\overline{\text{RESET}}$ or internal functions of the MC146818.

PIE – The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{\text{IRQ}}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{\text{IRQ}}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal MC146818 functions, but is cleared to "0" by a $\overline{\text{RESET}}$.

AIE – The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) bit in Register C to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code of binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{\text{IRQ}}$ signal. The $\overline{\text{RESET}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE – The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flag (UF) bit in Register C to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

SQWE – When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

DM – The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12 – The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

DSE – The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

REGISTER C ($0C)

MSB LSB — Read-Only Register

b7	b6	b5	b4	b3	b	b1	b0
IRQF	PF	AF	UF	0	0	0	0

IRQF – The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"
AF = AIE = "1"
UF = UIE = "1"

i.e., IRQF = PF•PIE + AF•AIE + UF•UIE

Any time the IRQF bit is a "1", the $\overline{\text{IRQ}}$ pin is driven low. All flag bits are cleared after Register C is read by the program or when the $\overline{\text{RESET}}$ pin is low.

PF – The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{\text{IRQ}}$ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a $\overline{\text{RESET}}$ or a software read of Register C.

AF – A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{\text{IRQ}}$ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A $\overline{\text{RESET}}$ or a read of Register C clears AF.

UF – The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{\text{IRQ}}$. UF is cleared by a Register C read or a $\overline{\text{RESET}}$.

b3 TO b0 – The unused bits of Status Register 1 are read as "0's". They can not be written.

REGISTER D ($0D)

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read Only
VRT	0	0	0	0	0	0	0	Register

VRT – The valid RAM and time (VRT) bit indicates the condition of the contents of the RAM, provided the power sense (PS) pin is satisfactorily connected. A "0" appears in the VRT bit when the power-sense pin is low. The processor program can set the VRT bit when the time and calendar are initialized to indicate that the RAM and time are valid. The VRT is a read only bit which is not modified by the $\overline{\text{RESET}}$ pin. The VRT bit can only be set by reading Register D.

b6 TO b0 – The remaining bits of Register D are unused. They cannot be written, but are always read as "0's."

TYPICAL INTERFACING

The MC146818 is best suited for use with microprocessors which generate an address-then-data multiplexed bus. Figures 17 and 18 show typical interfaces to bus-compatible processors. These interfaces assume that the address decoding can be done quickly. However, if standard metal-gate CMOS gates are used the $\overline{\text{CE}}$ setup time may be violated. Figure 19 illustrates an alternative method of chip selection which will accommodate such slower decoding.

The MC146818 can be interfaced to single-chip microcomputers (MCU) by using eleven port lines as shown in Figure 20. Non-multiplexed bus microprocessors can be interfaced with additional support.

There is one method of using the multiplexed bus MC146818 with non-multiplexed bus processors. The interface uses available bus control signals to multiplex the address and data bus together.

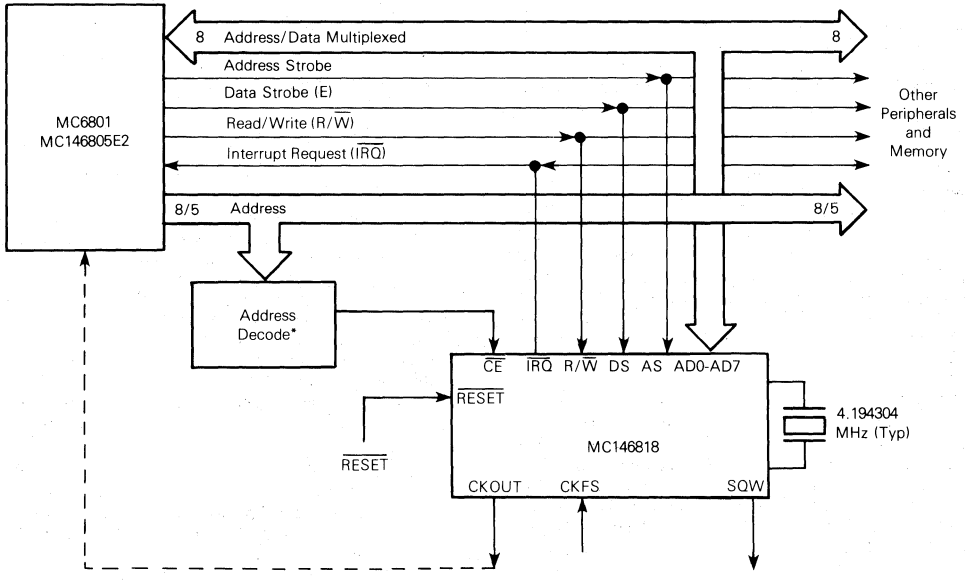
An example using either the Motorola MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 21.

Figure 22 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data:

Accumulator A: The address of the RTC to be accessed.
Accumulator B: Write: The data to be written.
Read: The data read from the RTC.

The RTC is mapped to two consecutive memory locations – RTC and RTC + 1 as shown in Figure 21.

FIGURE 17 – MC146818 INTERFACED WITH MOTOROLA COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS



*High-Speed Silicon-Gate CMOS or TTL Address Decoding

FIGURE 18 — MC146818 INTERFACED WITH
COMPETITOR COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

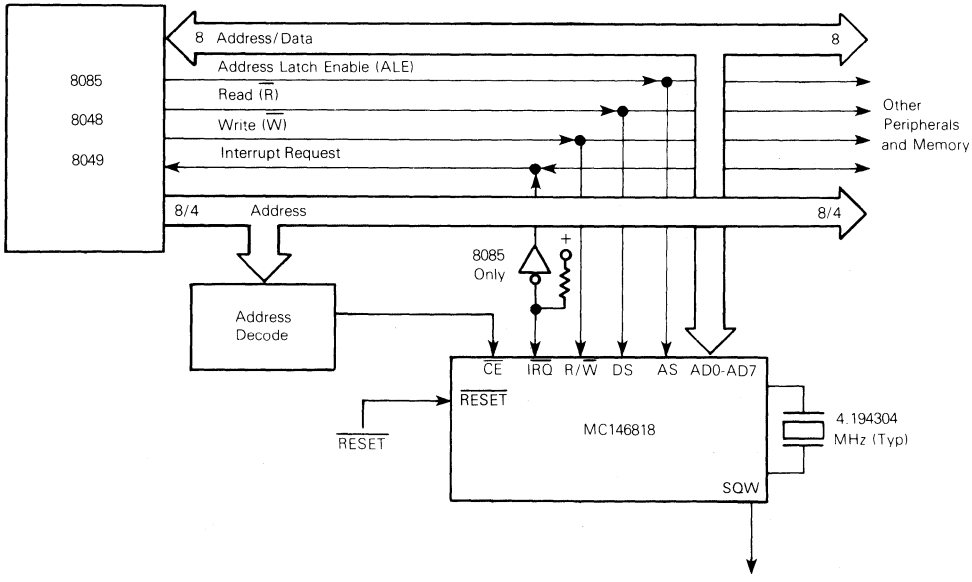


FIGURE 19 — MC146818 INTERFACE WITH MC146805E2
CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING

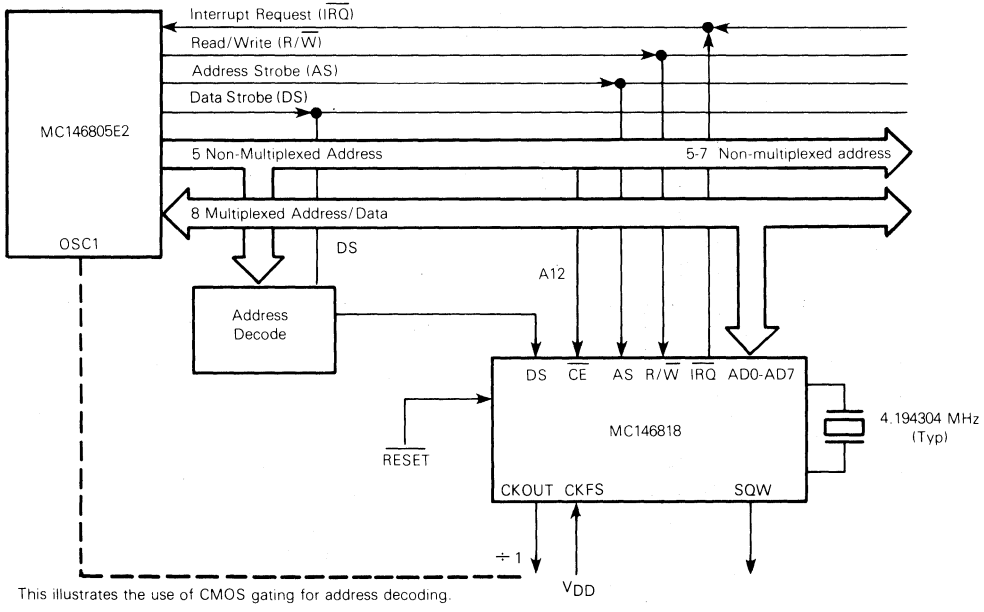


FIGURE 20 — MC146818 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER

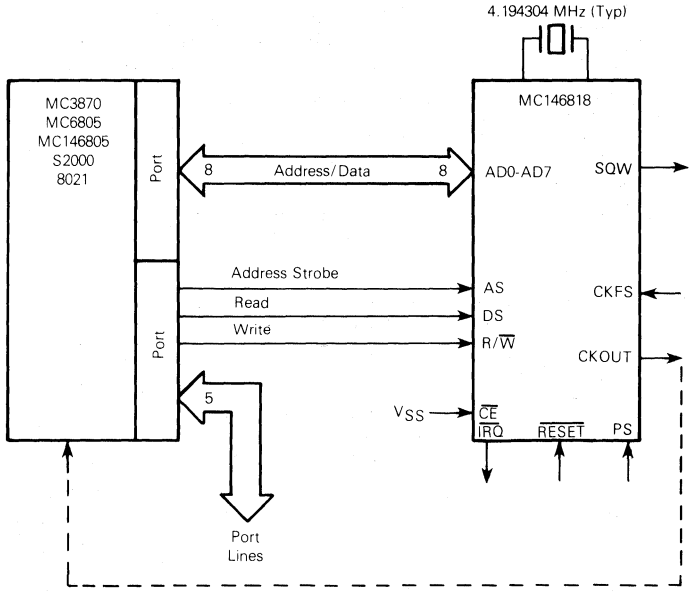


FIGURE 21 — MC146818 INTERFACED WITH MOTOROLA PROCESSORS

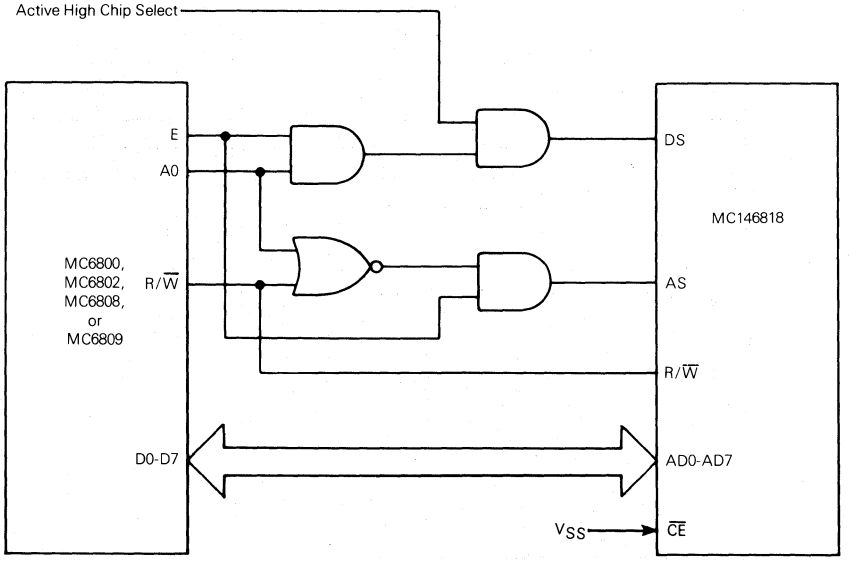


FIGURE 22 — SUBROUTINE FOR READING AND WRITING THE MC146818 WITH A NON-MULTIPLEXED BUS

```
READ    STA     RTC     Generate AS and Latch Data from ACCA
        LDAB    RTC+1   Generate DS and Get Data
        RTS

WRITE   STA     RTC     Generate AS and Latch Data from ACCA
        STAB    RTC+1   Generate DS and Store Data
        RTS
```

IMPORTANT NOTICES

Those devices made with date code 3N4GXXXX have the following exceptions when used in the Motorola mode of MOTEL.

1. V_{DD} = 3 to 5.25 V for operation
2. DS V_{IL} = 0.6 V Max.

The falling edge of chip select should occur during the active high pulse of address strobe, only on those units with date code GC6XXXX.

MC146823

Advance Information

CMOS
(HIGH-DENSITY HIGH-PERFORMANCE SILICON-GATE)

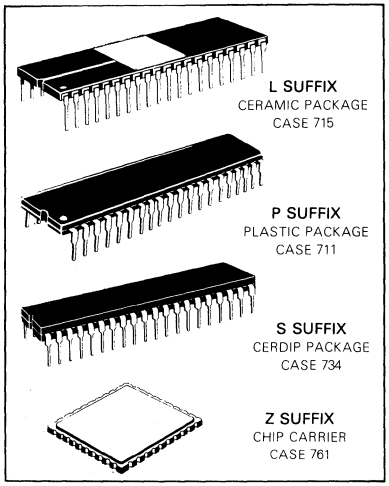
PARALLEL INTERFACE

CMOS PARALLEL INTERFACE

The MC146823 CMOS parallel interface (CPI) provides a universal means of interfacing external signals with the MC146805E2 CMOS microprocessor and other multiplexed bus microprocessors. The unique MOTEL circuit on-chip allows direct interfacing to most industry CMOS microprocessors, as well as many NMOS MPUs.

The MC146823 CPI includes three bidirectional 8-bit ports or 24 I/O pins. Each I/O line may be separately established as an input or an output under program control via data direction registers associated with each port. Using the bit change and test instructions of the MC146805E2, each individual I/O pin can be separately accessed. All port registers are read/write bytes to accommodate read-modify-write instructions. Features include:

- 24 Individually Programmed I/O Pins
- MOTEL Circuit for Bus Compatibility with Many Microprocessors
- Multiplexed Bus Compatibility with: MC146805E2, MC6801, MC6803, and Competitive Microprocessors
- Data Direction Registers for Ports A, B, and C
- Four Port C I/O Pins May Be Used as Control Lines for:
  - Four Interrupt Inputs
  - Input Byte Latch
  - Output Pulse
  - Handshake Activity
- 15 Registers Addressed as Memory Locations
- Handshake Control Logic for Input and Output Peripheral Operation
- Interrupt Output Pin
- Reset Input to Clear Interrupts and Initialize Internal Registers
- 3.0 Volt to 5.5 Volt Operating V_{DD}



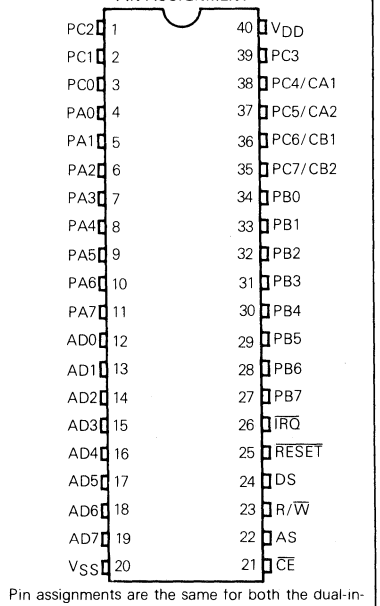
L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT



Pin assignments are the same for both the dual-in-line and chip carrier package.

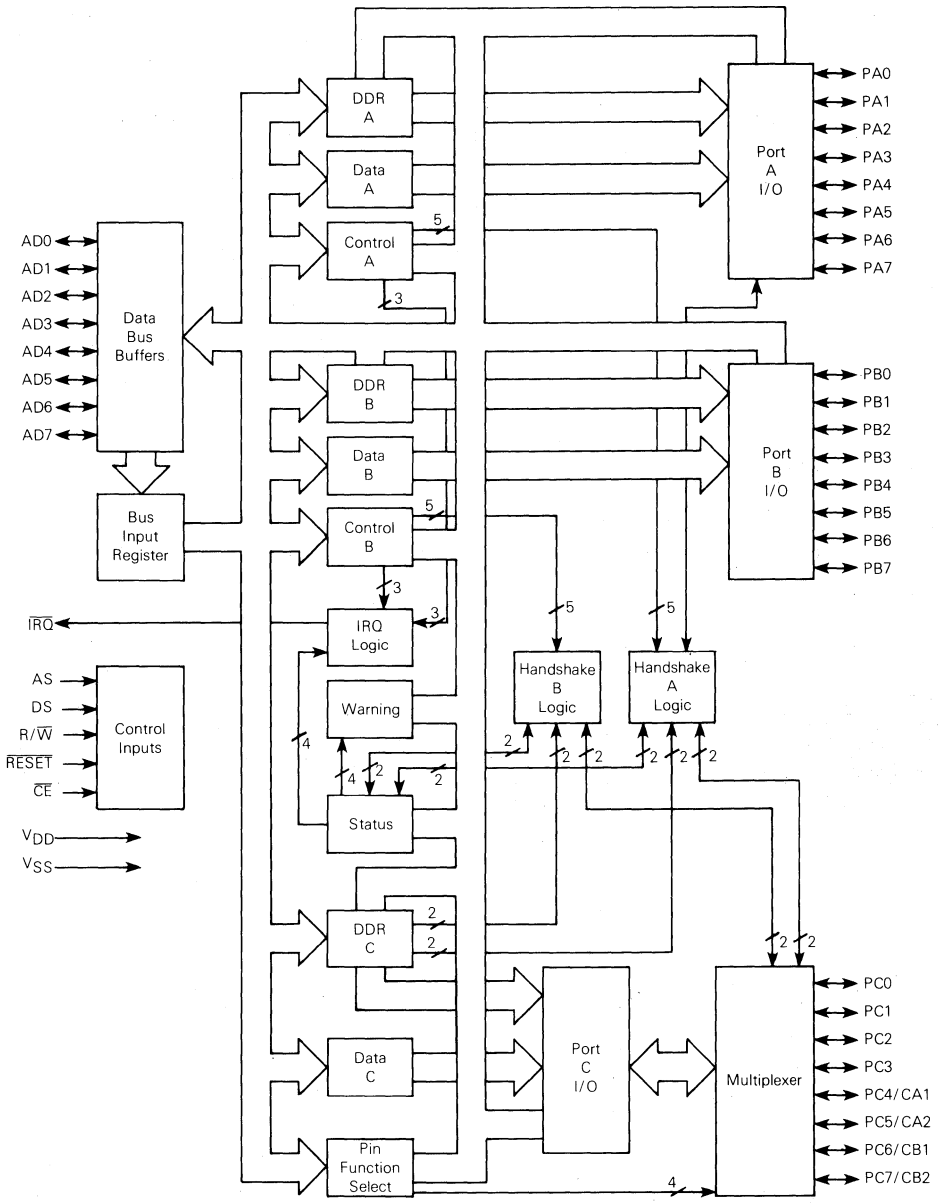
ORDERING INFORMATION

(T_A = 0°C to +70°C)

Package Type	Order Number
Plastic – P Suffix	MC146823P
Ceramic (Side Brazed) – L Suffix	MC146823L
Cerdip – S Suffix	MC146823S
Chip Carrier – Z Suffix	MC146823Z

This document contains information on a new product. Specifications and information herein are subject to change without notice.

BLOCK DIAGRAM



MAXIMUM RATINGS (Voltages reference to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \geq (V_{in}$ or $V_{out}) \geq V_{DD}$. Leakage currents are reduced and reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

THERMAL CHARACTERISTICS

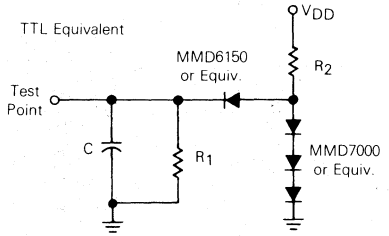
Characteristics	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Plastic		100	
Cerdip		60	
Chip Carrier		TBD	

3

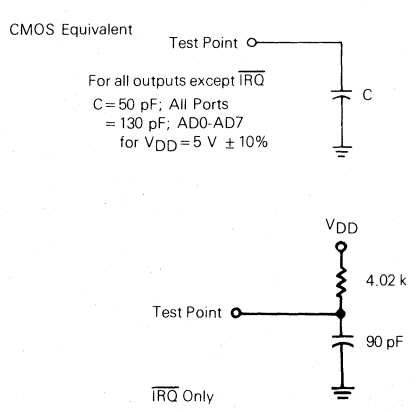
DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=0°C$ to 70°C, unless otherwise noted)

Parameter	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10\ \mu A$)	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Output High Voltage				
($I_{Load}=-1.6$ mA) AD0-AD7	V_{OH}	4.1	V_{DD}	
($I_{Load}=-0.2$ mA) PA0-PA7, PC0-PC7	V_{OH}	4.1	V_{DD}	V
($I_{Load}=-0.36$ mA) PB0-PB7	V_{OH}	4.1	V_{DD}	
Output Low Voltage				
($I_{Load}=1.6$ mA) AD0-AD7, PB0-PB7	V_{OL}	V_{SS}	0.4	
($I_{Load}=0.8$ mA) PA0-PA7, PC0-PC7	V_{OL}	V_{SS}	0.4	V
($I_{Load}=1.0$ mA) $\overline{IRQ}$	V_{OL}	V_{SS}	0.4	
Input High Voltage, AD0-AD7, AS, DS, R/$\overline{W}$, $\overline{CE}$, PA0-PA7, PB0-PB7, PC0-PC7	V_{IH}	$V_{DD}-2.0$	V_{DD}	V
$\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	V_{DD}	
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Quiescent Current – No dc Loads (All Ports Programmed as Inputs, All Inputs = V_{DD} – 0.2 V)	I_{DD}	–	160	μA
Total Supply Current (All Ports Programmed as Inputs, CE = V_{IL}, $t_{cyc}=1\ \mu s$)	I_{DD}	–	3.0	mA
Input Current, $\overline{CE}$, AS, R/$\overline{W}$, DS, $\overline{RESET}$	I_{in}	–	±1.0	μA
Hi-Z State Leakage, AD0-AD7, PA0-PA7, PB0-PB7, PC0-PC7	I_{TSL}	–	±10.0	μA

EQUIVALENT TEST LOADS



Pin	R1	R2	C
AD0-AD7	2.55k	2.0k	130 pF
PA0-PA7, PC0-PC7	20.0k	4.32k	50 pF
PB0-PB7	11.5k	2.1k	50 pF

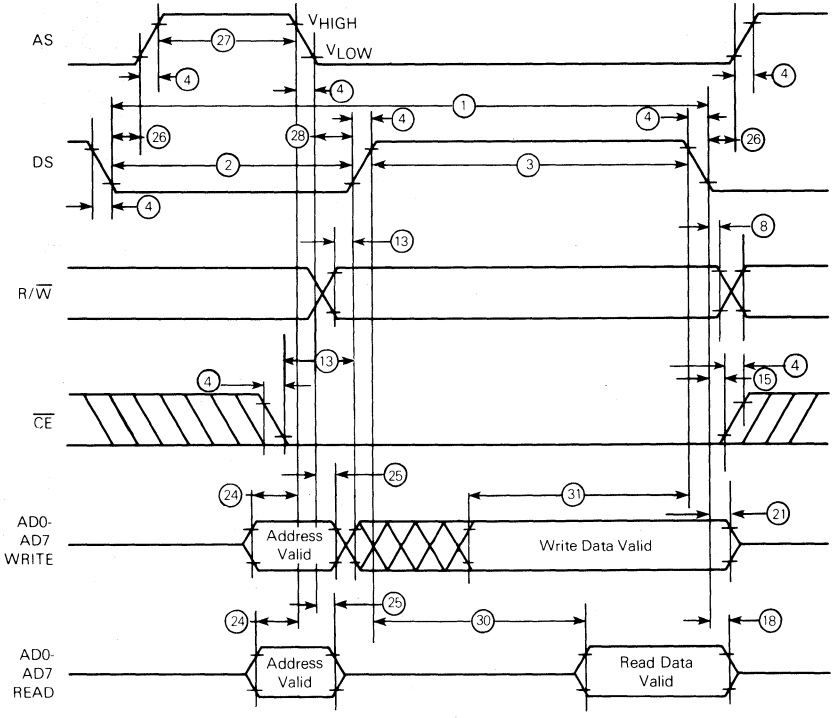


BUS TIMING (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Ident. Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1000	dc	ns
2	Pulse Width, $\overline{DS}$/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	300	–	ns
3	Pulse Width, $\overline{DS}$/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	ns
13	R/$\overline{W}$ and $\overline{CE}$ Setup Time Before $\overline{DS}$/E	t_{RWS}	25	–	ns
15	Chip Enable Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	100	ns
21	Write Data Hold Time	t_{DHW}	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	25	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	60	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	170	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	220	–	ns

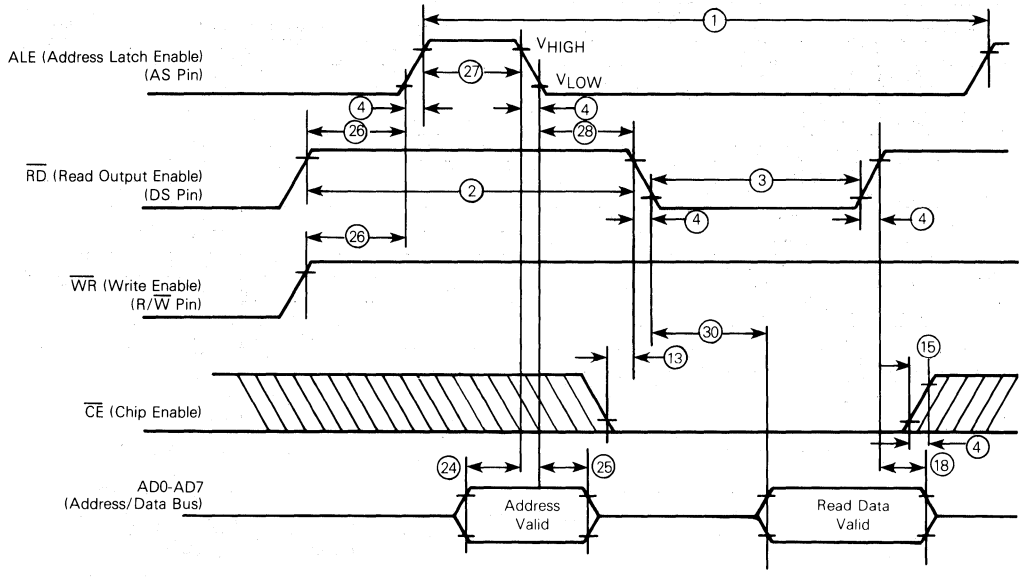
NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.

BUS TIMING DIAGRAM

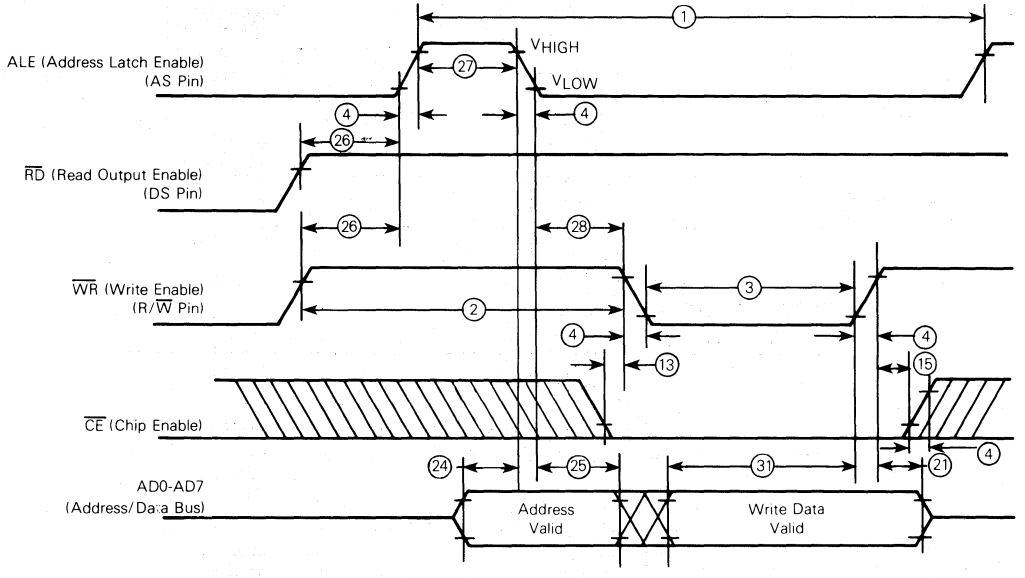


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

BUS READ TIMING COMPETITOR MULTIPLEXED BUS



BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V $\pm 10\%$

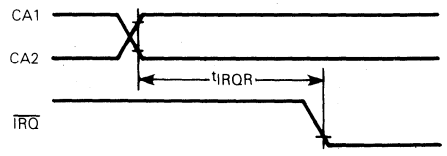
CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C)

Parameter	Symbol	Min	Max	Unit
Interrupt Response (Input Modes 1 and 3)	t_{IRQR}	TBD	–	μs
Delay, CA1 (CB1) Active Transition to CA2 (CB2) High (Output Mode 0)	t_{C2}	TBD	–	μs
Delay, CA2 Transition from Positive Edge of AS (Output Modes 0 and 1)	t_{A2}	TBD	–	μs
Delay, CB2 Transition from Negative Edge of AS (Output Modes 0 and 1)	t_{B2}	TBD	–	μs
CA2/CB2 Pulse Width (Output Mode 1)	t_{PW}	TBD	TBD	ns
Delay, V_{DD} Rise to $\overline{RESET}$ High	t_{RLH}	TBD	–	μs
Pulse Width, $\overline{RESET}$	t_{RW}	TBD	–	ns

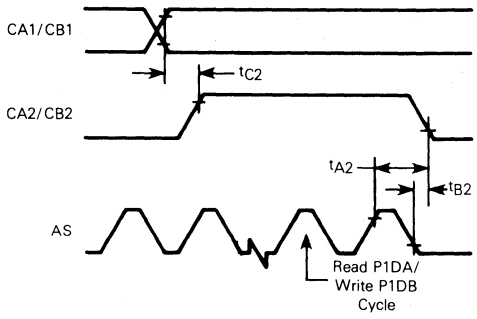
TBD = To be determined.

CONTROL TIMING DIAGRAMS

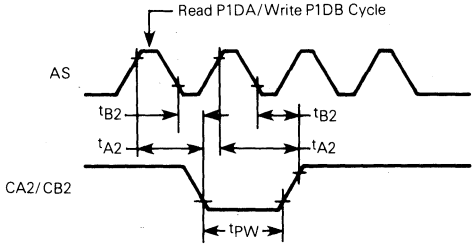
$\overline{IRQ}$ RESPONSE (INPUT MODES 1 AND 3)

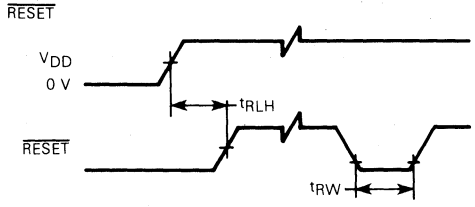


CA2/CB2 DELAY (OUTPUT MODE 0)



CA2/CB2 DELAY (OUTPUT MODE 1)





GENERAL DESCRIPTION

The MC146823, CMOS parallel interface (CPI), contains 24 individual bidirectional I/O lines configured in three 8-bit ports. The 15 internal registers, which control the mode of operation and contain the status of the port pins, are accessed via an 8-bit multiplexed address/data bus. The lower four address bits (AD0-AD3) of the multiplexed address bus determine which register is to be accessed (see Figure 1). The four address bits (AD4, AD5, AD6, and AD7) must be separately decoded to position this memory map within each 256 byte address space available via the 8-bit multiplexed address bus. For more detailed information refer to **REGISTER DESCRIPTION.**

FIGURE 1 — REGISTER ADDRESS MAP

0	Port A Data, Clear CA1 Interrupt	P1DA
1	Port A Data, Clear CA2 Interrupt	P2DA
2	Port A Data	PDA
3	Port B Data	PDB
4	Port C Data	PDC
5	Not Used	—
6	Data Direction Register for Port A	DDRA
7	Data Direction Register for Port B	DDRB
8	Data Direction Register for Port C	DDRC
9	Control Register for Port A	CRA
A	Control Register for Port B	CRB
B	Pin Function Select Register for Port C	FSR
C	Port B Data, Clear CB1 Interrupt	P1DB
D	Port B Data, Clear CB2 Interrupt	P2DB
E	Handshake/Interrupt Status Register	HSR
F	Handshake Over-Run Warning Register	HWR

The CPI is implemented with the MOTEL circuit which allows direct interface with either of the two major multiplexed microprocessor bus types. A detailed description of the MOTEL circuit is provided in the **MOTEL** section.

Three data direction registers (DDRs), one for each port, determine which pins are outputs and which are inputs. A logic zero on a DDR bit configures its associated pin as an input; and a logic one configures the pin as an output. Upon reset, the DDRs are cleared to logic zero to configure all port pins as inputs.

Actual port data may be read or written via the port data registers (PDA, PDB, and PDC). Ports A and B each have two additional data registers (P1DA and P2DA — P1DB and P2DB) which are used to clear the associated handshake/interrupt status register bits (HSA1 and HSA2 — HSB1 and HSB2), respectively. Port A may also be configured as an 8-bit latch when used with CA1. Reset has no effect on the contents of the port data registers. Users are advised to initialize the port data registers before changing any port pin to an output.

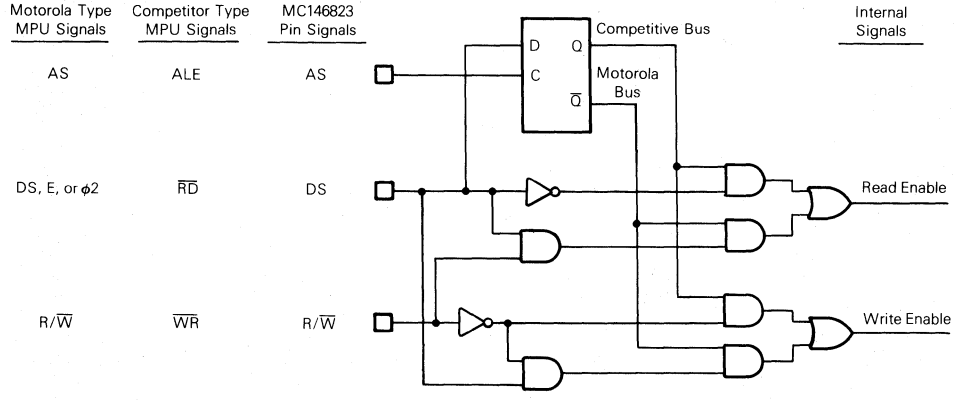
Four pins on port C (PC4/CA1, PC5/CA2, PC6/CB1, and PC7/CB2) may additionally be programmed as handshake lines for ports A and B via the port C function select register (FSR). Both ports A and B have one input-only line and one bidirectional handshake line each associated with them. The handshake lines may be programmed to perform a variety of tasks such as interrupt requests, setting flags, latching data, and data transfer requests and/or acknowledgements. The handshake functions are programmed via control registers A and B (CRA and CRB). Additional information may be found in **PIN DESCRIPTIONS, REGISTER DESCRIPTION,** or **HANDSHAKE OPERATION.**

MOTEL

The MOTEL circuit is a concept that permits the MC146823 to be directly interfaced with different types of multiplexed bus microprocessors without any additional external logic. For a more detailed description of the multiplexed bus, see **MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7).** Most multiplexed microprocessors use one of two synchronous buses to interface peripherals. One bus was originated by Motorola in the MC6803 and the other by Intel in the 8085.

The MOTEL circuit (for MOTorola and intEL bus) is built into peripheral and memory ICs to permit direct connection to either type of bus. A functional diagram of the MOTEL circuit is shown in Figure 2.

FIGURE 2 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



The microprocessor type is automatically selected by the MOTEL circuit through latching the state of the DS/$\overline{RD}$ pin with AS/ALE. Since DS is always low during AS and $\overline{RD}$ is always high during ALE, the latch automatically indicates with which type microprocessor bus it is interfaced.

PIN DESCRIPTIONS

The following paragraphs contain a brief description of the input and output pins. References (if applicable) are given to other paragraphs that contain more detail about the function being performed.

MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7)

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion of the bus cycle for data. Address-then-data multiplexing does not slow the access time of the MC146823 since the bus reversal from address to data is occurring during the internal register access time.

The address must be valid t_{ASL} prior to the fall of AS/ALE at which time the MC146823 latches the address present on the AD0-AD3 pins. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146823 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to high impedance) t_{DHR} hold time after DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case.

ADDRESS STROBE (AS)

The address strobe input pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the addresses AD0-AD3 to be latched within the MC146823. The automatic MOTEL circuit in the MC146823 also latches the state of the DS pin with the falling edge of AS or ALE.

DATA STROBE OR READ (DS)

The DS input pin has two interpretations via the MOTEL circuit. When generated by a Motorola microprocessor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), or phase 2 (phase 2 clock). During read cycles, DS or $\overline{RD}$ signifies the time that the CPI is to drive the bidirectional bus. In write cycles, the trailing edge of DS or rising edge of $\overline{WR}$ causes the parallel interface to latch the written data present on the bidirectional bus.

The second MOTEL interpretation of DS is that of $\overline{RD}$, MEMR, or I/OR originating from the competitor's microprocessor. In this case, DS identifies the time period when the parallel interface drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the MC146823, latches the state of the DS pin on the falling edge of AS/ALE. When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus microprocessors. To insure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

READ/WRITE (R/$\overline{W}$)

The MOTEL circuit treats the R/$\overline{W}$ input pin in one of two ways. First, when a Motorola microprocessor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ while DS is high.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and I/$\overline{OW}$ from competitor's microprocessors. The MOTEL circuit in this mode gives the R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

CHIP ENABLE ($\overline{CE}$)

The $\overline{CE}$ input signal must be asserted (low) for the bus cycle in which the MC146823 is to be accessed. $\overline{CE}$ is not latched and must be stable prior to and during DS (in the Motorola case of MOTEL) and prior to and during $\overline{RD}$ and $\overline{WR}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the MC146823. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is high, all data, DS, and R/$\overline{W}$ inputs from the microprocessor are disconnected within the MC146823. This permits the MC146823 to be isolated from a powered-down microprocessor.

RESET ($\overline{RESET}$)

The $\overline{RESET}$ input pin is an active-low line that is used to restore all register bits, except the port data register bits, to logical zeros. After reset, all port lines are configured as inputs and no interrupt or handshake lines are enabled.

INTERRUPT REQUEST ($\overline{IRQ}$)

The $\overline{IRQ}$ output line is an open-drain active-low signal that may be used to interrupt the microprocessor with a service request. The "open-drain" output allows this and other interrupt request lines to be wire ORed with a pullup resistor. The $\overline{IRQ}$ line is low when bit 7 of the status register is high. Bit 7 (IRQF) of the handshake/interrupt status register (HSR) is set if any enabled handshake transition occurs; and its associated control register bit is set to allow interrupts. Refer to **INTERRUPT DESCRIPTION** or **HANDSHAKE OPERATION** for additional information.

PORT A, BIDIRECTIONAL I/O LINES (PA0-PA7)

Each line of port A, PA0-PA7, is individually programmable as either an input or output via its data direction register (DDRA). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. See Figure 3 for typical I/O circuitry and Table 1 for I/O operation.

There are three data registers associated with port A: PDA, P1DA, and P2DA. P1DA and P2DA are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to the port A data register, PDA, is latched into the port A output latch regardless of the state of the DDRA. Data written to P1DA or P2DA is ignored and has no affect upon the output data latch or the I/O lines. An MPU read of port bits programmed as outputs reflect the last value written to the PDA register. Port A pins programmed as inputs may be latched via the handshake line PC4/CA1 (see

FIGURE 3 — TYPICAL PORT I/O CIRCUITRY

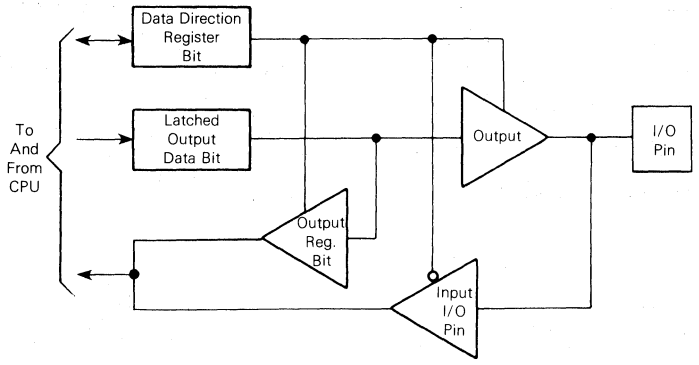


TABLE 1 — PORT DATA REGISTER ACCESSES (ALL PORTS)

R/$\overline{W}$	DDR Bit	Results
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

HANDSHAKE OPERATION) and latched input data may be read via any of the three port A data registers. If the port A input latch feature is not enabled, an MPU read of any port A data register reflects the current status of the port A input pins if the corresponding DDRA bits equal zero. Reset has no effect upon the contents of the port A data register; however, all pins will be placed in the input mode (all DDRA bits forced to equal zero) and all handshake lines will be disabled.

PORT B BIDIRECTIONAL I/O LINES (PB0-PB7)

Each line of port B, PB0-PB7, is individually programmable as either an input or an output via its data direction register (DDRB). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one.

There are three data registers associated with port B: PDB, P1DB, and P2DB. PDB is used for simple port B data reads and writes. P1DB and P2DB are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to PDB or P1DB data register is latched into the port B output latch regardless of the state of the DDRB. An MPU read of port bits programmed as outputs reflect the last value written to a port B data register. An MPU read of any port B register reflects the current status of the input pins whose DDRB bits equal zero. Reset has no effect upon the contents of the port B data register; however, all pins will be placed in the input mode (all DDRB bits forced to equal zero) and all handshake lines will be disabled.

PORT C, BIDIRECTIONAL I/O LINES (PC0-PC3)

Each line of port C, PC0-PC3, is individually programmable as either an input or an output via its data direction register (DDRC). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. Port C data register (PDC) is used for simple port C data reads and writes.

Data written into PDC is latched into the port C data latch regardless of the state of the DDRC. An MPU read of port C bits programmed as outputs reflect the last value written to the PDC register. An MPU read of the port C register reflects the current status of the corresponding input pins whose DDRC bits equal zero. Reset has no effect upon the contents of the port C data register; however, all pins will be placed in the input mode (all DDRC bits forced to equal zero) and all handshake lines will be disabled.

PORT C BIDIRECTIONAL I/O LINE OR PORT A INPUT HANDSHAKE LINE (PC4/CA1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC4/CA1 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC4/CA1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT A BIDIRECTIONAL HANDSHAKE LINE (PC5/CA2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC5/CA2 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC5/CA2 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B INPUT HANDSHAKE LINE (PC6/CB1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O pin, PC6/CB1 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC6/CB1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B BIDIRECTIONAL HANDSHAKE LINE (PC7/CB2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O line, PC7/CB2 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC7/CB2 performs as described in **HANDSHAKE OPERATION.**

HANDSHAKE OPERATION

Up to four port C pins can be configured as handshake lines for ports A and B (one input-only and one bidirectional line for each port) via the port C function select register (FSR). The direction of data flow for the two bidirectional handshake lines (CA2 and CB2) is determined by bits 5 and 7, respectively, of the port C data direction register (DDRC). Actual handshake operation is defined by the appropriate port control register (CRA or CRB).

The control registers allow each handshake line to be programmed to operate in one of four modes. CA2 and CB2 each have four input and four output modes. For detailed information, see Tables 2 and 3.

A summary of the handshake modes is given in the input and output sections that follow. All handshake activity is disabled by reset.

INPUT

Handshake lines programmed as inputs operate in any of four different modes as defined by the control registers (see Table 2). A bit in the handshake/interrupt status register (HSR) is set to a logic one on an active transition of any handshake line programmed as an input. Modes 0 and 1 define a negative transition as active; modes 2 and 3 define a positive transition as active. If modes 1 or 3 are selected on any input handshake line then the active transition of that line results in the IRQF bit of the HSR being set to a logic one and causes the interrupt line ($\overline{IRQ}$) to go low. $\overline{IRQ}$ is released by clearing the HSR bits that are input handshake lines which have interrupts enabled.

If an active transition occurs while the associated HSR bit is set to a logic one, the corresponding bit in the handshake warning register (HWR) is set to a logic one indicating that service of at least one active transition was missed. An HWR bit is cleared to a logic zero by first accessing the appropriate port data register, to clear the appropriate HSR status bit, followed by a read of the HWR.

TABLE 2 — INPUT HANDSHAKE MODES

Mode	Control Register Bits*	Active Edge	Status Bit In HSR	$\overline{IRQ}$ Pin
0	00	– Edge	Set high on active edge.	Disabled
1	01	– Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.
2	10	+ Edge	Set high on active edge.	Disabled
3	11	+ Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.

* Cleared to logic zero on reset.

TABLE 3 — OUTPUT HANDSHAKE LINES (CA2 AND CB2 ONLY)

Mode	Control Register CRA(B) Bits 3 and 4*	Handshake Line Set High	Handshake Line Cleared Low	Default Level
0	00	Handshake set high on active transition of CA1 input. Handshake set high on active transition of CB1 input.	Read of P1DA or a read of P2DA while HSA1 is cleared. Write of port B P1DB or write of P2DB while HSB1 is cleared.	High
1	01	High on the first positive (negative) transition of AS while CA2 (CB2) is low.	Low on the first positive (negative) transition on AS following a read (write) of port A(B) data registers P1DA(B) or P2DA(B).	High
2	10	Never	Always	Low
3	11	Always	Never	High

* Cleared to logic zero on reset.

INPUT LATCH

Port A input-only handshake line (PC4/CA1) can be programmed to function as a latch enable for port A input data via CA1 LE (bit 2 of CRA). If CA1 LE is programmed to a logic one, an active transition of PC4/CA1 will latch the current status of the port A input pins into all three port A data registers (PDA, P1DA, and P2DA). When CA1 LE is enabled, port A and PC4/CA1 function as an 8-bit transparent latch; that is, if the HSA1 bit in the HSR is a logic zero then a read of any port A register reflects the current state of the port A input pins and corresponding bits of the output data latch for port A output pins. If HSA1 is a logic one, a read of any port A data register reflects the state of the port A input pins when HSA1 was set and the corresponding bits of the port A output data latch for port A output pins.

Further transitions of PC4/CA1 result only in setting the HWA1 bit in the HWR and do not relatch data into the port A registers. Latched data is released only by clearing HSA1 in the HSR to a logic zero (HSA1 is cleared by reading P1DA).

OUTPUT

Each bidirectional handshake line programmed as an output by the DDRC operates in one of four modes as described in Table 3. Modes 2 and 3 force the output handshake line to reflect the state of bit 4 in the appropriate control register.

In modes 0 and 1, PC5/CA2 is forced low during the cycle following a read of P1DA or a read of P2DA while HSA1 is cleared. PC7/CB2 is forced low during the cycle following a write to P1DB or a write to P2DB while HSB1 is cleared. Because of these differences, port A is the preferred input port and port B is the preferred output port.

In mode 0, PC5/CA2 (PC7/CB2) is set high by an active transition of PC4/CA1 (PC6/CB1). In mode 1, PC5/CA2 (PC7/CB2) is set high in the cycle following the cycle in which PC5/CA2 (PC7/CB2) goes low. Mode 1 forces a low-going pulse on PC5/CA2 (PC7/CB2) following a read (write) of P1DA (P1DB) or P2DA (P2DB) that is approximately one cycle time wide.

When entering an output handshake mode for the first time after a reset, the handshake line outputs the default level as listed in Table 3.

INTERRUPT DESCRIPTION

The MC146823 allows an MPU interrupt request ($\overline{IRQ}$ low) via the input handshake lines. The input handshake line, operating in modes 1 or 3 as defined by the control registers (CRA and CRB), causes $\overline{IRQ}$ to go low when IRQF (interrupt flag) in the HSR is set to a logic one. $\overline{IRQ}$ is released when IRQF is cleared. See **Handshake/Interrupt Status Register** under **REGISTER DESCRIPTION** for additional information.

REGISTER DESCRIPTION

The MC146823 has 15 registers (see Figure 1) which define the mode of operation and status of the port pins. The following paragraphs describe these registers.

Register Names:

Control Register A (CRA)
Control Register B (CRB)

Register Addresses:

$9 (CRA)
$A (CRB)

Register Bits:

	7	6	5	4 3	2	1 0
$9	X	X	X	CA2 Mode	CA1 LE	CA1 Mode
$A	X	X	X	CB2 Mode	X	CB1 Mode

Purpose:

These two registers control the handshake and interrupt activity for those pins defined as handshake lines by the port C function select register (FSR).

Description:

CA2 and CB2 are programmed as inputs or outputs via the associated DDRC bits. Each handshake line is controlled by two mode bits. Bit 2 of CRA enables the Port A latch for an active CA1 transition. Table 2 describes the input handshake modes (CA1, CB1, CA2, CB2) and Table 3 describes the output handshake modes for CA2 and CB2.

Register Names:

Port A Data Registers (PDA, P1DA, P2DA)

Register Addresses:

$2 (PDA), $0 (P1DA), $1 (P2DA)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

These three registers serve different purposes. PDA is used to read input data and latch data written to the port A output pins. P1DA and P2DA are used to read input data and to affect handshake and status activity for PC4/CA1 and PC5/CA2. If enabled, port A input data may be latched into the three port A data registers on an active PC4/CA1 transition as described in **HANDSHAKE OPERATION.**

Description:

Data written into PDA is latched into the port A output latch (see Figure 3) regardless of the state of DDRA. Output pins, as defined by DDRA, assume the logic levels of the corresponding bits in the PDA output latch. The PDA output latch allows the user to read the state of the port A output data. If the input latch is not enabled, a read of any port A data register reflects the current state of the port A input pins as defined by DDRA and the contents of the output latch for output pins. Writes into P1DA or P2DA have no effect upon the output pins or the output data latch. Users are recommended to initialize the port A output latch before changing any pin to an output via the DDRA.

MPU accesses of P1DA or P2DA are primarily used to affect handshake and status activity. A summary of the effects on the status and warning bits of port A data register accesses is given in Table 4. For more information, see **HANDSHAKE OPERATION** and **Control Register A (CRA)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port A data register.

Register Names:

Port B Data Registers (PDB, P1DB, P2DB)

Register Addresses:

$3 (PDB), $C(P1DB), $D (P2DB)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

These three registers serve different purposes. The Port B data registers are used to read input data and to latch data written to the port B output pins. Writes to PDB and P1DB affect the contents of the output data latch while writes to P2DB do not affect the output data latch. P1DB and P2DB accesses additionally affect handshake and status activity for PC6/CB1 and PC7/CB2.

Description:

Data written into PDB and P1DB port B registers is latched into the port B output latch (see Figure 3) regardless of the state of DDRB. Output pins, as defined by DDRB, assume the logic levels of the corresponding bits in the port B output latch. Reads of any port B data registers reflect the contents of the output data latch for output pins and the current state of the input pins (as determined by DDRB). Users are recommended to initialize the port B output latch before changing any pin to an output via the DDRB.

MPU accesses of P1DB or P2DB are primarily used to affect handshake and status activity. A summary of the effects on status and warning register bits of port B data register accesses is given in Table 5. For more information, see **HANDSHAKE OPERATION** or **Control Register B (CRB)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port B data register.

TABLE 4 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT A DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDA	None	None	None	Yes	Yes
P1DA	HSA1 cleared to a logic zero.	HWA1 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No
P2DA	HSA2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No

TABLE 5 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT B DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDB	None	None	None	Yes	Yes
P1DB	HSB1 cleared to a logic zero.	HWB1 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in the CRB.	Yes	Yes
P2DB	HSB2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in CRB.	Yes	No

Register Name:

Port C Data Register (PDC)

Register Address:

$4

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

The port C data register (PDC) is used to read input data and to latch data written to the output pins.

Description:

Data is written into the port C output latch (see Figure 3) regardless of the state of DDRC. Any port C pin defined as a handshake line by the port C function select register (FSR) is not affected by PDC. Output pins, as defined by DDRC, assume logic levels of the corresponding bits in the port C output latch. A read of PDC reflects the contents of the output latch for output pins and the current state of the input pins (as reflected in the DDRC). Reset has no effect upon the contents of PDC. Users are recommended to initialize the port C output data latch before changing any pin to an output via the DDRC.

Register Name:

Data Direction Register for Port A (B) (C)

Register Address:

$6 ($7) ($8)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:

Each of the three data direction registers (DDRA, DDRB, and DDRC) define the direction of data flow of the port pins for ports A, B, and C.

Description:

A logic zero in a DDR bit places the corresponding port pin in the input mode. A logic one in a DDR bit places the corresponding pin in the output mode. Any port C pins defined as bidirectional handshake lines also use the port C DDR (DDRC). Input-only handshake lines are not affected by DDRC. Reset clears all DDR bits to logic zero configuring all port pins as inputs. The DDRs have no write-inhibit control over the port data output latches. Data may be written to the port data registers even though the pins are configured as inputs.

Register Name:

Port C Pin Function Select Register (FSR)

Register Address:

$B

Register Bits:

7	6	5	4	3	2	1	0
CFB2	CFB1	CFA2	CFA1	XX	XX	XX	XX

Purpose:

The port C pin function select register defines whether the multifunction port C pins are to operate as "normal" port C lines or as handshake lines.

Description:

A logic zero in any FSR bit defines the corresponding port C pin as a "normal" I/O pin. A logic one in any valid FSR bit defines the corresponding port C pin as a handshake line. Pins defined as handshake lines function according to the contents of control register A (CRA) or control register B (CRB). The port C data direction register (DDRC) is valid regardless of FSR contents for all pins except PC4/CA1 and PC6/CB1. Transitions on port C pins not defined as handshake pins do not effect the handshake/interrupt status register. Reset clears all FSR bits to a logic zero. Users are recommended to initialize the data direction and control registers before modifying the FSR.

Register Name:

Handshake/Interrupt Status Register (HSR)

Register Address:

$E

Register Bits:

7	6	5	4	3	2	1	0
IRQF	XX	XX	XX	HSB2	HSA2	HSB1	HSA1

Purpose:

The handshake interrupt status register is a read-only flag register that may be used during a polling routine to determine if any enabled input handshake transition, as defined by the control register (CRA and CRB), has occurred.

Description:

If an enabled input handshake transition occurs then the appropriate HSR bit (HSB2, HSA2, HSB1, or HSA1) is set. The IRQ flag bit (bit 7, IRQF) is set when one or more of the HSR bits 0-3 and their corresponding control register bits are set to a logic one as shown in the following equation:

$$\text{Bit } 7 = \text{IRQF} = [\text{HSB2} \bullet \text{CRB2(3)}] + [\text{HSA2} \bullet \text{CRA2(3)}] + [\text{HSB1} \bullet \text{CRB1(0)}] + [\text{HSA1} \bullet \text{CRA1(0)}]$$

The numbers in ( ) indicate which bit in the control register enables the interrupt.

Handshake/interrupt status register bits are cleared by accessing the appropriate port data register. The following table lists the HSR bit and the port data register that must be accessed to clear the bit.

To Clear HSR Bit	Access Register
HSB2	P2DB
HSA2	P2DA
HSB1	P1DB
HSA1	P1DA

Reset clears all handshake/interrupt status register bits to a logic zero.

Register Name:

Handshake Warning Register (HWR)

Register Address:

$F

Register Bits:

7	6	5	4	3	2	1	0
XX	XX	XX	XX	HWB2	HWA2	HWB1	HWA1

Purpose:

The warning register is a read-only flag register that may be used to determine if a second attempt to set a handshake/interrupt status register bit has been made before the original had been serviced.

Description:

Each bit in the handshake/interrupt status register, except IRQF, has a corresponding bit in the handshake warning register. If an attempt is made to set a bit in the handshake/interrupt status register that is already set, then the corresponding bit in the handshake warning register is also set. An attempt is the occurrence of any enabled input handshake transition as defined by the control registers.

A handshake warning register bit is cleared by first reading the appropriate data register then reading the handshake warning register. Reading the data register (either P1DA, P2DA, P1DB, or P2DB) loads a buffer latch with the proper bit in the handshake warning register (HWA1, HWA2, HWB1, and HWB2, respectively). The next read of the handshake warning register clears the appropriate bit without affecting the other three handshake warning register bits. The upper four bits, HWR4-HWR7, always read as logic zeros. If a port data register is not read before reading the handshake warning register, then the handshake warning register bits will remain unaffected. Reset clears all HWR bits to a logic zero.

Recommended status register handling sequence:

1. Read status register (User determines which if any enabled handshake transition occurred)
2. Read/write port data indicated by status register (Clears associated status bit and latches appropriate warning register bit in the buffer latch)
3. Read warning register (Latched warning bit is cleared and the remaining bits are unaffected)

TYPICAL INTERFACING

The MC146823 is best suited for use with microprocessors which generate an address-then-data-multiplexed bus. Figure 4 shows the MC146823 in a typical CMOS system that uses the MC146805E2 CMOS MPU. Other multiplexed microprocessors can be used as easily.

A single-chip microcomputer (MCU) may be interfaced with 11 port lines as shown in Figure 5. This interface also requires some software overhead to gain up to 13 additional I/O lines and the MC146823 handshake lines.

FIGURE 4 — A TYPICAL CMOS MICROPROCESSOR SYSTEM

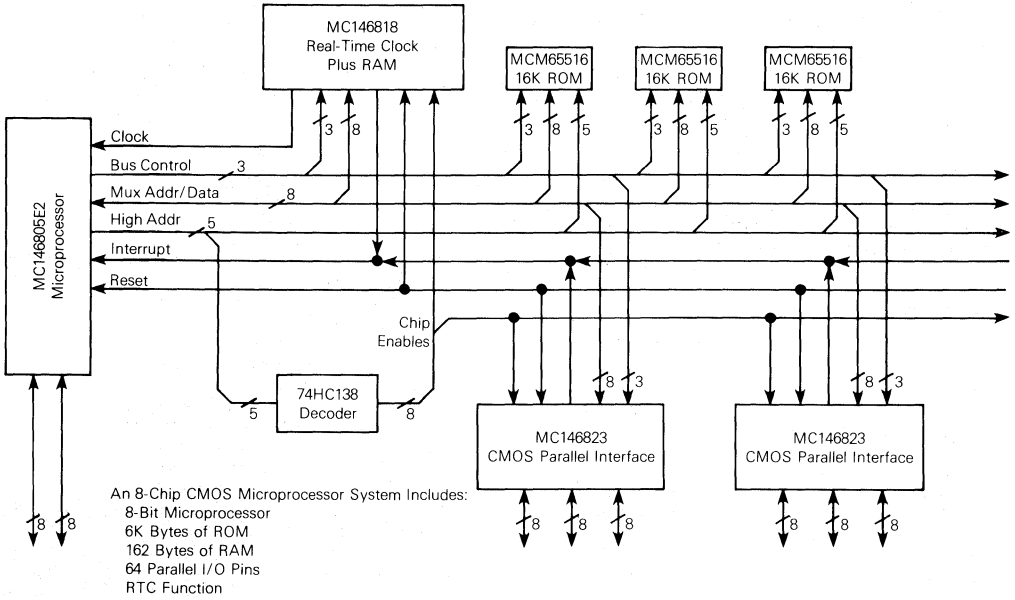
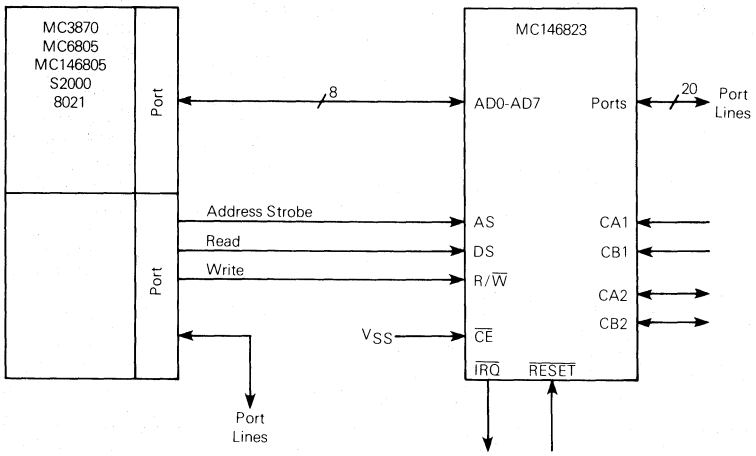


FIGURE 5 — MC146823 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE-CHIP MICROCOMPUTER



MOTOROLA

MC1468705F2

Product Preview

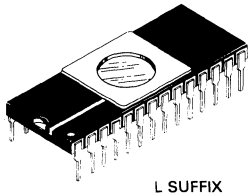
CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT EPROM MICROCOMPUTER

8-BIT EPROM MICROCOMPUTER UNIT

The MC1468705F2 Microcomputer Unit (MCU) is an EPROM member of the M6805 Family of low-cost single-chip microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turnaround time for prototype evaluation of mask ROM versions. This 8-bit microcomputer contains a CPU, on-chip oscillator, EPROM, bootstrap ROM, RAM, I/O, and a TIMER.

The MC1468705F2 is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low-power consumption constitutes an important factor.



L SUFFIX
CERAMIC PACKAGE
CASE 719

3

HARDWARE FEATURES

- Low Power Wait Mode
- Typical Stop Mode Power of 25 μW
- 8-Bit Architecture
- Fully Static Operation
- Single 3- to 5.5-Volt Supply
- 1080 Bytes of On-Chip User EPROM
- 64 Bytes of On-Chip RAM
- Memory Mapped I/O
- 16 Bidirectional I/O Lines
- Four Input-Only Lines
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Bootstrap Program in ROM Simplifies EPROM Programming
- Master Reset and Power-On Reset
- On-Chip Oscillator
- 1 μs Cycle Time
- 28-Pin Dual-In-Line Package

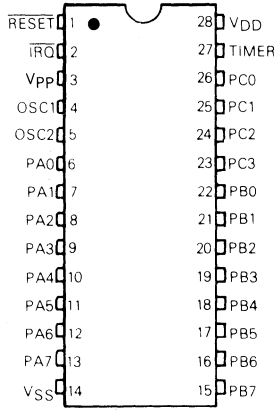
SOFTWARE FEATURES

- Similar to M6800 Family
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- 10 Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory mapped I/O
- Two Power Saving Standby Modes

USER PROGRAMMABLE OPTIONS

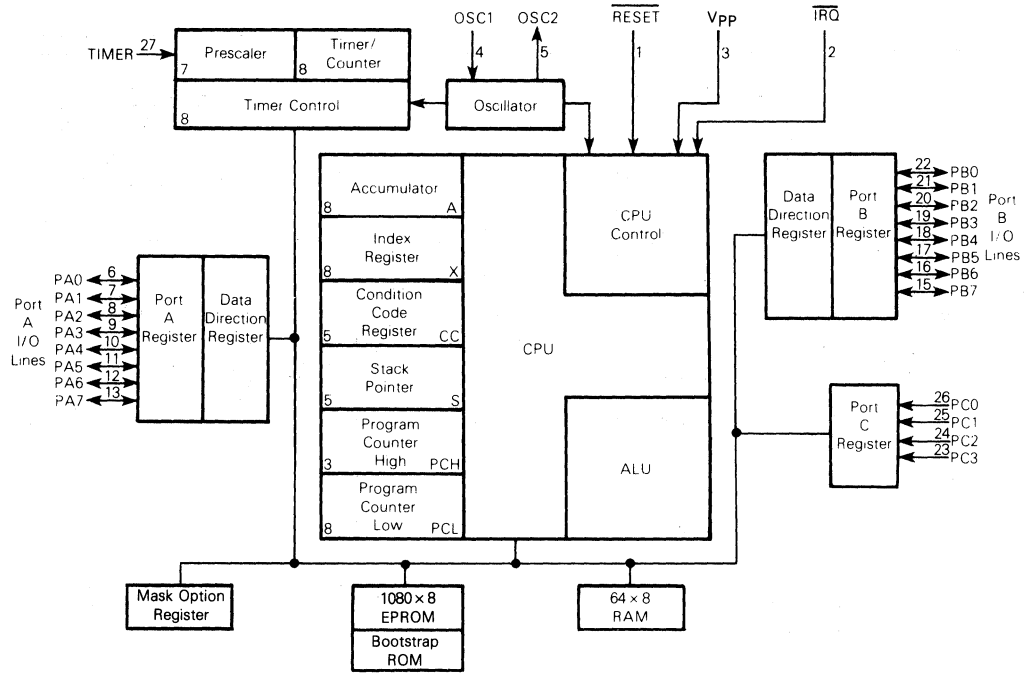
- Crystal or Low-Cost Resistor Oscillator Option
- Oscillator Internally Divided by Two or Four
- Interrupts Edge Sensitive Only or Level and Edge Sensitive

PIN ASSIGNMENT

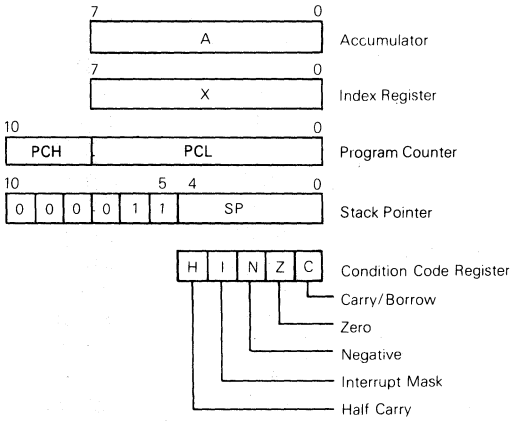


This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

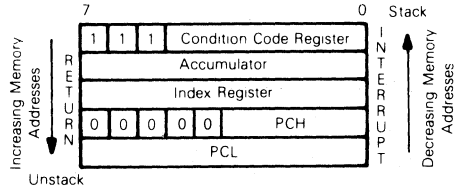
BLOCK DIAGRAM



PROGRAMMING MODEL



STACKING ORDER



NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

MOTOROLA

MC1468705G2

Advance Information

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT EPROM
MICROCOMPUTER

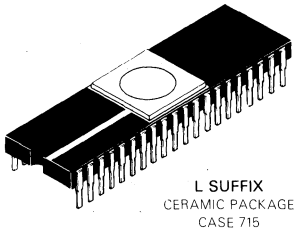
3

8-BIT EPROM MICROCOMPUTER UNIT

The MC1468705G2 Microcomputer Unit (MCU) is an EPROM member of the MC146805 CMOS Family of Microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turnaround time for prototype evaluation of the mask ROM versions. This 8-bit microcomputer contains on-chip oscillator, CPU, RAM, EPROM, self-programming bootstrap ROM, I/O, and TIMER.

In addition to power saving STOP and WAIT modes, fully static design allows operation at frequencies down to dc, further reducing its already low power consumption. It is a low-power processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor.

The following are the major features of the MC1468705G2 EPROM MCU.



L SUFFIX
CERAMIC PACKAGE
CASE 715

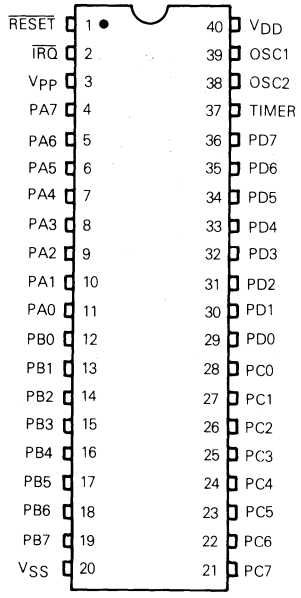
HARDWARE FEATURES

- Typical Full Speed Operating Power of 20 mW at 5 V
- Typical WAIT Mode Power of 5 mW
- Typical STOP Mode Power of 5 μW
- Fully Static Operation
- 112 Bytes of On-Chip RAM
- 2106 Bytes of UV Erasable, User Programmable ROM (EPROM)
- 32 Bidirectional I/O Lines
- High Current Drive
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler
- External Timer Input
- External and Timer Interrupts
- Master Reset and Power-On Reset
- Single 3 to 5.5 V Supply
- On-Chip Oscillator with RC or Crystal Options Selected by EPROM Mask Option Register
- Plug-In Compatible with the MC146805G2
- Self-Programming Bootstrap Program in ROM Simplifies EPROM Programming

SOFTWARE FEATURES

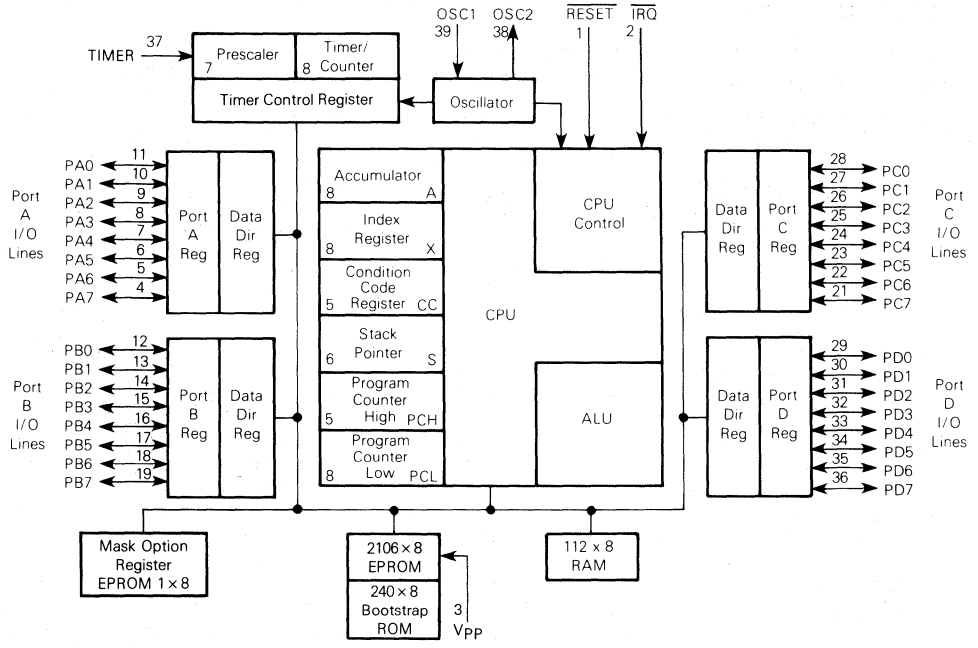
- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power Saving Standby Modes; WAIT and STOP
- Fully Compatible with M146805 CMOS Family Microcomputers

PIN ASSIGNMENT



This document contains information on a new product. Specifications and information herein are subject to change without notice.

FIGURE 1 — BLOCK DIAGRAM



MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +5.5	V
Input Voltage	V_{in}	V_{SS}−0.3 to V_{DD}+0.3	V
EPROM Programming Voltage (V_{PP} Pin)	V_{in}	−0.3 to −13.5	V
Current Drain Per Pin Excluding V_{DD}, V_{SS}, and V_{PP} V_{PP}	I	10 −30	mA mA
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain Total (PD4-PD7 Only)	I_{OH}	40	mA

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Ceramic	θ_{JA}	50	°C/W

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 $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs except OSC2 and V_{PP} are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Be sure that the EPROM window is shielded from light with an opaque cover at all times except when erasing.

BOOTSTRAP PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS (V_{DD} = 5.25 Vdc, V_{SS} = 0 Vdc, T_A = 25°C unless otherwise noted)

Characteristics	Symbol	Min	Typ	Max	Unit
Programming Voltage (V_{PP} Pin) (Figure 20)	V_{PP}	–	– 13.5	–	V
V_{PP} Supply Current V_{PP} = – 13.5 V	I_{PP}	–	30	–	mA
Programming Oscillator Frequency	f_{oscp}	–	–	1.0	MHz
Bootstrap Programming Mode Voltage ($\overline{IRQ}$ Pin) I_{in} = 100 μA Max	V_{IRQP}	–	– 12.0	–	V
TIMER Pin Programming Mode Voltage	V_{TIMP}	V_{DD}	–	V_{DD}	V

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 0 Vdc, T_A = 0°C to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output Voltage, $I_{Load} \leq$ 10.0 μA	V_{OL} V_{OH}	– V_{DD} – 0.1	0.1 –	V
Output High Voltage (I_{Load} = – 100 μA) PB0-PB7, PC0-PC7 (I_{Load} = – 2 mA) PA0-PA7, PD0-PD3 (I_{Load} = – 8 mA) PD4-PD7	V_{OH}	2.4	–	V
Output Low Voltage (I_{Load} = 800 μA) PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	V_{OL}	–	0.4	V
Input High Voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7 $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	V_{IH}	V_{DD} – 2.0 V_{DD} – 0.8	V_{DD} V_{DD}	V
Input Low Voltage All Inputs (except V_{PP})	V_{IL}	V_{SS}	0.8	V
Input Low Voltage V_{PP} (normal oper. mode)	V_{IL}	V_{SS}	V_{SS}	V
Total Supply Current (C_L = 50 pF on Ports, no dc Loads, t_{cyc} = 1 μs) RUN (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V) WAIT (See Note 1) STOP (See Note 1)	I_{DD}	– – –	10 5 250	mA mA μA
I/O Ports Input Leakage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	± 10	μA
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	± 1	μA
Capacitance Ports $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	C_{out} C_{in}	– –	12 8	pF pF

NOTES:
1. Test conditions for I_{DD} are as follows:
   All ports programmed as inputs
   V_{IL} = 0.2 V (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)
   V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
   OSC1 input is a squarewave from 0.2 V to V_{DD} – 0.2 V
   OSC2 output load = 20 pF (WAIT I_{DD} is affected linearly by the OSC2 capacitance)
2. V_{PP} is pin 3 on the MC1468705G2 and is connected to V_{SS} in the normal operating mode.

TABLE 1 — CONTROL TIMING
(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc; T_A = 0 to 70°C)

Characteristic	Symbol	Min	Max	Unit
Crystal Oscillator Startup Time (See Figure 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (See Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (See Figure 4)	t_{TH}, t_{TL}	500	–	t_{cyc}
$\overline{RESET}$ Pulse Width (See Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (See Figure 4)	t_{TLTL}	1000	–	ns
Interrupt Pulse Width Low (See Figure 14)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (See Figure 14)	t_{ILIL}	*	–	t_{cyc}
OSC1 Pulse Width (with External Clock)	t_{OH}, t_{OL}	125	–	ns
Cycle Time	t_{cyc}	1000	–	ns
Frequency of Operation Crystal (÷ 4 option) External Clock (÷ 4 option) Crystal (÷ 2 option) External Clock (÷ 2 option)	f_{osc} f_{osc} 	– dc – dc	4.0 4.0 2.0 2.0	MHz MHz MHz MHz

* The minimum period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routines plus 20 t_{cyc} cycles.

FIGURE 2 — EQUIVALENT TEST LOAD

Port	R_1	R_2
B and C	24.3 kΩ	4.32 kΩ
A, PD0-PD3	1.21 kΩ	3.1 kΩ
PD4-PD7	300 Ω	1.64 kΩ

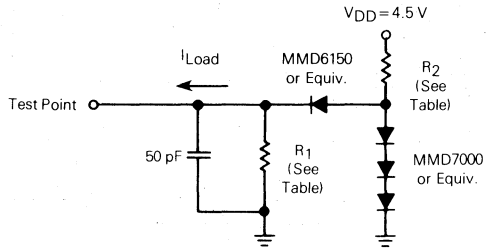


FIGURE 3 — TYPICAL OPERATING CURRENT vs INTERNAL FREQUENCY

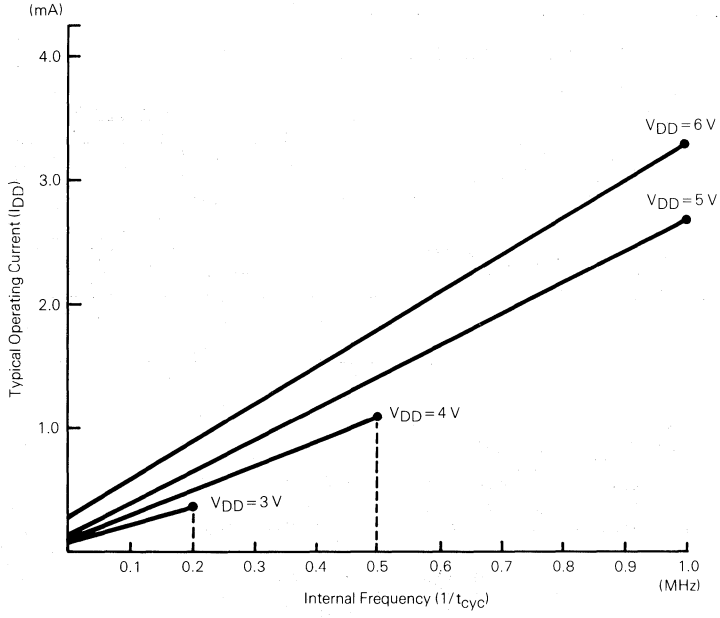


FIGURE 4 — TIMER RELATIONSHIPS

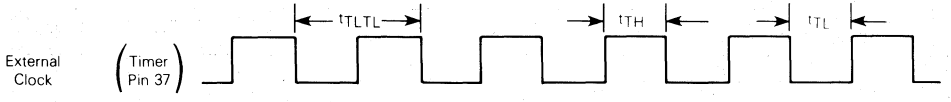
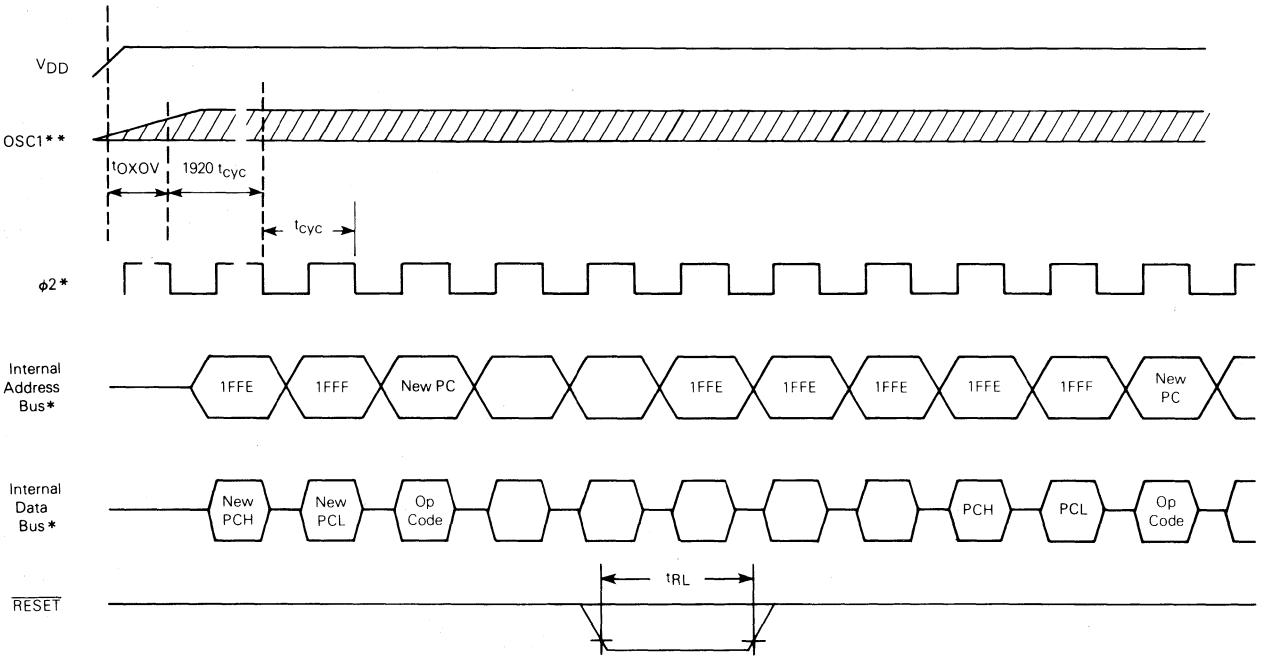
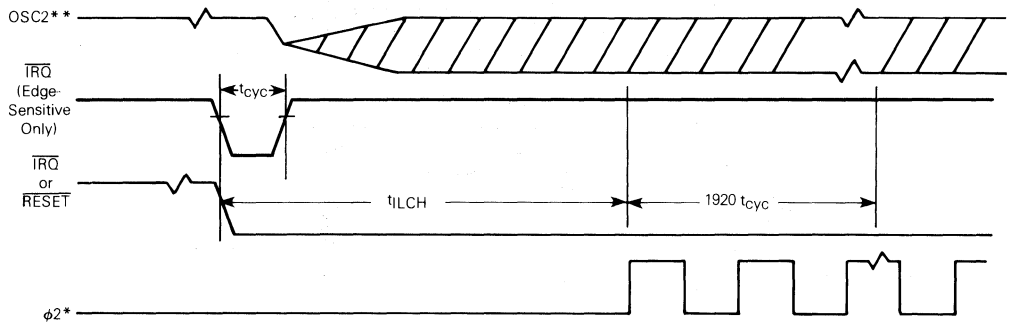


FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$



*Internal timing signal and bus information not available externally.
**OSC1 line is not meant to represent frequency. It is only used to represent time.

FIGURE 6 — STOP RECOVERY AND POWER-ON RESET



*Internal timing signals not available externally.
**Represents the internal gating of the OSC1 input pin.

FUNCTIONAL PIN DESCRIPTION

V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power and V_{SS} is ground.

IRQ (MASKABLE INTERRUPT REQUEST)

IRQ is a programmable option which provides two different choices of interrupt triggering sensitivity. These options are: (1) negative edge-sensitive triggering only, or (2) both negative-edge sensitive and level-sensitive triggering. In the latter case, either type of input to the IRQ pin will produce the interrupt. The MCU completes the current instruction before it responds to the interrupt request. When the IRQ pin goes low for at least one t_{cyc}, a logic one is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logic one, and the interrupt mask bit (I bit) in the condition code register is clear, the MCU then begins the interrupt sequence.

If the option is selected to include level-sensitive triggering, then the IRQ input requires an external resistor to V_{DD} for "wire-OR" operation. See **INTERRUPTS** for more detail. This pin also detects a negative voltage that is used to initiate the bootstrap mode program.

RESET

The RESET input is not required for startup but can be used to reset the MCU internal state and provide an orderly software startup procedure. Refer to **RESETS** for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. This pin is connected to V_{DD} for the bootstrap mode (EPROM programming). Refer to **TIMER** for additional information about the timer circuitry.

V_{PP}

The V_{PP} pin is used when programming the EPROM. By applying the negative programming voltage to this pin, one of the requirements is met for programming the EPROM. Refer to **PROGRAMMING FIRMWARE** and the **PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS** table.

NOTE

In normal operation, this pin is connected *directly* to V_{SS}.

OSC1, OSC2

The MC1468705G2 can be configured to accept either a crystal input or an RC network to control the internal oscillator. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the internal oscillator frequency (f_{osc}). Both of these options are programmable via the mask option register (MOR) in the EPROM array. The programmable options provided via the MOR in the MC1468705G2 are mask options in the MC146805G2.

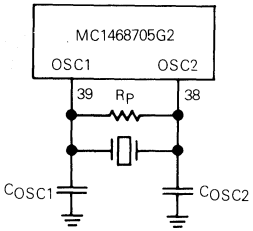
RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 7(d). The relationship between R and f_{osc} is shown in Figure 8.

CRYSTAL — The circuit shown in Figure 7(b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc} in Table 1 Control Timing. Using an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to Table 1 for V_{DD} specifications.

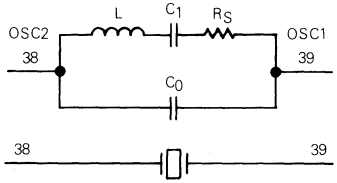
FIGURE 7— OSCILLATOR CONNECTIONS

	1 MHz	4 MHz	Units
R_S (Max)	400	75	Ω
C_0 (Max)	5	7	pF
C_1	0.008	0.012	μF
C_{OSC1}	15-40	15-30	pF
C_{OSC2}	15-30	15-25	pF
R_P (Min)	10	10	MΩ
Q	30	40	K

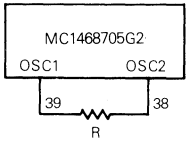
(a) Crystal Parameters



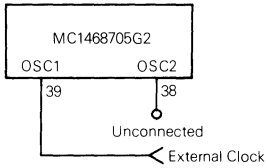
(b) Crystal Oscillator Connections



(c) Equivalent Crystal Circuit

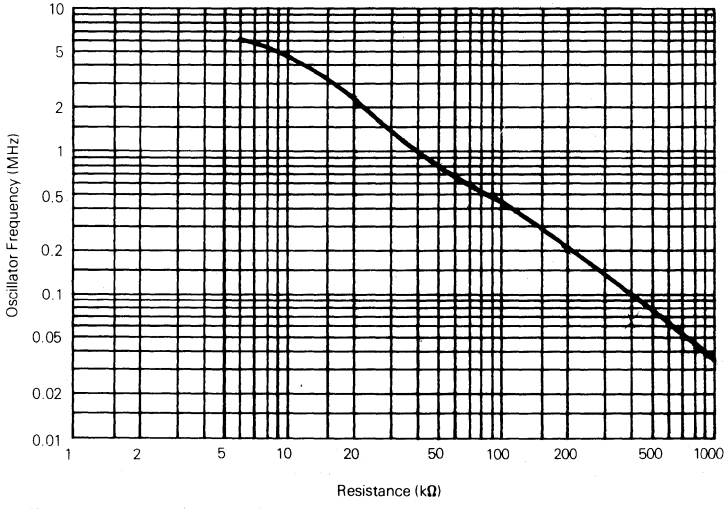


(d) RC Oscillator Connection



(e) External Clock Source Connections

FIGURE 8 — TYPICAL FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 7(e). An external clock should be used with the crystal oscillator option and its pulse width should be the t_{OH}, t_{OL} specification. The t_{OXOV} or t_{ILCH} specifications do not apply when using an external clock input.

PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to **PROGRAMMING** for a detailed description of I/O programming.

PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to **PROGRAMMING** for a detailed description of I/O programming.

PC0-PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to **PROGRAMMING** for a detailed description of I/O programming.

PD0-PD7

These eight lines comprise Port D. PD4-PD7 also are capable of driving LEDs directly. The state of any pin is software programmable. Refer to **PROGRAMMING** for a detailed description of I/O programming.

PROGRAMMING

INPUT/OUTPUT PROGRAMMING

Any port pin may be software programmed as an input or output by the state of the corresponding bit in the port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At reset, all DDRs are cleared, which configures all port pins as inputs. A port pin configured as an output will output the data in the corresponding bit of its port data latch. Refer to Figure 9 and Table 2.

FIGURE 9 — TYPICAL PORT I/O CIRCUITRY

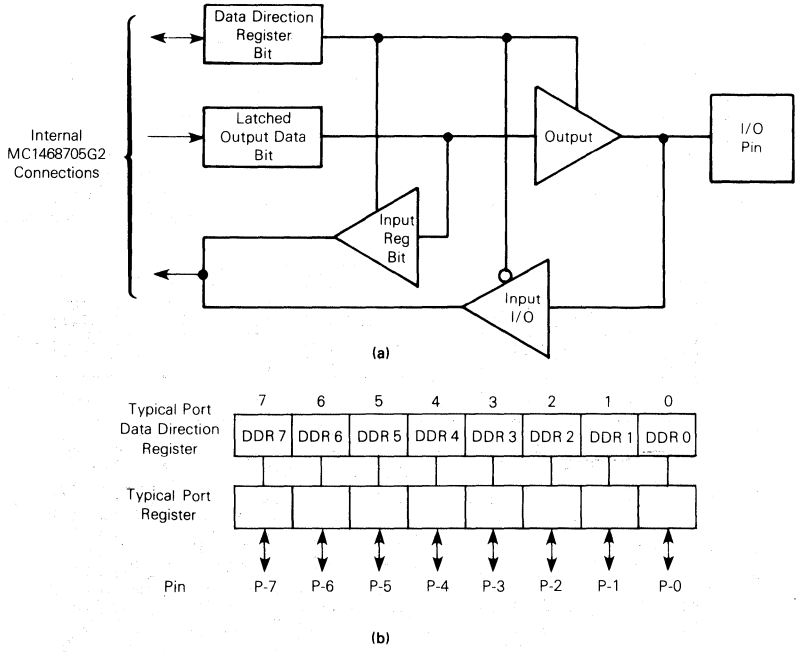


TABLE 2 — I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

*R/$\overline{W}$ is an internal signal.

EPROM PROGRAMMING

When programming the EPROM array within the MC1468705G2, ports are used in a special arrangement. See **PROGRAMMING FIRMWARE** and Figure 20 for a detailed description.

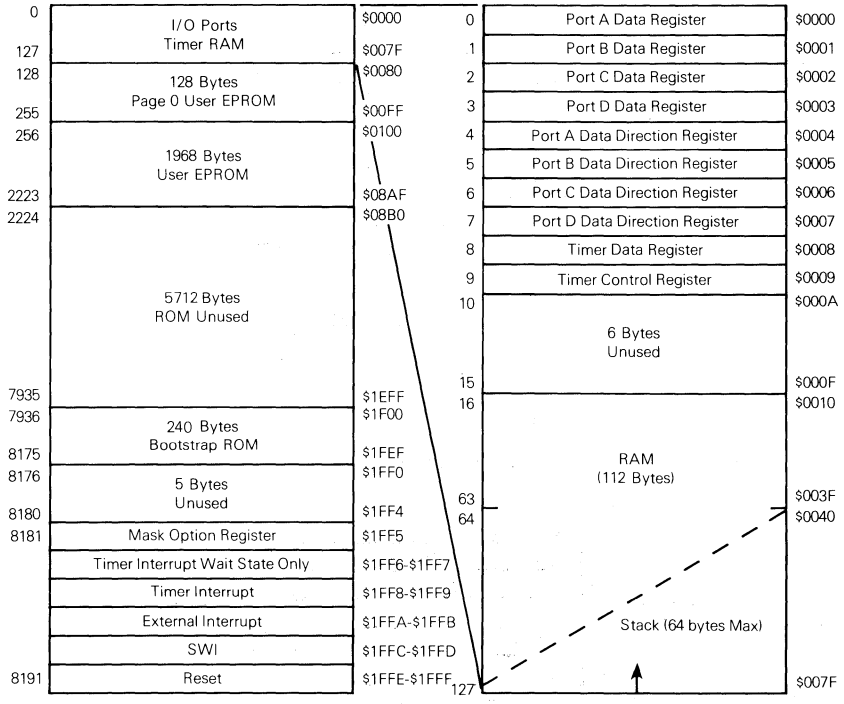
MEMORY

As shown in Figure 10, the MCU is capable of addressing 8192 bytes of memory and I/O registers with its program counter. The MC1468705G2 MCU has implemented 2469 bytes of these locations. This consists of: 2106 bytes of user EPROM, 240 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM mask option register (MOR), eight bytes of I/O, and two timer registers. The user EPROM is located in two areas. The main EPROM area is in memory locations $0080 to $08AF. The second area is reserved for ten interrupt/reset vector bytes at memory locations $1FF6 through $1FFF. The MCU uses 10 of the lowest 16 memory locations for program control and I/O features such as data ports, the port DDRs, and the timer. The mask option register at memory location $1FF5 completes the total. The 112 bytes of user RAM include up to 64 bytes for the stack. Except for the MOR, the memory mapping is similar to the MC146805G2; however, the MC1468705G2 has no self-check ROM because of the bootstrap ROM requirement.

The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The contents of the CPU registers are pushed onto the stack in the order shown in Figure 12. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the higher order five bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call causes only the program counter (PCL, PCH) contents to be pushed onto the stack; the remaining CPU registers are not pushed.

3

FIGURE 10 — ADDRESS MAP



* Reads of unused locations undefined.

REGISTERS

The MC1468705G2 contains five registers, as shown in the programming model of Figure 11. The interrupt stacking order is shown in Figure 12.

ACCUMULATOR (A)

The accumulator is an 8-bit general purpose register used to hold operands, results of the arithmetic calculations, and data manipulations.

INDEX REGISTER (X)

The X register is an 8-bit register which is used during the indexed modes of addressing. It provides an 8-bit value which is used to create an effective address. The index register is also used for data manipulations with the read-modify-write type of instructions and as a temporary storage register when not performing addressing operations.

PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed by the processor.

STACK POINTER (SP)

The stack pointer is a 13-bit register containing the address of the next free location on the stack. When accessing memory, the seven most significant bits are permanently configured to 0000001. These seven bits are appended to the six least significant register bits to produce an address within the range of $007F to $0040. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external or power-on reset, and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit ($007F). Nested interrupt and/or subroutines may use up to 64 (decimal) locations, beyond which the stack pointer wraps around and points to

FIGURE 11 — PROGRAMMING MODEL

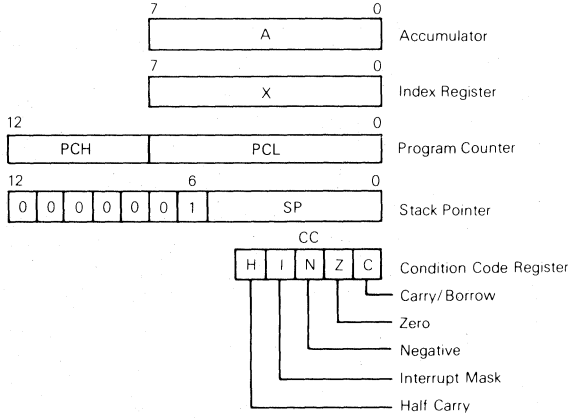
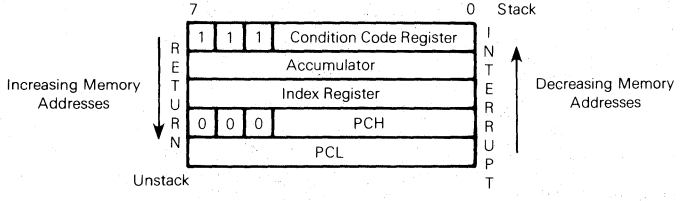


FIGURE 12 — STACKING ORDER



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

its upper limit; thereby, losing the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register which indicates the results of the instruction just executed. These bits can be individually tested by a program and specified action taken as a result of their state. Each bit is explained in the following paragraphs.

HALF CARRY BITS (H) — The H bit is set to a one when a carry occurs between bits 3 and 4 of the ALU and during an ADD or ADC instruction. The H bit is useful in binary coded decimal subroutines.

INTERRUPT MASK BIT (I) — When the I bit is set, both the external interrupt and the timer interrupt are disabled. Clearing this bit enables the above interrupts. If an interrupt occurs while the I bit is set, the interrupt is latched and is processed after the I bit is next cleared.

NEGATIVE (N) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

ZERO (Z) — When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is zero.

CARRY/BORROW (C) — Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

RESETS

The MC1468705G2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function; refer to Figure 5.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the $\overline{\text{RESET}}$ pin must stay low for a minimum of one t_{CYC}. The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger as part of its input to improve its noise immunity.

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 1920 t_{CYC} delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 1920 t_{CYC} time out, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high.

Either of the two types of reset conditions causes the following to occur:

- Timer control register interrupt request bit TCR7 is cleared to a logic zero to preclude premature timer interrupts.
- Timer control register interrupt mask bit TCR6 is set to a logic one to preclude timer interrupt processing.
- All data direction register bits are cleared to logic zeros to define all ports as input.
- Stack pointer is preset to its upper limit, $007F.
- The internal address bus is forced to the reset vector ($1FFE, $1FFF).
- Condition code register interrupt mask bit (I) is set to a logic one to mask any external interrupts.
- STOP and WAIT latches are cleared to place MCU in normal operation.
- External interrupt latch is cleared to ensure no external interrupt is processed.
- MCU operation is set up per mask option register (MOR). External reset does not affect the MOR.

All other functions, such as other registers (including I/O ports), the timer, etc. are not cleared by the reset conditions.

BOOTSTRAP ROM

The bootstrap ROM contains a factory program which allows the MCU to present an address and fetch data from an external device and transfer it into the MC1468705G2 EPROM. The bootstrap program provides: timing of programming pulses, timing of V_{PP} input, and verification after programming. See **PROGRAMMING FIRMWARE.**

MASK OPTION REGISTER (MOR)

The mask option register is an 8-bit user programmed (EPROM) register in which three of the bits are used. Bits in this register are used to select the type of system clock (crystal/RC oscillator), the divide-by-four/divide-by-two clock option (bus frequency), and the edge-sensitive or edge- and level-sensitive triggered interrupt recognition. The MOR is not available on the MC146805G2 ROM-based part.

INTERRUPTS

Systems often require that normal processing be interrupted so that some external event may be serviced. The MC1468705G2 may be interrupted by one of three different methods: either one of two maskable hardware interrupts (external input or timer) or a nonmaskable software interrupt (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 12.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction execution is completed.

When the current instruction is complete, the processor checks all pending hardware interrupts and, if an interrupt is pending and is unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt servicing.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any

other instruction. Refer to Figure 13 for the interrupt and instruction processing sequence.

Table 3 shows the execution priority of the $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, and timer interrupts, and the software interrupt, SWI. Two conditions are shown, one with the I bit set and the other with the I bit clear; however, in either case $\overline{\text{RESET}}$ has the highest priority of execution. If the I bit is set as per Table 3(a), the second highest priority is assigned to SWI. This is illustrated in Figure 13 which shows that the $\overline{\text{IRQ}}$ or Timer interrupts are not executed when the I bit is set and the next instruction (including SWI) is fetched. If the I bit is clear as per Table 3(b), the priorities change in that the next instruction (including SWI) is not fetched until after the $\overline{\text{IRQ}}$ and Timer interrupts have been recognized (and serviced). Also, when the I bit is clear, if both $\overline{\text{IRQ}}$ and Timer interrupts are pending, the $\overline{\text{IRQ}}$ interrupt is always serviced before the Timer interrupt.

NOTE

The conditions for Table 3 assume that, except for $\overline{\text{RESET}}$, the current instruction is completed, thus the MCU is at an instruction boundary.

FIGURE 13 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

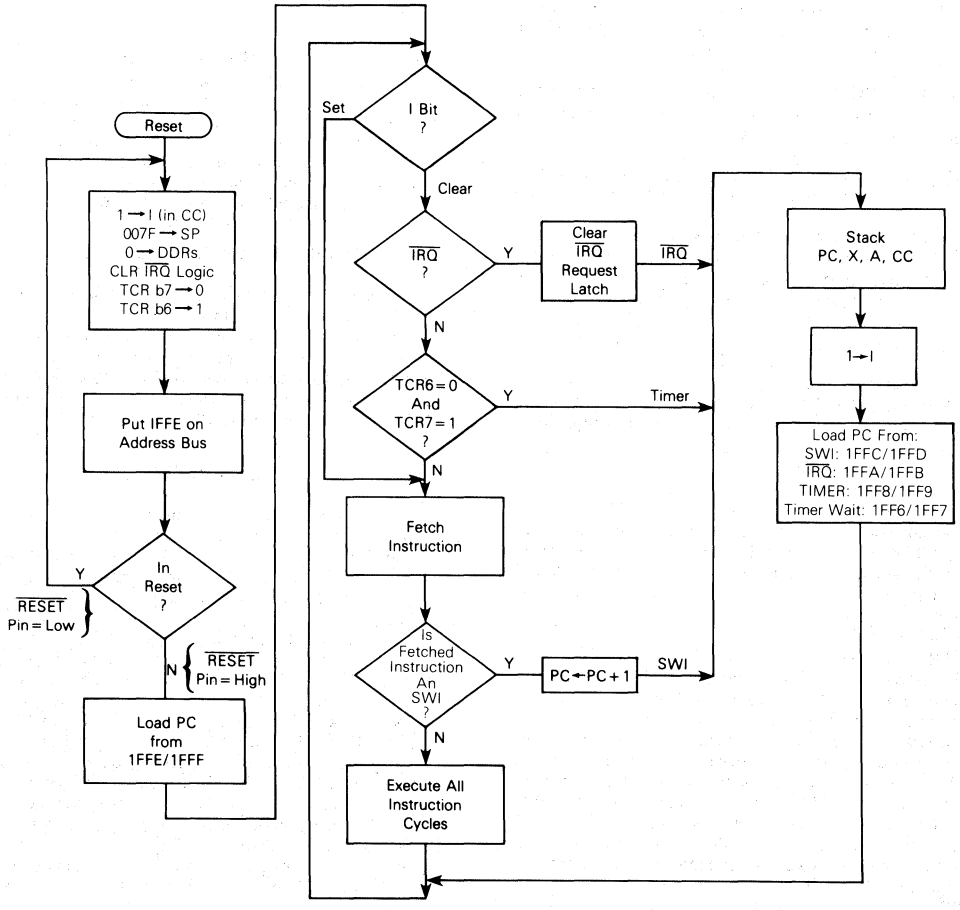


TABLE 3 — INTERRUPT INSTRUCTION EXECUTION PRIORITY AND VECTOR ADDRESS

(a) I Bit Set

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
SWI	2	$1FFC-$1FFD

Note: $\overline{\text{IRQ}}$ and TIMER Interrupts are not executed when the I bit is set; therefore, they are not shown.

(b) I Bit Clear

Interrupt/Instruction	Priority	Vector Address
$\overline{\text{RESET}}$	1	$1FFE-$1FFF
$\overline{\text{IRQ}}$	2	$1FFA-$1FFB
Timer	3	$1FF8-$1FF9 $1FF6-$1FF7*
SWI	4	$1FFC-$1FFD

*The Timer vector address from the WAIT mode is $1FF6-$1FF7.

TIMER INTERRUPT

If the timer interrupt mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00 to set TCR7) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit (in the condition code register) is cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode in which case the contents of $1FF6 and $1FF7 specify the timer service routine address. Software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

The actual timer interrupt request can be delayed by controlling TCR6 (interrupt mask bit). If TCR6 is programmed to a logic one, no interrupt is generated even if TCR7 (interrupt request bit) is set. Then, TCR6 can be programmed (after a specific time) to a logic zero to generate the actual timer interrupt request.

EXTERNAL INTERRUPT

If the interrupt mask bit of the condition code register has been cleared and the external interrupt pin ($\overline{\text{IRQ}}$) has gone low, then the external interrupt is recognized. The action of the external interrupt is identical to the timer interrupt with the exception that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of $1FFA and $1FFB. Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive only trigger are available as a mask option register (MOR) controlled programmable option. Figure 14 shows both a functional and mode timing diagram for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first method shows single pulses on the interrupt line spaced far enough apart to be serviced.

The minimum time between pulses is a function of the length of the interrupt service routine. Once a pluse occurs, the next pulse should not occur until the MPU software has exited the routine (an RTI occurs). This time (t_{ILIL}) is obtained by adding 20 instruction cycles (t_{cyc}) to the total number of cycles it takes to complete the service routine including the RTI instruction; refer to Figure 14. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.

NOTE

The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

SOFTWARE INTERRUPT

The software interrupt (SWI) is an executable instruction. The action of the software interrupt instruction is similar to the hardware interrupts. The software interrupt is executed regardless of the state of the interrupt mask bit in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 13 for interrupt and instruction processing flowchart.

LOW-POWER MODES

STOP

The STOP instruction places the MC1468705G2 in its lowest power consumption mode. In the STOP mode the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 15.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the condition code register is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged.

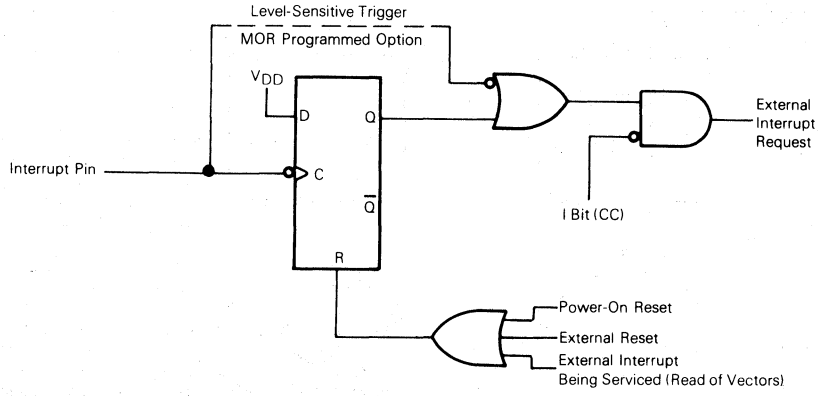
WAIT

The WAIT instruction places the MC1468705G2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode. In the WAIT mode, the internal clock is disabled from all internal circuitry except for the timer; refer to Figure 16. Thus, all internal processing is halted; however, the timer continues to count normally.

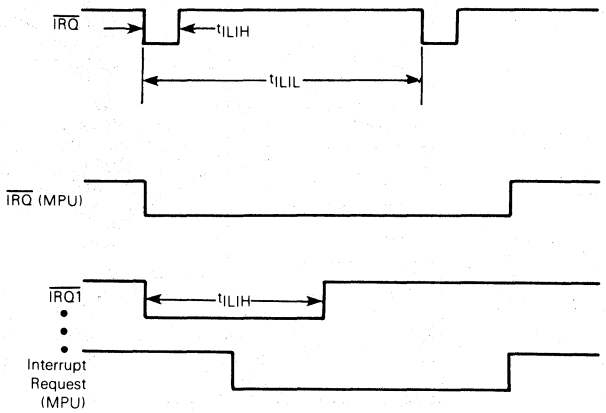
During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

FIGURE 14 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram



Edge-Sensitive Trigger Condition

The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

Level-Sensitive Trigger Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

Normally used with Wire-ORed Connection

FIGURE 15 — STOP FUNCTION FLOWCHART

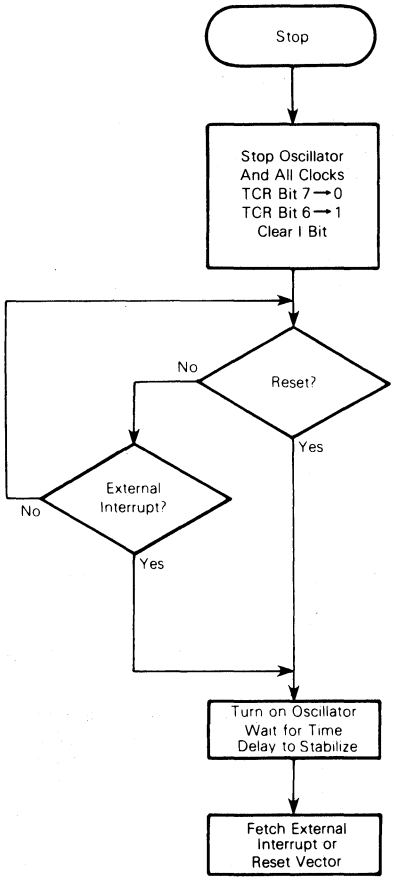
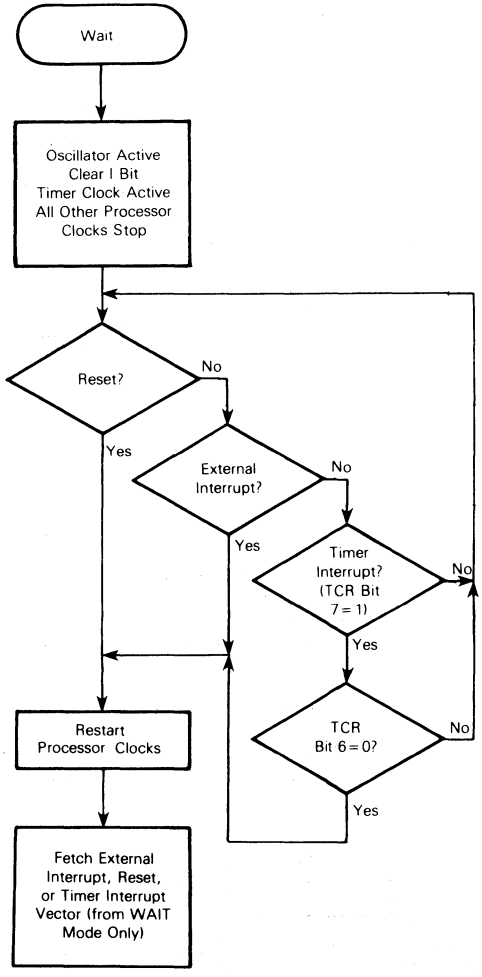


FIGURE 16 — WAIT FUNCTION FLOWCHART



3

MODES OF OPERATION

The MC1468705G2 has two modes of operation. These modes are the normal (single-chip) mode and the bootstrap mode (firmware used to program the EPROM). These two modes are entered as described below.

SINGLE-CHIP MODE

The normal operational mode of the part is the single-chip mode. The single-chip mode will be entered if the following conditions are satisfied: (1) the $\overline{\text{RESET}}$ line is brought low, (2) the $\overline{\text{IRQ}}$ pin is within its normal operational range ($V_{SS} - V_{DD}$), and (3) the V_{PP} pin is connected to V_{SS}. The next rising edge of the $\overline{\text{RESET}}$ pin then causes the part to enter the single-chip mode.

BOOTSTRAP MODE

The bootstrap mode is entered if certain conditions are met on the TIMER, $\overline{\text{IRQ}}$, and $\overline{\text{RESET}}$ pins. A negative voltage (V_{IRQP}) must be present on the $\overline{\text{IRQ}}$ pin. This value is latched internally on the rising edge of the external $\overline{\text{RESET}}$ pin. Also V_{DD} should be applied to the TIMER pin. The high state of the TIMER pin is then latched internally on the rising edge of the $\overline{\text{RESET}}$ pin. Refer to Figure 17.

TIMER

The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 18 contains a block diagram of the timer. The counter may be loaded under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit (i.e., bit 7 of the timer control register, TCR) is set. Then, if the timer interrupt is not masked (i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared) the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer vector address from locations $1FF8 and $1FF9 (or $1FF6 and $1FF7 if in the WAIT mode) in order to begin servicing; refer to **INTERRUPTS.**

FIGURE 17 — BOOTSTRAP MODE

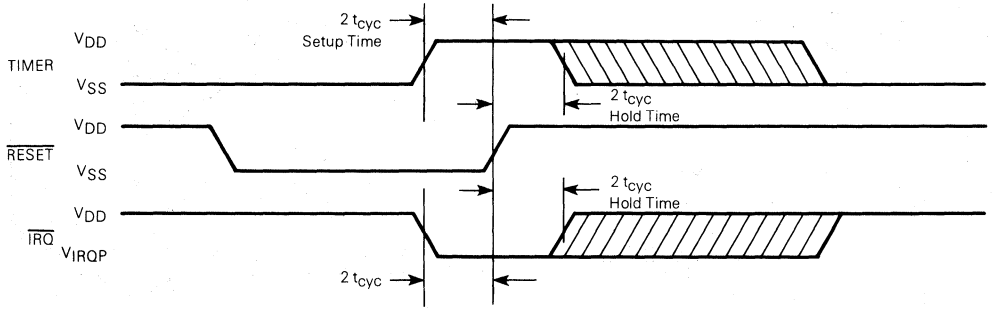
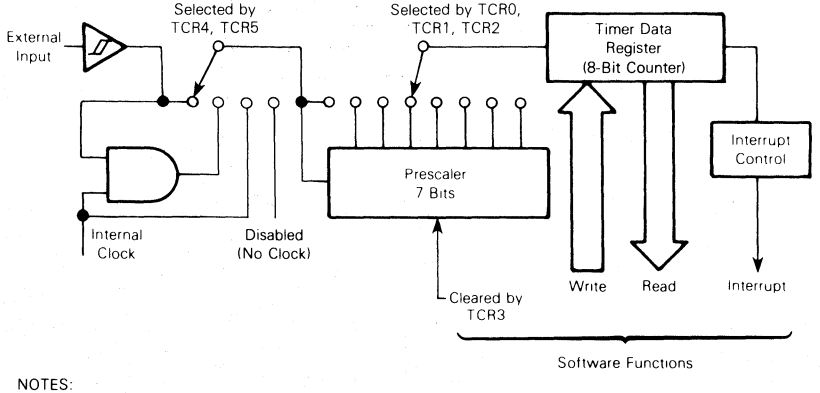


FIGURE 18 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register (8-bit counter) are clocked on the falling edge of the internal clock or external input.
2. The timer data register counts down continuously.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle, and do not change during the read. The timer interrupt request bit (TCR7) remains set until cleared by the software. If the timer interrupt request bit (TCR7) is cleared before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all zeros by the write operation into TCR when bit 3 of the written data equals a logic one. This allows for truncation-free counting.

The timer input can be configured for three different operating modes plus a disable mode, depending on the value written to the TCR4 and TCR5 timer control register bits. Refer to **TIMER CONTROL REGISTER.**

Figure 18 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction affect the state of the counter.

TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a zero, the input to the timer is from an internal clock and the TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock. During a WAIT instruction, the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR4 = 1 and TCR5 = 0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock through for the duration of the pulse. The resolution of the count in this mode is plus or minus one clock cycle; therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

If TCR4 = 0 and TCR5 = 1, then all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR4 = 1 and TCR5 = 1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The timer can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts. The counter is clocked on the falling edge of the external signal.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0	
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0	$0009

All bits in this register except bit 3 are Read/Write bits.

TCR7 — Timer interrupt request bit: bit used to indicate the timer interrupt when it is logic one.
1 — Set whenever the counter decrements to zero, or under program control.
0 — Cleared on external reset, power-on reset, STOP instruction, or program control.

TCR6 — Timer interrupt mask bit: when this bit is a logic one it inhibits the timer interrupt to the processor.
1 — Set on external reset, power-on reset, STOP instruction, or program control.
0 — Cleared under program control.

TCR5 — External or internal bit: selects the input clock source to be either the external TIMER pin or the internal clock. (Unaffected by reset.)
1 — Select external clock source.
0 — Select internal clock source (period = t_{CYC}).

TCR4 — External enable bit: control bit used to enable the external TIMER pin. (Unaffected by reset.)
1 — Enable external TIMER pin.
0 — Disable external TIMER pin.

TCR5	TCR4	
0	0	Internal clock to timer
0	1	AND of internal clock and TIMER pin to timer
1	0	Inputs to timer disabled
1	1	TIMER pin to timer

TCR3 — Timer prescaler reset bit: writing a one to this bit resets the prescaler to zero. A read of this location always indicates a zero. (Unaffected by reset.)

TCR2, TCR1, TCR0 — Prescaler select bits: decoded to select one of eight outputs of the prescaler. (Unaffected by reset.)

Prescaler			
TCR2	TCR1	TCR0	Result
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

OSCILLATOR AND INTERRUPT OPTIONS

The MC1468705G2 oscillator and interrupt options are implemented as an EPROM byte at address $1FF5 and are EPROM programmable. Selection of these programmable options is discussed below.

MASK OPTION REGISTER (MOR)

7	6	5	4	3	2	1	0	
CLK	DIV	0	INT	0	0	0	0	$1FF5

A discussion of the function of each bit is as follows.

B7, CLK – Determines the Clock Oscillator
0 – Crystal Oscillator.
1 – RC Oscillator.

B6, DIV – Determines Division of Clock Oscillator
0 – Divide-by-4 Oscillator Clock.
1 – Divide-by-2 Oscillator Clock.

B4, INT – Determines type of Interrupt Trigger Input
0 – Edge-Sensitive Triggered only.
1 – Edge-Sensitive or Level-Sensitive Triggered.

NOTE

Bits 0, 1, 2, 3, and 5 in the MOR must be programmed to zero.

The EPROM in the erased state will read all zeros. While in the bootstrap mode, the MC1468705G2 will operate under crystal oscillator and divide-by-4 options regardless of how the MOR is actually programmed; however, the interrupt trigger input will remain as programmed in the MOR. When the MC1468705G2 is in the single-chip mode and completes a power-on reset, the MCU operation is set up per the mask option register (MOR). The state of the V_{PP} pin does not affect the MOR controlled options.

ERASING THE EPROM

The MC1468705G2 EPROM can be erased by exposure to high-intensity ultraviolet (UV) light with a wavelength of 2537 angstroms. The recommended integrated dose (UV intensity x exposure time) is 15 Ws/cm^2. The lamps should be used without shortwave filters and the MC1468705G2 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC1468705G2 EPROM to the zero state. Data is then entered by programming ones into the desired bit locations.

CAUTION

Be sure that the EPROM window is shielded from light with an opaque cover at all times except when erasing. This protects both the EPROM and light-sensitive nodes.

PROGRAMMING FIRMWARE

A bootstrap program in ROM allows the MC1468705G2 to program its own internal EPROM. The alternate vectoring used to implement the self-check in the MC1468705G2 is used to start execution of this program.

When the V_{PP} voltage is placed on pin 3 (provided pin 2 has V_{IROP} applied and pin 37 has +5 V applied), the bootstrap program is executed and the MC1468705G2 programs itself. This ability to program itself is a function of the external hardware and the interaction between the internal hardware and the firmware. The amount of time for programming is determined by a value stored in the timer data register by the internal firmware. When the part is placed in the program and verify mode, the bootstrap vector will be fetched and the bootstrap firmware will start to execute.

Note that an MCM68764 (or MCM68766) UV EPROM must be programmed first with the exact duplicate of the information that is to be transferred to the MC1468705G2. Non-EPROM addresses are ignored by the on-chip ROM bootstrap. Since the MC1468705G2 and the MCM68764 (or MCM68766) EPROM are to be inserted and removed from the circuit, they should be mounted in sockets. Additionally, the precautions below should be observed (refer to Figures 19 and 20).

NOTE

The advanced programming information provided below applies to MC1468705G2 EPROM MCUs which were manufactured using a mask set other than the MJ3 series. For programming information regarding the MJ3 series of mask sets, consult Motorola Engineering Bulletin EB-110.

Figure 19 illustrates the memory location of the MCM68764 EPROM which corresponds to the equivalent memory in the MC1468705G2. Note that the MCM68764 memory locations which correspond to RAM locations or unused EPROM or ROM locations, in the MC1468705G2, may be programmed as either $00 or $FF (don't care).

CAUTION

Be sure that S1 is open and S2 is closed when inserting the MC1468705G2 and MCM68764 EPROMs into their respective sockets. This ensures that power is not applied and $\overline{\text{RESET}}$ is held low while inserting the devices.

To program the MC1468705G2, open S3 (to select the programming and verify mode) and then close S1 (to apply the proper voltages for the V_{DD}, TIMER, and $\overline{\text{IRQ}}$ pins). Next, open S2 (to remove reset and supply V_{PP}). When the reset cycle is complete, the internal ROM program initiates transfer of the external EPROM pattern one byte for each EPROM location. The MC1468705G2 bootstrap provides the address (A0-A12) and enable (TSC/$\overline{E}$) signals to permit complete self-programming. At the start of the data transfer from the MCM68764 EPROM, the programming LED (DS2) is turned on and remains on throughout the programming sequence. After completion of the programming sequence, the programming LED turns off. Transfer of the entire MCM68764 EPROM content requires approximately 200 seconds. The internal timer data register is then cleared and the loop is repeated to verify that the programmed data is precisely the same as the incoming data from the MCM68764 EPROM; if so, the verified LED is turned on. If the verified LED does not turn on, the exact program has not been loaded from the MCM68764 to the MC1468705G2, indicating a possible defect. Close S2 and open S1 prior to removing any device from its socket.

CAUTION

Once the MC1468705G2 is programmed and connected for normal operation, be sure that V_{PP} (pin 3) is connected directly to V_{SS}.

FIGURE 19 — MC1468705G2 MEMORY MAPPING ONTO MCM68764

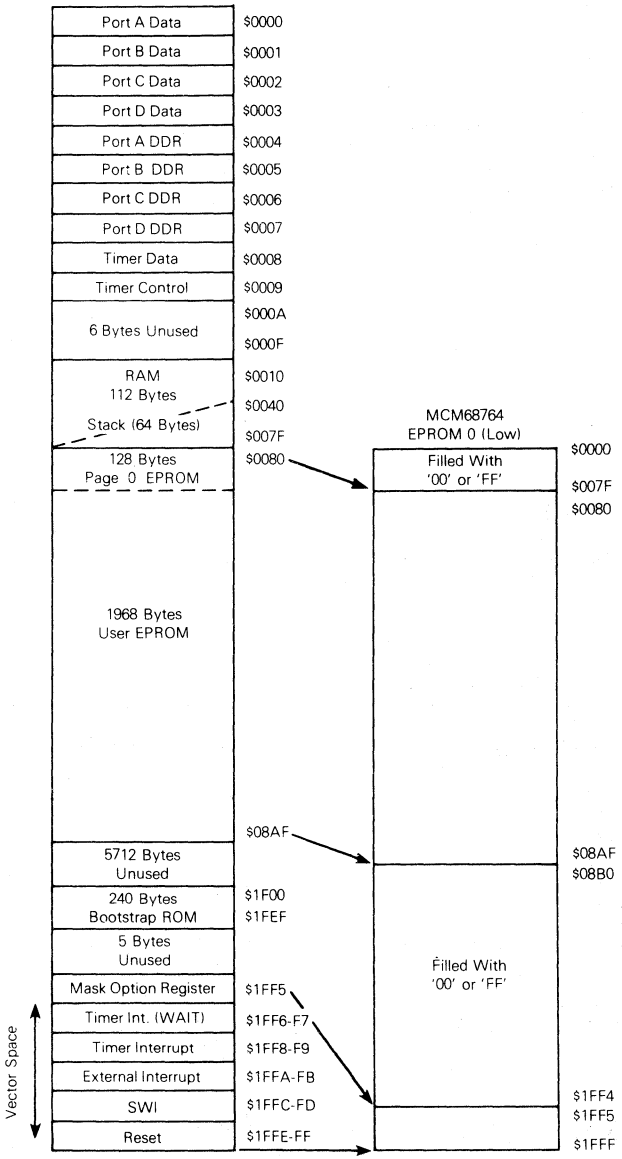
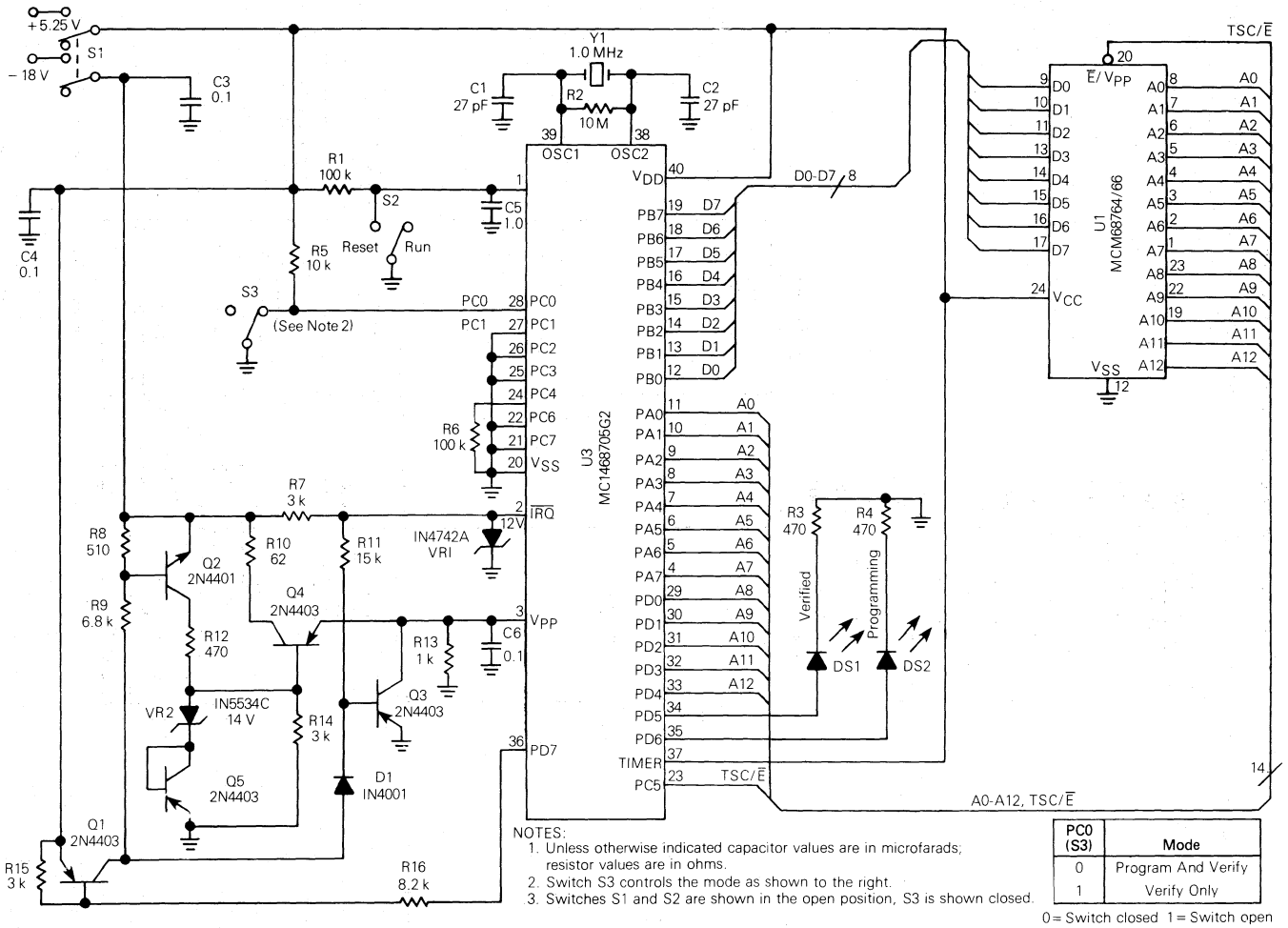


FIGURE 20 — PROGRAMMING CONNECTIONS



NOTES:
1. Unless otherwise indicated capacitor values are in microfarads; resistor values are in ohms.
2. Switch S3 controls the mode as shown to the right.
3. Switches S1 and S2 are shown in the open position, S3 is shown closed.

PC0 (S3)	Mode
0	Program And Verify
1	Verify Only

0 = Switch closed 1 = Switch open

INSTRUCTION SET

The MCU has a set of 61 basic instructions. 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.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The operand for the jump unconditional (JMP) and jump to subroutine (JSR) instructions is the program counter. Refer to Table 4.

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. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS

Most branch instructions test the state of the condition code register and if certain criteria are met, a branch is executed. This adds an offset between −127 and +128 to the current program counter. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 128 bytes of the memory space (where all port registers, port DDRs, timer, timer control, and on-chip RAM reside). Bit manipulation in the EPROM mapped area will not affect data in the EPROM. An additional feature allows the software to test and branch on the state of any bit within the first 256 locations. The bit set, bit clear, and bit test and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is automatically placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

OPCODE MAP

Table 9 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 10.

ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One and two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 10 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 9.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the byte address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of" the location or register referred to; e.g., (PC) indicates the contents of the location pointed to by the PC. An arrow indicates "is replaced by", and a colon indicates concatenations, refer to the **M6805 HMOS/M146805 CMOS Family Microcomputer/Microprocessor User's Manual.**

INHERENT

In inherent instructions, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediately following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

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 lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers, and 128 bytes of on-chip ROM. Direct addressing is efficient in both memory and time.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the most efficient addressing mode.

$$EA = (PC + 1):(PC + 2);\ PC \leftarrow PC + 3$$
$$\text{Address Bus High} \leftarrow (PC + 1);\ \text{Address Bus Low} \leftarrow (PC + 2)$$

INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or to address a frequently referenced RAM or I/O location.

$$EA = X;\ PC \leftarrow PC + 1$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

		Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

TABLE 5 — READ-MODIFY-WRITE INSTRUCTIONS

		Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Function	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (2's Complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate Left Thru Carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate Right Thru Carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical Shift Left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical Shift Right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic Shift Right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for Negative or Zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5

TABLE 6 — BRANCH INSTRUCTIONS

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	3
Branch IFF Higher	BHI	22	2	3
Branch IFF Lower or Same	BLS	23	2	3
Branch IFF Carry Clear	BCC	24	2	3
(Branch IFF Higher or Same)	(BHS)	24	2	3
Branch IFF Carry Set	BCS	25	2	3
(Branch IFF Lower)	(BLO)	25	2	3
Branch IFF Not Equal	BNE	26	2	3
Branch IFF Equal	BEQ	27	2	3
Branch IFF Half Carry Clear	BHCC	28	2	3
Branch IFF Half Carry Set	BHCS	29	2	3
Branch IFF Plus	BPL	2A	2	3
Branch IFF Minus	BMI	2B	2	3
Branch IFF Interrupt Mask Bit is Clear	BMC	2C	2	3
Branch IFF Interrupt Mask Bit is Set	BMS	2D	2	3
Branch IFF Interrupt Line is Low	BIL	2E	2	3
Branch IFF Interrupt Line is High	BIH	2F	2	3
Branch to Subroutine	BSR	AD	2	6

TABLE 7 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9 — MC1468705G2 CMOS INSTRUCTION SET OPCODE MAP

Hi / Low	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						Hi / Low
	BTB 0 0000	BSC 1 0001	REL 2 0010	DIR 3 0011	INH 4 0100	INH 5 0101	IX1 6 0110	IX 7 0111	INH 8 1000	INH 9 1001	IMM A 1010	DIR B 1011	EXT C 1100	IX2 D 1101	IX1 E 1110	IX F 1111	
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEG 3 1 INH	NEG 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL								SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for Address Modes

INH	Inherent
A	Accumulator
X	Index Register
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

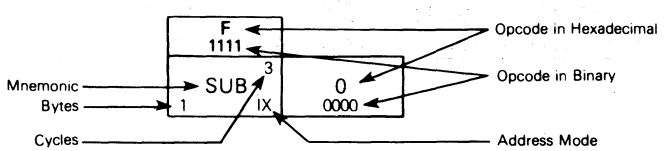


TABLE 10 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	•	Λ	Λ	Λ
AND		X	X	X		X	X	X			•	•	Λ	Λ	•
ASL	X		X			X	X				•	•	Λ	Λ	Λ
ASR	X		X			X	X				•	•	Λ	Λ	Λ
BCC					X						•	•	•	•	•
BCLR									X		•	•	•	•	•
BCS					X						•	•	•	•	•
BEQ					X						•	•	•	•	•
BHCC					X						•	•	•	•	•
BHCS					X						•	•	•	•	•
BHI					X						•	•	•	•	•
BHS					X						•	•	•	•	•
BIH					X						•	•	•	•	•
BIL					X						•	•	•	•	•
BIT		X	X	X		X	X	X			•	•	Λ	Λ	•
BLO					X						•	•	•	•	•
BLS					X						•	•	•	•	•
BMC					X						•	•	•	•	•
BMI					X						•	•	•	•	•
BMS					X						•	•	•	•	•
BNE					X						•	•	•	•	•
BPL					X						•	•	•	•	•
BRA					X						•	•	•	•	•
BRN					X						•	•	•	•	•
BRCLR										X	•	•	•	•	Λ
BRSET										X	•	•	•	•	Λ
BSET									X		•	•	•	•	•
BSR					X						•	•	•	•	•
CLC	X										•	•	•	•	0
CLI	X										•	0	•	•	•
CLR	X		X			X	X				•	•	0	1	•
CMP		X	X	X		X	X	X			•	•	Λ	Λ	Λ
COM	X		X			X	X				•	•	Λ	Λ	1
CPX		X	X	X		X	X	X			•	•	Λ	Λ	Λ
DEC	X		X			X	X				•	•	Λ	Λ	•
EOR		X	X	X		X	X	X			•	•	Λ	Λ	•
INC	X		X			X	X				•	•	Λ	Λ	•
JMP			X	X		X	X	X			•	•	•	•	•
JSR			X	X		X	X	X			•	•	•	•	•
LDA		X	X	X		X	X	X			•	•	Λ	Λ	•
LDX		X	X	X		X	X	X			•	•	Λ	Λ	•
LSL	X		X			X	X				•	•	Λ	Λ	Λ
LSR	X		X			X	X				•	•	0	Λ	Λ
NEG	X		X			X	X				•	•	Λ	Λ	Λ
NOP	X										•	•	•	•	•
ORA		X	X	X		X	X	X			•	•	Λ	Λ	•
ROL	X		X			X	X				•	•	Λ	Λ	Λ
ROR	X		X			X	X				•	•	Λ	Λ	Λ
RSP	X										•	•	•	•	•
RTI	X										?	?	?	?	?
RTS	X										•	•	•	•	•
SBC		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SEC	X										•	•	•	•	1
SEI	X										•	1	•	•	•
STA			X	X		X	X	X			•	•	Λ	Λ	•
STOP	X										•	0	•	•	•
STX			X	X		X	X	X			•	•	Λ	Λ	•
SUB		X	X	X		X	X	X			•	•	Λ	Λ	Λ
SWI	X										•	1	•	•	•
TAX	X										•	•	•	•	•
TST	X		X			X	X				•	•	Λ	Λ	•
TXA	X										•	•	•	•	•
WAIT	X										•	0	•	•	•

Condition Code Symbols

H	Half Carry (From Bit 3)	Λ	Test and Set if True, Cleared Otherwise
I	Interrupt Mask	•	Not Affected
N	Negative (Sign Bit)	?	Load CC Register From Stack
Z	Zero	0	Cleared
C	Carry/Borrow	1	Set

INDEXED, 8-BIT OFFSET

Here the EA is obtained by adding the contents of the byte following the opcode to that of the index register; therefore, the operand is located anywhere within the lowest 511 memory locations. For example, this mode of addressing is useful for selecting the mth element in an n element table. All instructions are two bytes. The contents of the index register (X) is not changed. The contents of (PC + 1) is an unsigned 8-bit integer. One byte offset indexing permits look-up tables to be easily accessed in either RAM or ROM.

EA = X + (PC + 1); PC ← PC + 2
Address Bus High ← K; Address Bus Low ← X + (PC + 1)

Where:
K = The carry from the addition of X + (PC + 1)

INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed 8-bit offset, except that this three byte instruction allows tables to be anywhere in memory (e.g., jump tables in ROM). As with direct and extended, the M6805 assembler determines the most efficient form of indexed offset; 8-or 16-bit. The contents of the index register are not changed.

EA = X + [(PC + 1):(PC + 2))]; PC – PC + 3
Address Bus High ← (PC + 1) + K;
Address Bus Low ← X + (PC + 2)

Where:
K = The carry from the additon of X + (PC + 2)

RELATIVE

Relative addressing is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) are added to the PC if and only if the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is limited to the range of – 126 to + 129 bytes from the branch instruction opcode location. The Motorola assembler calculates the proper offset and checks to see if it is within the span of the branch.

EA = PC + 2 + (PC + 1); PC ← EA if branch taken;
otherwise, EA = PC ← PC + 2

BIT SET/CLEAR

Direct addressing and bit addressing are combined in instructions which set and clear individual memory and I/O bits. In the bit set and clear instructions, the byte is specified as a direct address in the location following the opcode. The first 256 addressable locations are thus accessed. The bit to be modified within that byte is specified with the first three bits of the opcode. The bit set and clear instructions occupy two bytes, one for the opcode (including the bit number) and the other to address the byte which contains the bit of interest.

EA = (PC + 1); PC ← PC + 2
Address Bus High ← 0; Address Bus Low ← (PC + 1)

BIT TEST AND BRANCH

Bit test and branch is a combination of direct addressing, bit set/clear addressing, and relative addressing. The actual bit to be tested, within the byte, is specified within the low order nibble of the opcode. The address of the data byte to be tested is located via a direct address in the location following the opcode byte (EA1). The signed relative 8-bit offset is in the third byte (EA2) and is added to the PC if the specified bit is set or cleared in the specified memory location. This single three byte instruction allows the program to branch based on the condition of any bit in the first 256 locations of memory.

EA1 = (PC + 1)
Address Bus High ← 0; Address Bus Low ← (PC + 1)
EA2 = PC + 3 + (PC + 2); PC ← EA2 if branch taken;
otherwise, PC – PC + 3

MOTOROLA

TCA5600
TCF5600

Product Preview

UNIVERSAL MICROPROCESSOR POWER SUPPLY CONTROLLER

The TCA5600 is a versatile power supply control circuit for microprocessor based systems and mainly intended for automotive applications and battery powered instruments. To cover a wide range of applications, the device offers high circuit flexibility with minimum of external components.

Functions included in this IC are a temperature compensated voltage reference, on chip dc/dc converter, programmable and remote controlled voltage regulator, fixed 5.0 V supply voltage regulator with external PNP power device, undervoltage detection circuit, power-on RESET delay and watchdog feature for safe and hazard free microprocessor operations.

- 6.0 to 30 V Operation Range
- 2.5 V Reference Voltage Accessible for Other Tasks
- Fixed 5.0 V ± 4% Microprocessor Supply Regulator Including Current Limitation, Overvoltage Protection and Undervoltage Monitor
- Programmable 6.0 to 30 V Voltage Regulator Exhibiting High Peak Current (150 mA), Current Limiting and Thermal Protection
- Two Remote Inputs to Select the Regulator's Operation Mode: OFF, 5.0 V, 5.0 V Standby and Programmable Output Voltage
- Self Contained dc/dc Converter Fully Controlled By the Programmable Regulator to Guarantee Safe Operation Under All Working Conditions
- Programmable Power-On RESET Delay
- Watchdog Select Input
- Negative Edge Triggered Watchdog Input
- Low Current Consumption in the V_{CC1} Standby Mode
- All Digital Control Ports are TTL- and MOS-Compatible

APPLICATIONS INCLUDE

- Microprocessor Systems with E^2PROMs
- High Voltage Crystal and Plasma Displays
- Decentralized Power Supplies in Computer and Telecommunication Systems

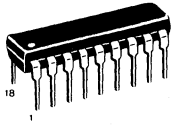
RECOMMENDED OPERATION CONDITIONS

Characteristic	Symbol	Min	Max	Unit
Power Supply Voltage	V_{CC1} V_{CC2}	5.0 5.5	30 30	V
Collector Current	I_C	—	800	mA
Output Voltage	V_{out2}	6.0	30	V
Reference Source Current	I_{ref}	0	2.0	mA

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

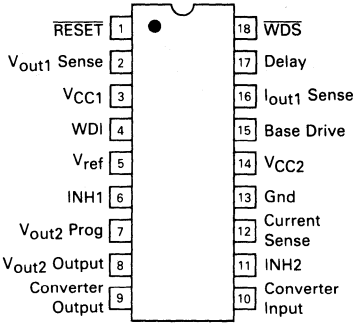
UNIVERSAL MICROPROCESSOR POWER SUPPLY CONTROLLER

SILICON MONOLITHIC INTEGRATED CIRCUITS



PLASTIC PACKAGE
CASE 707-02

PIN CONNECTIONS



(Top View)

ORDERING INFORMATION

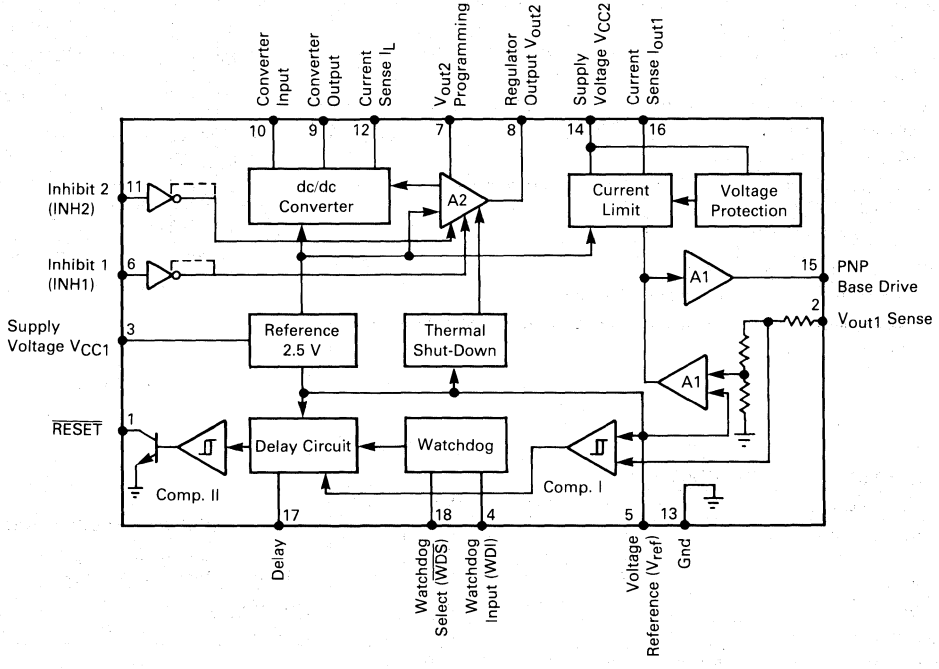
Device	Operating Junction Temperature Range	Package
TCA5600	0 to +125°C	Plastic DIP
TCF5600	−40 to +150°C	Plastic DIP

MAXIMUM RATINGS (T_A = +25°C unless otherwise noted, Note 1)

Rating	Symbol	Value	Unit
Power Supply Voltage (Pin 3, 14)	V_{CC1}, V_{CC2}	35	Vdc
Base Drive Current (Pin 15)	I_B	20	mA
Collector Current (Pin 10)	I_C	1.0	A
Forward Rectifier Current (Pin 10–Pin 9)	I_F	1.0	A
Logic Inputs INH1, INH2, $\overline{WDS}$ (Pin 6, 11, 18)	V_{INP}	−0.3 V to V_{CC1}	Vdc
Logic Input Current WDI (Pin 4)	I_{WDI}	±0.5	mA
Output Sink Current $\overline{RESET}$ (Pin 1)	$I_{\overline{RES}}$	10	mA
Analog Inputs (Pin 2) (Pin 7)	— —	−0.3 to 10 −0.3 to 5.0	V
Reference Source Current (Pin 5)	I_{ref}	5.0	mA
Power Dissipation (Note 2) T_A = +75°C TCA5600 T_A = +85°C TCF5600	P_D	 500 650	mW
Thermal Resistance (Junction to Air)	$R_{\theta JA}$	100	°C/W
Operating Temperature Range TCA5600 TCF5600	T_A	 0 to +75 −40 to +85	°C
Operating Junction Temperature TCA5600 TCF5600	T_J	 +125 +150	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

NOTES:
1. Values beyond which damage may occur.
2. Derate at 10 mW/°C for junction temperature above +75°C (TCA5600).
   Derate at 10 mW/°C for junction temperature above +85°C (TCF5600).

FUNCTIONAL BLOCK DIAGRAM



ELECTRICAL CHARACTERISTICS (V_{CC1} = V_{CC2} = 12 V; T_J = 25°C; I_{ref} = 0; I_{out1} = 0 (Note 3); R_{SC} = 0.5 Ω; INH1 = "High"; INH2 = "High"; $\overline{WDS}$ = "High"; I_{out2} = 0 (Note 4); if not otherwise specified)

Characteristic	Figure	Symbol	Min	Typ	Max	Unit
REFERENCE SECTION						
Nominal Reference Voltage	1	$V_{ref\ nom}$	2.42	2.5	2.58	V
Reference Voltage I_{ref} = 0.5 mA, $T_{low} \leq T_J \leq T_{high}$ (Note 5), 6.0 V ≤ V_{CC1} ≤ 18 V		V_{ref}	2.4	—	2.6	V
Line Regulation (6.0 V ≤ V_{CC2} ≤ 18 V)		Reg_{line}	—	2.0	15	mV
Average Temperature Coefficient $T_{low} \leq T_J \leq T_{high}$ (Note 5)	2	$\frac{\Delta V_{ref}}{\Delta T_J}$	—	—	+/−0.5	mV/°C
Ripple Rejection Ratio f = 1.0 kHz, V_{sin} = 1.0 V_{pp}	3	RR	60	70	—	dB
Output Impedance 0 ≤ I_{ref} ≤ 2.0 mA		Z_O	—	1.0	—	Ohm
Standby Current Consumption V_{CC2} = Open	4	I_{CC1}	—	3.0	—	mA

NOTES:
3. The external PNP power transistor satisfies the following minimum specifications:
   $h_{FE} \geq 60$ at I_C = 500 mA and V_{CE} = 5.0 V; $V_{CE(sat)} \leq 300$ mV at I_B = 10 mA and I_C = 300 mA
4. Regulator V_{out2} programmed for nominal 24 V output by means of R4, R5 (see Figure 1)
5. T_{low} = 0°C for TCA5600; T_{low} = −40°C for TCF5600.
   T_{high} = 125°C for TCA5600; T_{high} = 150°C for TCF5600.

5.0 V MICROPROCESSOR VOLTAGE REGULATOR SECTION

Nominal Output Voltage		$V_{out1(nom)}$	4.8	5.0	5.2	V
Output Voltage 5.0 mA ≤ I_{out1} ≤ 300 mA, $T_{low} \leq T_J \leq T_{high}$ (Note 5) 6.0 V ≤ V_{CC2} ≤ 18 V	5 6	V_{out1}	4.75	—	5.25	V
Line Regulation (6.0 V ≤ V_{CC2} ≤ 18 V)		Reg_{line}	—	10	50	mV
Load Regulation (5.0 mA ≤ I_{out1} ≤ 300 mA)		Reg_{load}	—	20	100	mV
Base Current Drive (V_{CC2} = 6.0 V, V_{15} = 4.0 V)		I_B	10	15	—	mA
Ripple Rejection Ratio f = 1.0 kHz, V_{sin} = 1.0 V_{pp}	3	RR	50	65	—	dB
Undervoltage Detection Level (R_{SC} = 5.0 Ω)	7	V_{low}	4.5	0.93 x V_{out1}	—	V
Current Limitation Threshold (R_{SC} = 5.0 Ω)		V_{RSC}	210	250	290	mV
Average Temperature Coefficient $T_{low} \leq T_J \leq T_{high}$ (Note 5)		$\frac{\Delta V_{out1}}{\Delta T_J}$	—	—	±1.0	mV/°C

Characteristic	Figure	Symbol	Min	Typ	Max	Unit
PROGRAMMABLE VOLTAGE REGULATOR SECTION (Note 6)						
Nominal Output Voltage		$V_{out2(nom)}$	23	24	25	V
Output Voltage 1.0 mA ≤ I_{out2} ≤ 100 mA, T_{low} ≤ T_J ≤ T_{high} (Notes 5, 7)	8	V_{out2}	22.8	—	25.2	V
Load Regulation 1.0 mA ≤ I_{out2} ≤ 100 mA (Note 7)		Reg_{load}	—	40	200	mV
DC Output Current		I_{out2}	100	—	—	mA
Peak Output Current (Internally Limited)		$I_{out2\ p}$	150	200	—	mA
Ripple Rejection Ratio f = 20 kHz, V = 0.4 V_{pp}		RR	45	55	—	dB
Output Voltage (Fixed 5.0 V) 1.0 mA ≤ I_{out2} ≤ 20 mA, T_{low} ≤ T_J ≤ T_{high}, INH1 = "High" (Note 5)		$V_{out2(5.0\ V)}$	4.75	—	5.25	V
OFF State Output Impedance (INH2 = "Low")		R_{out1}	—	10	—	kΩ
Average Temperature Coefficient T_{low} ≤ T_J ≤ T_{high} (Note 5)		$\frac{\Delta V_{out2}}{\Delta T_J V_{out2}}$	—	—	±0.25	mV/°C V

NOTES:
6. V_9 = 28 V, INH1 = "Low" for this Electrical Characteristic section unless otherwise specified.
7. Pulse tested t_p ≤ 300 μs

DC/DC CONVERTER SECTION						
Collector Current Detection Level "High" R_C = 10 k "Low"	9	$V_{12}(H)$ $V_{12}(L)$	350 —	400 50	450 —	mV
Collector Saturation Voltage I_C = 600 mA (Note 7)	10	$V_{CE(sat)}$	—	—	1.6	V
Rectifier Forward Voltage Drop I_F = 600 mA (Note 7)	11	V_F	—	—	1.4	V
WATCHDOG AND $\overline{\text{RESET}}$ CIRCUIT SECTION						
Threshold Voltage "High" (static) "Low"		$V_{C5(H)}$ $V_{C5(L)}$	— —	2.5 1.0	— —	V
Current Source T_{low} ≤ T_J ≤ T_{high} (Note 5) Power-Up $\overline{\text{RESET}}$ Watchdog Time Out Watchdog $\overline{\text{RESET}}$		I_{C5}	 −1.8 — —	 −2.5 5xI_{C5} −50xI_{C5}	 −3.2 — —	μA
Watchdog Input Voltage Swing		V_{WDI}	—	—	±5.5	V
Watchdog Input Impedance		r_i	12	15	—	kΩ
Watchdog Reset Pulse Width (C8 = 1.0 nF) (Note 9)		t_p	—	—	10	μs
DIGITAL PORTS: $\overline{\text{WDS}}$, INH 1, INH 2, $\overline{\text{RESET}}$ (Note 8)						
Input Voltage Range		V_{INP}	—	—	−0.3 to V_{CC1}	V
Input HIGH Current 2.0 V ≤ V_{IH} ≤ 5.5 V 5.5 V ≤ V_{IH} ≤ V_{CC1}		I_{IH}	 — —	 — —	 100 150	μA
Input LOW Current −0.3 V ≤ V_{IL} ≤ 0.8 V for INH1, INH2, −0.3 V ≤ V_{IL} ≤ 0.4 V for $\overline{\text{WDS}}$		I_{IL}	—	—	−100	μA
Leakage Current Immunity (INH2, High "Z" State)	12	I_Z	±20	—	—	μA
Output LOW Voltage $\overline{\text{RESET}}$ (I_{OL} = 6.0 mA)		V_{OL}	—	—	0.4	V
Output HIGH Current $\overline{\text{RESET}}$ (V_{OH} = 5.5 V)		V_{OH}	—	—	20	μA

NOTES:
8. Temperature range T_{low} ≤ T_J ≤ T_{high} applies to this Electrical Characteristics section.
9. For test purposes, a negative pulse is applied to Pin 4 (−2.5 V ≥ V_4 ≥ −5.5 V).

TYPICAL CHARACTERISTICS

FIGURE 1 — REFERENCE VOLTAGE versus SUPPLY VOLTAGE

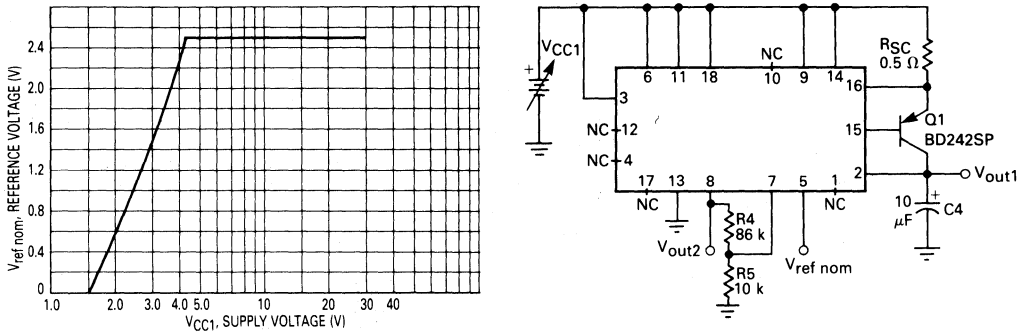


FIGURE 2 — REFERENCE STABILITY versus TEMPERATURE

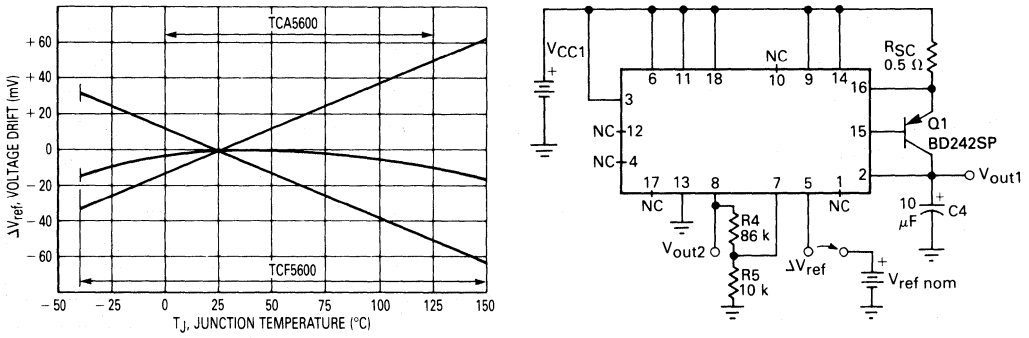


FIGURE 3 — RIPPLE REJECTION versus FREQUENCY

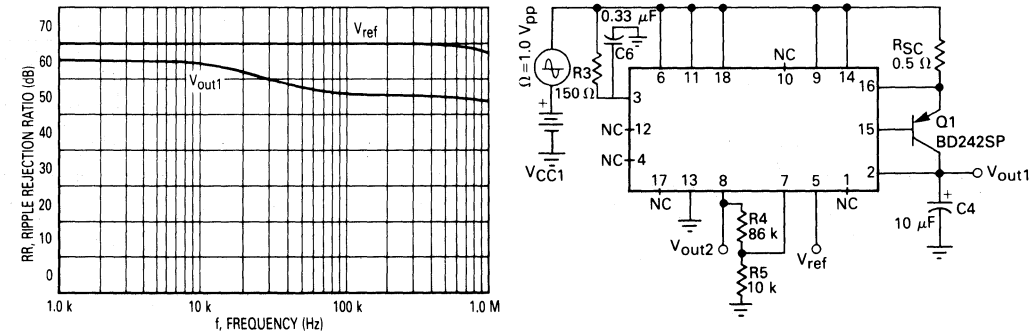
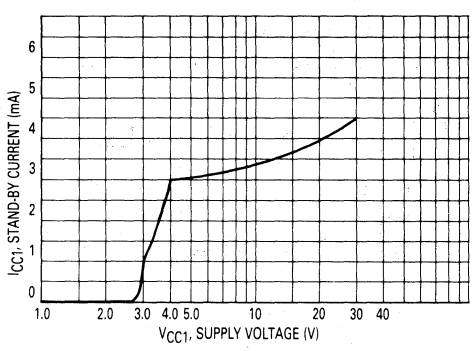
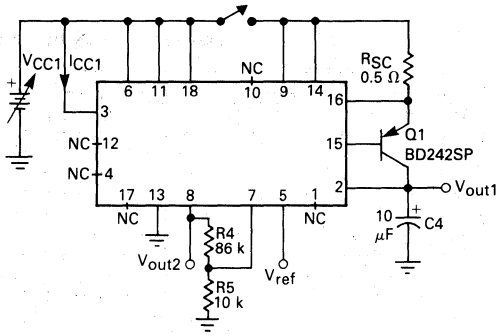


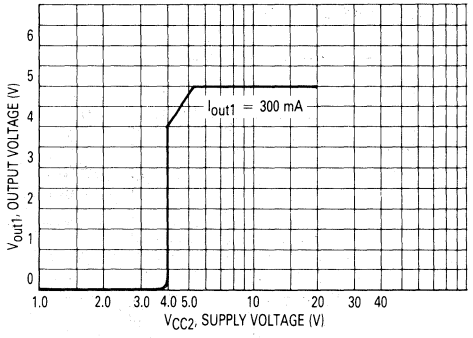
FIGURE 4 — STAND-BY CURRENT versus SUPPLY VOLTAGE





3

FIGURE 5 — POWER-UP BEHAVIOR OF THE 5.0 V REGULATOR



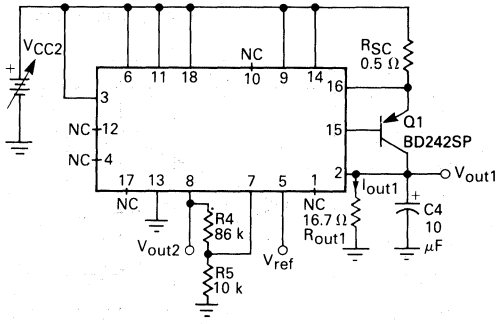
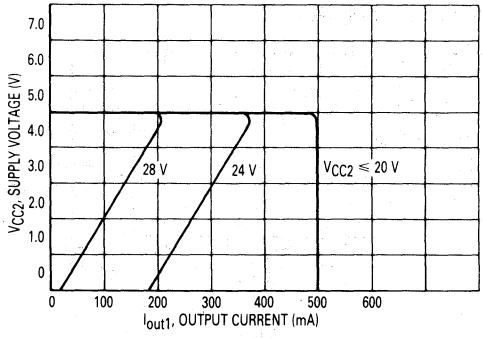


FIGURE 6 — FOLDBACK CHARACTERISTICS OF THE 5.0 V REGULATOR



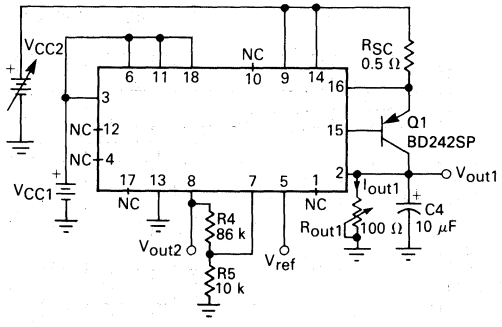


FIGURE 7 — UNDERVOLTAGE LOCKOUT CHARACTERISTICS

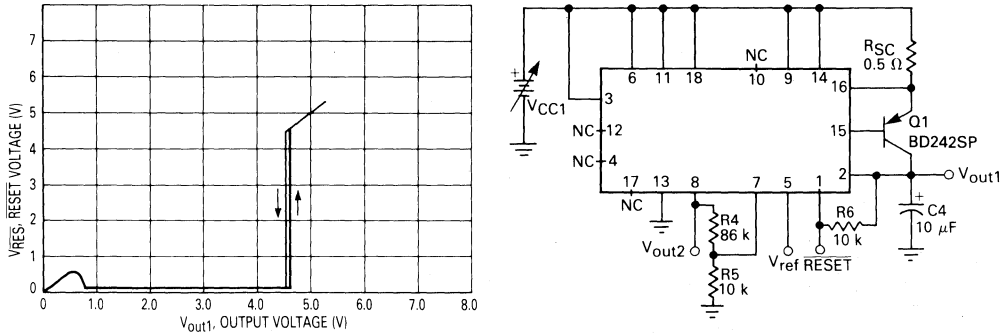


FIGURE 8 — OUTPUT CURRENT CAPABILITY OF THE PROGRAMMING REGULATOR

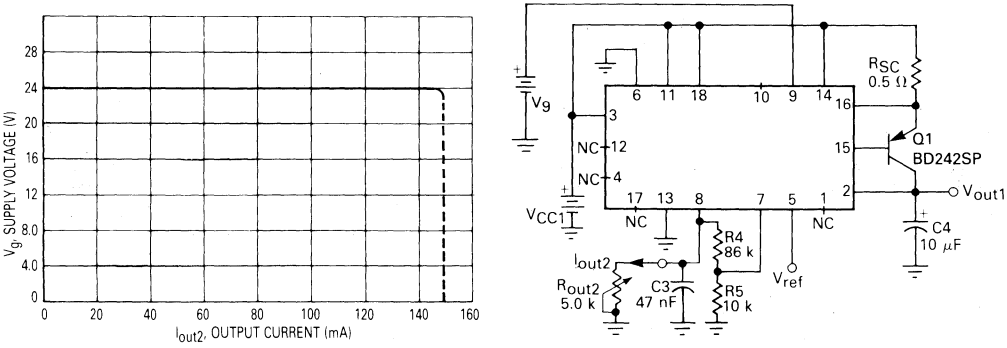


FIGURE 9 — COLLECTOR CURRENT DETECTION LEVEL

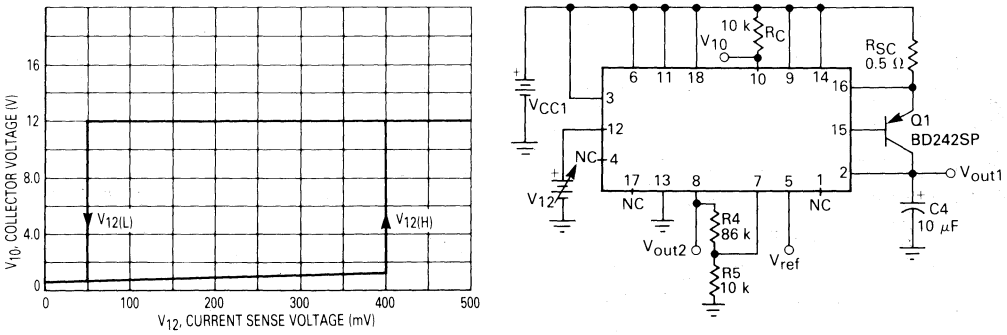


FIGURE 10 — POWER SWITCH CHARACTERISTICS

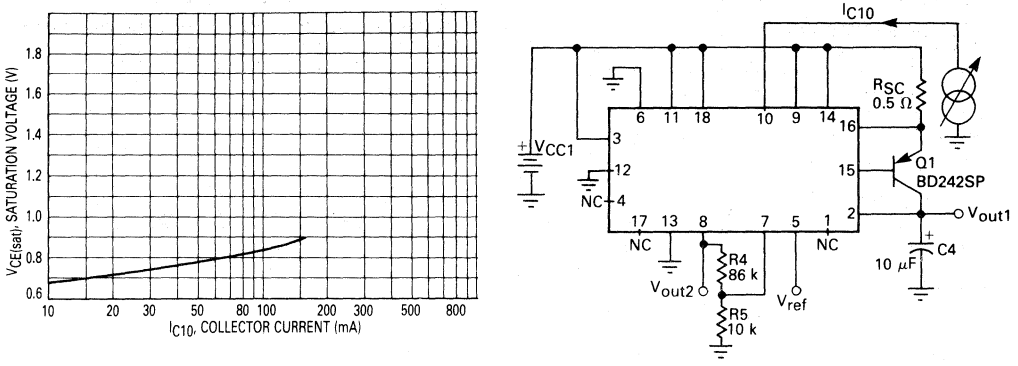


FIGURE 11 — RECTIFIER CHARACTERISTICS

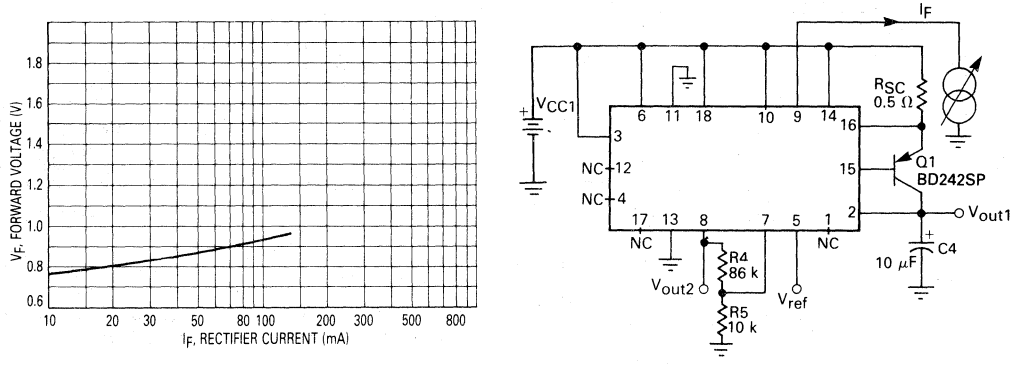
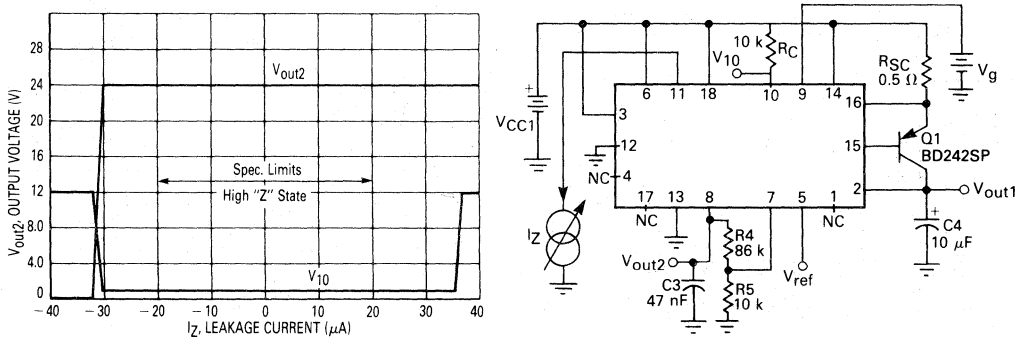


FIGURE 12 — INH 2 LEAKAGE CURRENT IMMUNITY

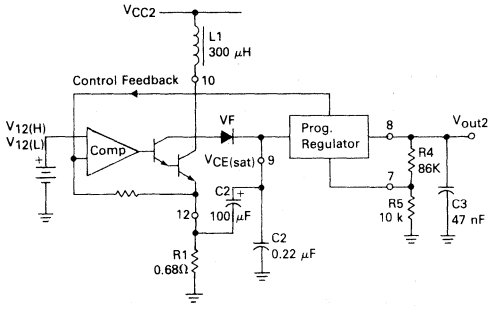


APPLICATIONS INFORMATION
(See Figure 18)

1. VOLTAGE REFERENCE V_{ref}
The voltage reference V_{ref} is based upon a highly stable bandgap voltage reference and is accessible on Pin 5 for additional tasks. This circuit part has its own supply connection on Pin 3 and is therefore able to operate in standby mode. The RC network R3, C6 improves the ripple rejection on both regulators.

2. DC/DC CONVERTER
The dc/dc converter performs according to the fly back principle and does not need a time base circuit. The maximum coil current is well defined by means of the current sensing resistor R1 under all working conditions (start-up phase, circuit overload, wide supply voltage range and extreme load current change). Figure 13 shows the simplified converter schematic:

FIGURE 13 — SIMPLIFIED CONVERTER SCHEMATIC



A simplified method on "how to calculate the coil inductance" is given below. The operation point at min. supply voltage (V_{CC2}) and max. output current (I_{out2}) for a fixed output voltage (V_{out2}) determines the coil data. Figure 14 shows the typical voltage and current wave forms on the coil L1 (coil losses neglected).

The equations (1) and (2) yield the respective coil voltage V_L- and V_L+ (see Figure 14):

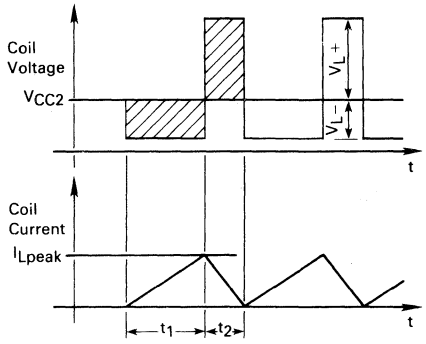
$$V_L+ = V_{out2} + \Delta V_{(Pin\ 9\ -\ Pin\ 8)} + V_F - V_{CC2} \quad (1)$$

$$V_L- = V_{CC2} - V_{CE(sat)} - V_{12(H)} \quad (2)$$

($\Delta V_{(Pin\ 9\ -\ Pin\ 8)}$: input/output voltage drop of the regulator, 2.5 V typical)

(V_F, $V_{CE(sat)}$, $V_{12(H)}$: see electrical characteristics)

FIGURE 14 — VOLTAGE AND CURRENT WAVEFORM ON THE COIL (not to scale)



The time ratio α for the charging time to dumping time is defined by equation (3):

$$\alpha = \frac{t_1}{t_2} = \frac{V_L+}{V_L-} \quad (3)$$

The coil charging time t_1 is found using equation (4):

$$t_1 = \frac{1}{(1 + \frac{1}{\alpha}) \cdot f} \quad (4)$$

(f : min. oscillation frequency which should be chosen above the audio frequency band (e.g. 20 kHz))

Knowing the dc output current I_{out2} of the programmable regulator, the peak coil current $I_{L(peak)}$ can now be calculated:

$$I_{L(peak)} = 2 \cdot I_{out2} \cdot (1 + \alpha) \quad (5)$$

The coil inductance L1 of the nonsaturated coil is given by equation (6):

$$L1 = \frac{t_1}{I_{L(peak)}} \cdot V_L- \quad (6)$$

The formula (6a) yields the current sensing resister R1 for a defined peak coil current $I_{L(peak)}$:

$$R1 = \frac{V_{12(H)}}{I_{L(peak)}} \quad (6a)$$

In order to limit the by-pass current through capacitor C7 during the energy dumping phase the value C2>>C7 should be implemented.

For all other operation conditions, the feedback signal from the programmable voltage regulator controls the activity of the converter.

3. PROGRAMMABLE VOLTAGE REGULATOR

This series voltage regulator is programmable by the voltage divider R4, R5 for a nominal output voltage 6.0 V ≤ V_{out2} ≤ 30 V.

$$R4 = \frac{(V_{out2} - V_{ref\ nom}) \cdot R5}{V_{ref\ nom}} \quad (7)$$

($R5 = 10\ k$, $V_{ref\ nom} = 2.5\ V$)

Current limitation and thermal shutdown capability are standard features of this regulator. The voltage drop $\Delta V_{(Pin\ 9\ -\ Pin\ 8)}$ across the series pass transistor generates the feedback signal to control the dc/dc converter (see Figure 13).

4. CONTROL INPUTS INH1, INH2

The dc/dc converter and/or the regulator V_{out2} are remote controllable through the TTL, MOS compatible inhibit inputs INH1 and INH2 where the latter is a 3-level detector (Logic "0", high impedance "Z", Logic "1"). Both inputs are setup to provide the following truth table:

FIGURE 15 — INH1, INH2 TRUTH TABLE

Mode	INH1	INH2	V_{out2}	dc/dc
1	0	0	OFF	INT
2	0	High "Z"	V_{out2}	ON
3	0	1	V_{out2}	INT
4	1	0	OFF	INT
5	1	High "Z"	5.0 V	ON
6	1	1	5.0 V	INT

INT: Intermittent operation of the converter means that the converter operates only if $V_{CC2} < V_{out2}$.

ON: The converter loads the storage capacitor C2 to its full charge (V_9 = 33 V), allowing fast response time of the regulator V_{out2} when addressed by the control software.

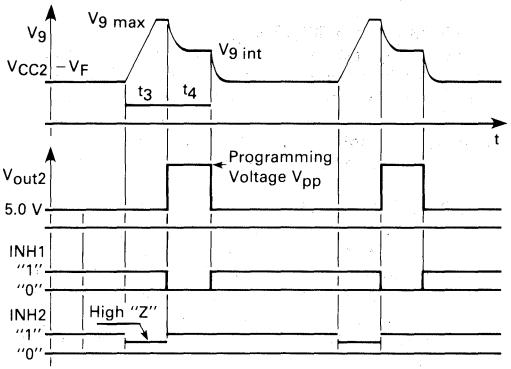
OFF: High impedance (internal resistor 10 k to ground)

Figure 16 represents a typical timing diagram for an E^2PROM programming sequence in a microprocessor based system. The high "Z" state enables the dc/dc converter to ramp during t_3 to the voltage V_9 at Pin 9 to a high level before the write cycle takes place in the memory.

5. MICROPROCESSOR SUPLY REGULATOR

Together with an external PNP power transistor (Q1), a 5.0 V supply exhibiting low voltage drop is obtained to power microprocessor systems and auxilliary circuits. Using a power Darlington with adequate heat sink in the output stage boosts the output current I_{out1} above 1 amp.

FIGURE 16 — TYPICAL E^2PROM PROGRAMMING SEQUENCE (not to scale)



The current limitation circuit measures the emitter current of Q1 by means of the sensing resistor R_{SC}.

$$R_{SC} = \frac{V_{RSC}}{I_E} \quad (8)$$

(I_E: emitter current of Q1)

(V_{RSC}: threshold voltage (see electrical characteristics))

The voltage protection circuit performs a fold-back characteristic above a nominal operating voltage $V_{CC2} \geq 18$ V.

6. DELAY AND WATCHDOG CIRCUIT

The under voltage monitor supervises the power supply V_{out1} and releases the delay circuit $\overline{RESET}$ as soon as the regulator output reaches the microprocessor operating range (e.g. $V_{LOW} \geq 0.93 \cdot V_{out1(nom)}$). The $\overline{RESET}$ output has an open-collector and may be connected in a "wired-OR" configuration.

The watchdog circuit consists of a retriggerable monostable with a negative edge sensitive control input WDI. The watchdog feature may be disabled by means of the watchdog select input $\overline{WDS}$ driven to a "1". Figure 17 displays the typical $\overline{RESET}$ timing diagram.

The commuted current source I_{C5} on Pin 17, threshold voltage $V_{C5(L)}$, $V_{C5(H)}$ and an external capacitor C5 define the $\overline{RESET}$ delay and the watchdog timing. The relationship of the timing signals are indicated by the equations (9) to (11).

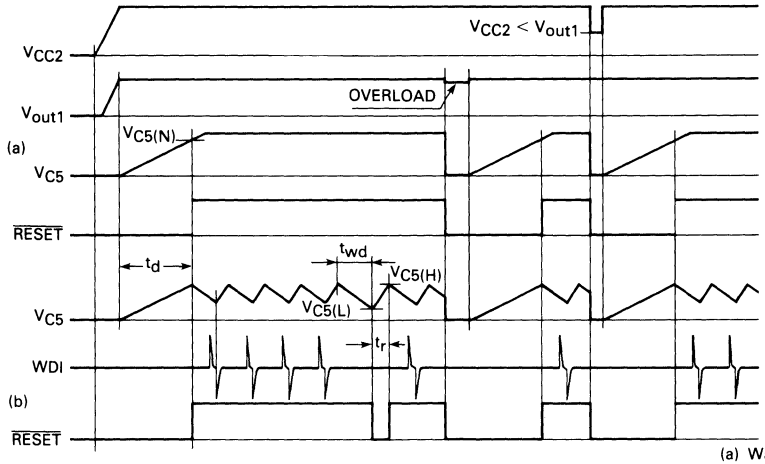
$\overline{RESET}$ delay:
$$t_d = \frac{C5 \cdot V_{C5(H)}}{|I_{C5}|} \quad (9)$$

Watchdog time-out:
$$t_{wd} = \frac{C5 \cdot (V_{C5(H)} - V_{C5(L)})}{5 \cdot I_{C5}} \quad (10)$$

Watchdog $\overline{RESET}$:
$$t_r = \frac{C5 \cdot (V_{C5(H)} - V_{C5(L)})}{50 \cdot |I_{C5}|} \quad (11)$$

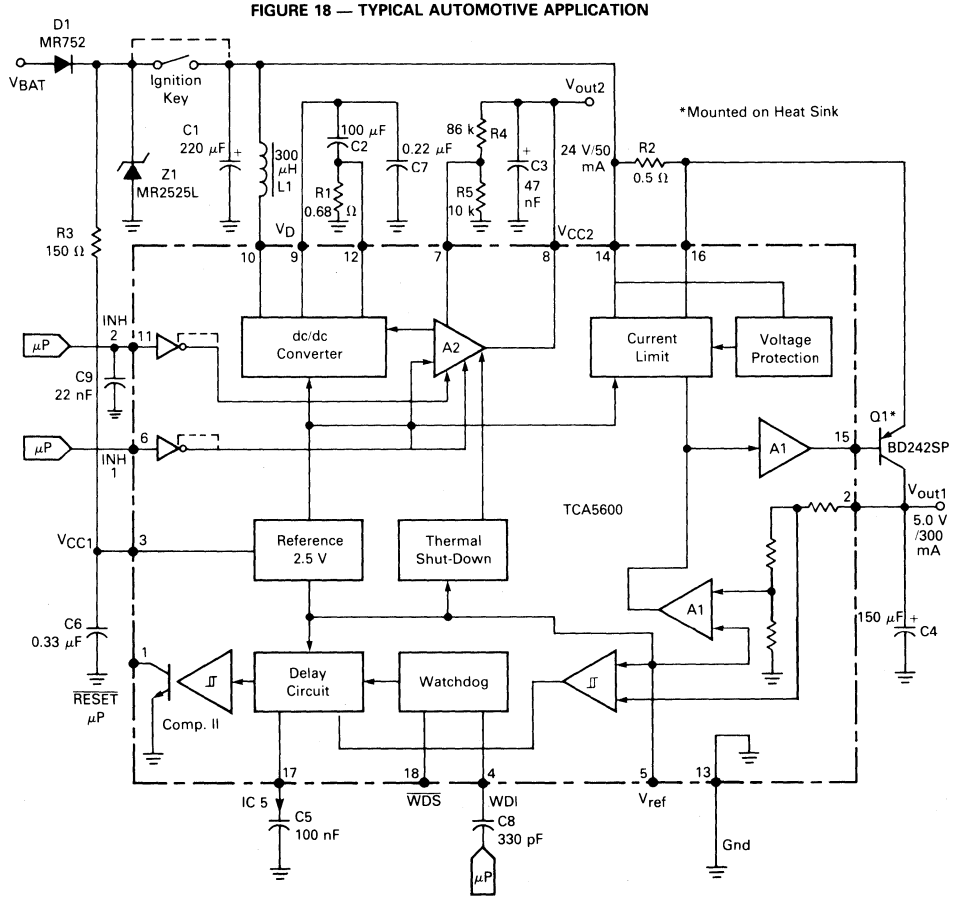
(I_{C5}, $V_{C5(H)}$, $V_{C5(L)}$: see electrical characteristics.)

FIGURE 17 — TYPICAL $\overline{\text{RESET}}$ TIMING DIAGRAM
(not to scale)



(a) Watchdog inhibited, $\overline{\text{WDS}}$ = "1"
(b) Watchdog operational, $\overline{\text{WDS}}$ = "0"

FIGURE 18 — TYPICAL AUTOMOTIVE APPLICATION



Mechanical Data

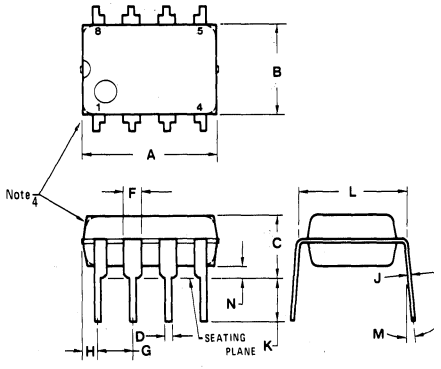
4

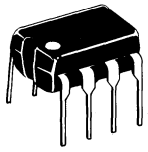
MECHANICAL DATA

The package availability for each device is indicated on the front page of the individual data sheets. Dimensions for the packages are given in this chapter.

8-PIN PACKAGE

PLASTIC PACKAGE
CASE 626-04



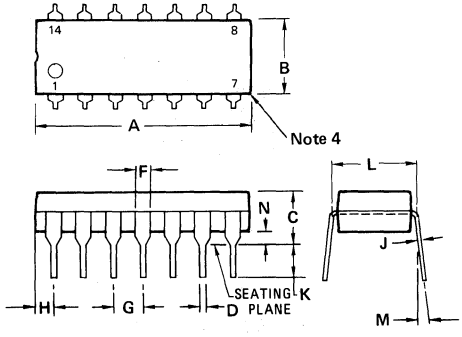


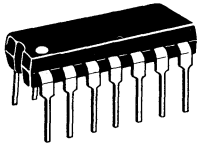
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	9.40	10.16	0.370	0.400
B	6.10	6.60	0.240	0.260
C	3.94	4.45	0.155	0.175
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	0.76	1.27	0.030	0.050
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	–	10^0	–	10^0
N	0.51	0.76	0.020	0.030

NOTES:
1. LEAD POSITIONAL TOLERANCE: ⌖ Ø 0.13 (0.005) Ⓜ | T | A Ⓜ | B Ⓜ
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CORNERS).
4. DIMENSIONS A AND B ARE DATUMS.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

14-PIN PACKAGE

PLASTIC PACKAGE
CASE 646-05



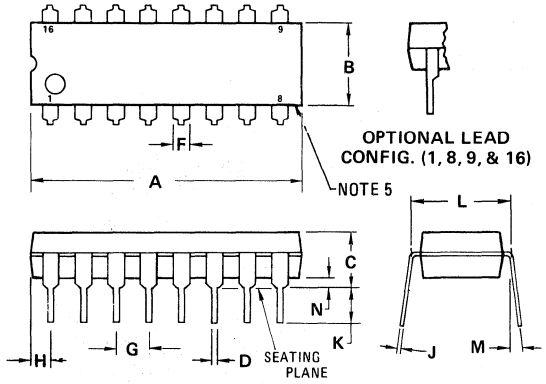


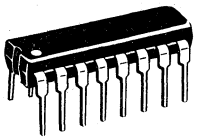
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	18.16	19.56	0.715	0.770
B	6.10	6.60	0.240	0.260
C	4.06	5.08	0.160	0.200
D	0.38	0.53	0.015	0.021
F	1.02	1.78	0.040	0.070
G	2.54 BSC		0.100 BSC	
H	1.32	2.41	0.052	0.095
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0^0	10^0	0^0	10^0
N	0.51	1.02	0.020	0.040

NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.
4. ROUNDED CORNERS OPTIONAL.

16-PIN PACKAGE

PLASTIC PACKAGE
CASE 648-05

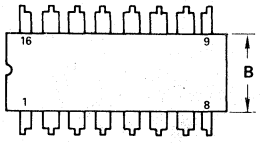


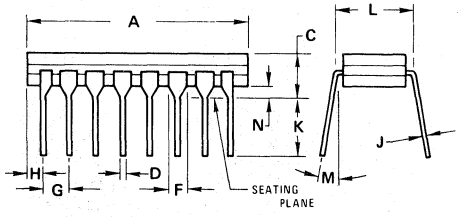


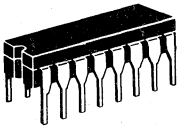
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	18.80	21.34	0.740	0.840
B	6.10	6.60	0.240	0.260
C	4.06	5.08	0.160	0.200
D	0.38	0.53	0.015	0.021
F	1.02	1.78	0.040	0.070
G	2.54 BSC		0.100 BSC	
H	0.38	2.41	0.015	0.095
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	10°	0°	10°
N	0.51	1.02	0.020	0.040

NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.
4. "F" DIMENSION IS FOR FULL LEADS. "HALF" LEADS ARE OPTIONAL AT LEAD POSITIONS 1, 8, 9, and 16).
5. ROUNDED CORNERS OPTIONAL.

CERAMIC PACKAGE
CASE 620-02





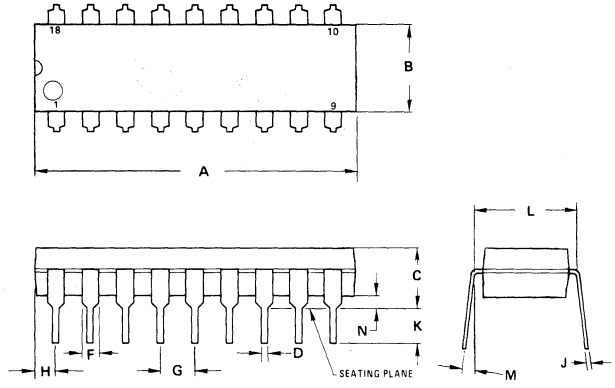


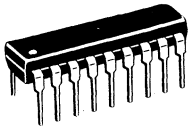
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	19.05	19.81	0.750	0.780
B	6.22	6.98	0.245	0.275
C	4.06	5.08	0.160	0.200
D	0.38	0.51	0.015	0.020
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.14	0.020	0.045
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.37	7.87	0.290	0.310
M	–	15°	–	15°
N	0.51	1.02	0.020	0.040

NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

18-PIN PACKAGE

PLASTIC PACKAGE
CASE 707-02





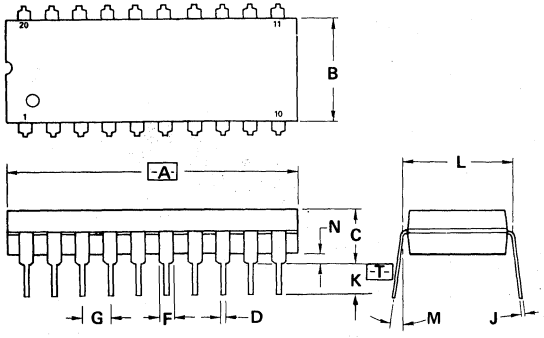
NOTES:

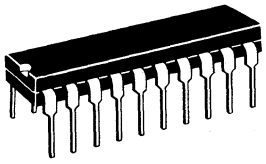
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	22.22	23.24	0.875	0.915
B	6.10	6.60	0.240	0.260
C	3.56	4.57	0.140	0.180
D	0.36	0.56	0.014	0.022
F	1.27	1.78	0.050	0.070
G	2.54 BSC		0.100 BSC	
H	1.02	1.52	0.040	0.060
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

20-PIN PACKAGE

PLASTIC PACKAGE
CASE 738-02

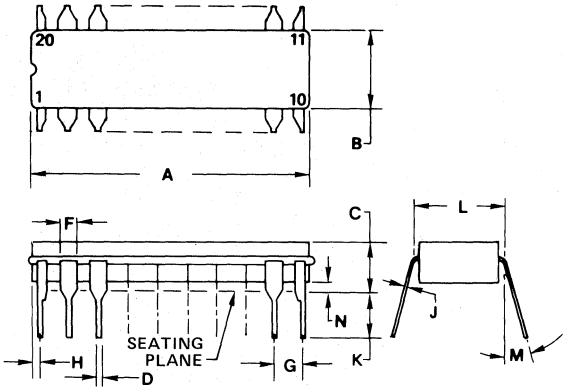


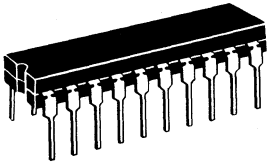


NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS;
   ⌖ Ø 0.25 (0.010) Ⓜ T A Ⓜ
3. -T- IS SEATING PLANE.
4. DIM "B" DOES NOT INCLUDE MOLD FLASH.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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

CERAMIC PACKAGE
CASE 732-03



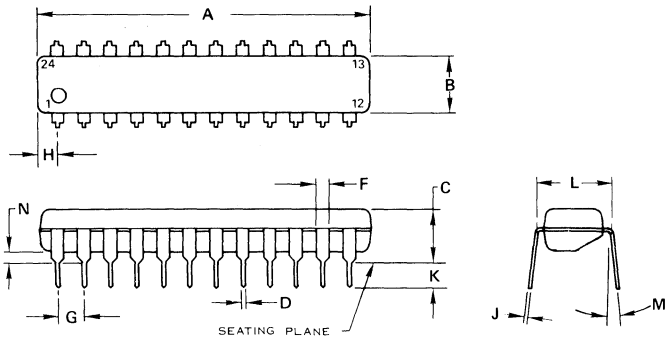


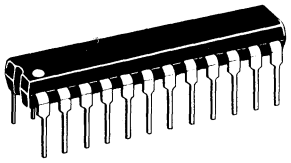
NOTES:
1. LEADS WITHIN 0.25 mm (0.010) DIA, TRUE POSITION AT SEATING PLANE, AT MAXIMUM MATERIAL CONDITION.
2. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIM A AND B INCLUDES MENISCUS.

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	23.88	25.15	0.940	0.990
B	6.60	7.49	0.260	0.295
C	3.81	5.08	0.150	0.200
D	0.38	0.56	0.015	0.022
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.27	0.020	0.050
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.25	1.02	0.010	0.040

24-PIN PACKAGE

PLASTIC PACKAGE
CASE 724-02

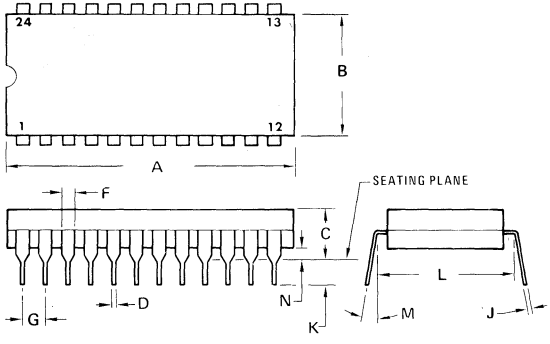


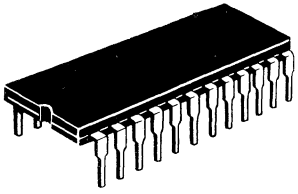


NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION (DIM D).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.13	1.230	1.265
B	6.35	6.86	0.250	0.270
C	4.06	4.57	0.160	0.180
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.60	2.11	0.063	0.083
J	0.18	0.30	0.007	0.012
K	2.92	3.43	0.115	0.135
L	7.37	7.87	0.290	0.310
M	–	10^0	–	10^0
N	0.51	1.02	0.020	0.040

CERAMIC PACKAGE
CASE 623-05



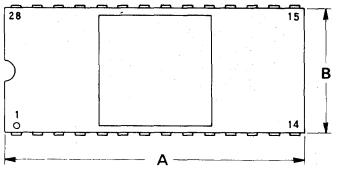


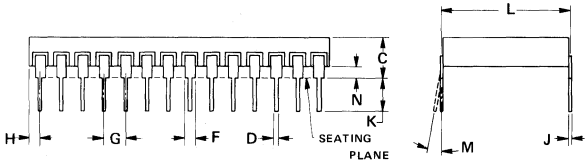
NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL).

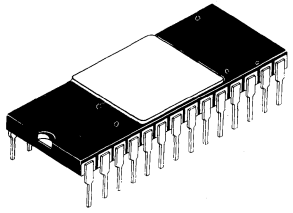
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.77	1.230	1.290
B	12.70	15.49	0.500	0.610
C	4.06	5.59	0.160	0.220
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.27	0.020	0.050

28-PIN PACKAGES

CERAMIC PACKAGE
CASE 719-03



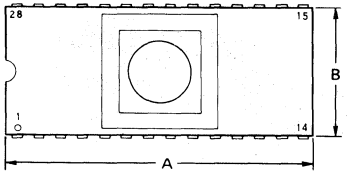


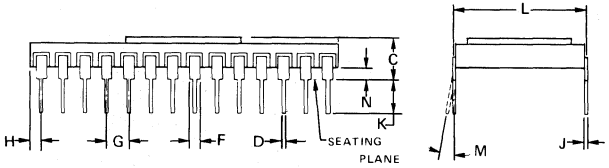


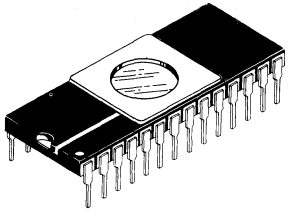
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	35.20	35.92	1.386	1.414
B	14.73	15.34	0.580	0.604
C	3.05	4.19	0.120	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.19	0.100	0.165
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	0.51	1.52	0.020	0.060

NOTES:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIAMETER (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

CERAMIC PACKAGE
CASE 719-04





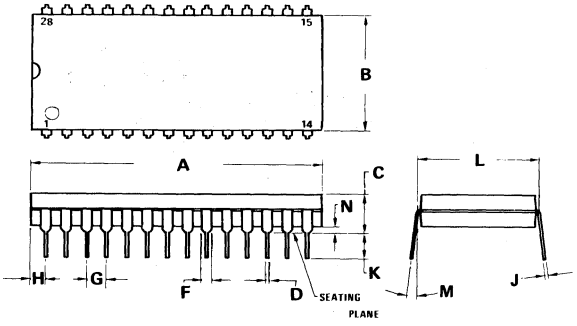


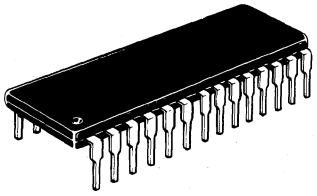
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	35.20	35.92	1.386	1.414
B	14.73	15.34	0.580	0.604
C	3.18	5.08	0.125	0.200
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.57	0.100	0.180
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	0.51	1.52	0.020	0.060

NOTES:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIAMETER (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

28-PIN PACKAGES (Continued)

PLASTIC PACKAGE
CASE 710-02

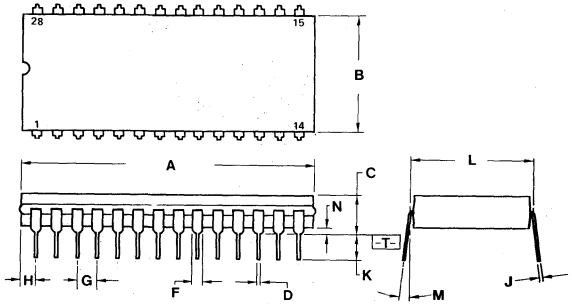


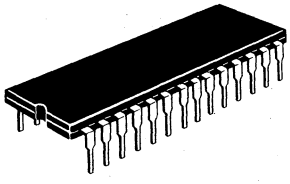


DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

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.

CERPID PACKAGE
CASE 733-03



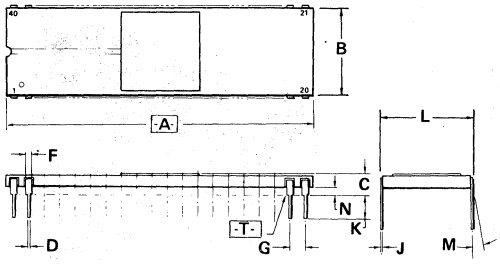


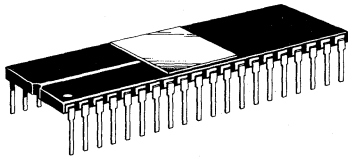
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.85	1.435	1.490
B	12.70	15.37	0.500	0.605
C	4.06	5.84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS:
   ⌖ | Ø 0.25 (0.010) Ⓜ | T | A Ⓜ
3. -T- IS SEATING PLANE.
4. DIM A AND B INCLUDES MENISCUS.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

40-PIN PACKAGES

CERAMIC PACKAGE
CASE 715-05





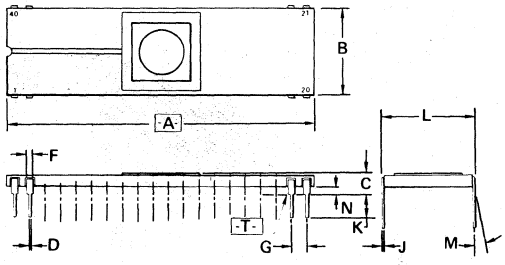
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	50.29	51.31	1.980	2.020
B	14.63	15.49	0.576	0.610
C	2.79	4.32	0.110	0.170
D	0.38	0.53	0.015	0.021
F	0.76	1.52	0.030	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.33	0.008	0.013
K	2.54	4.57	0.100	0.180
L	14.99	15.65	0.590	0.616
M	–	10°	–	10°
N	1.02	1.52	0.040	0.060

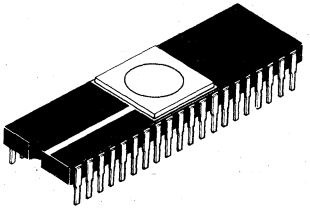
NOTES:
1. DIMENSION -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:

   ⌖ 0.25 (0.010) Ⓜ T A Ⓜ

3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

CERAMIC PACKAGE
CASE 715-06





DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	50.29	51.31	1.980	2.020
B	14.63	15.49	0.576	0.610
C	3.18	5.08	0.125	0.200
D	0.38	0.53	0.015	0.021
F	0.76	1.52	0.030	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.33	0.008	0.013
K	2.54	4.57	0.100	0.180
L	14.99	15.65	0.590	0.616
M	–	10°	–	10°
N	1.02	1.52	0.040	0.060

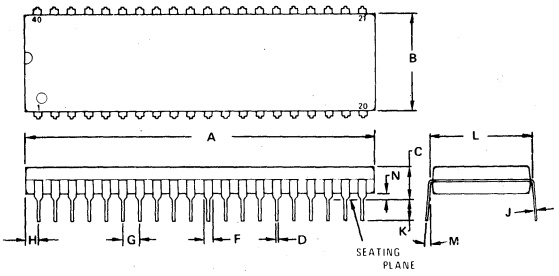
NOTES:
1. DIMENSION -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:

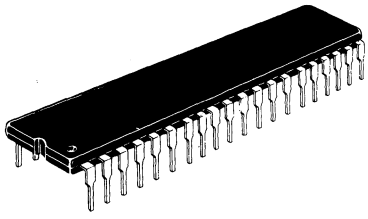
   ⌖ 0.25 (0.010) Ⓜ T A Ⓜ

3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

40-PIN PACKAGES (Continued)

PLASTIC PACKAGE
CASE 711-03

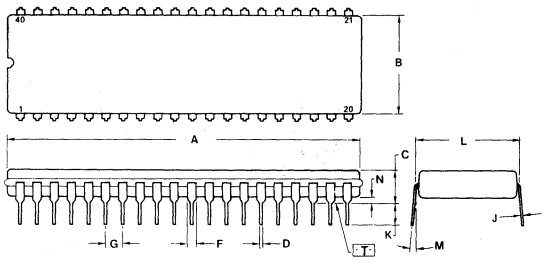


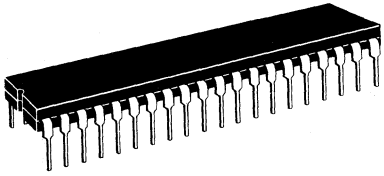


DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0^0	15^0	0^0	15^0
N	0.51	1.02	0.020	0.040

NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

CERDIP PACKAGE
CASE 734-04



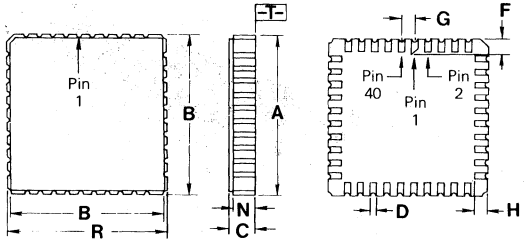


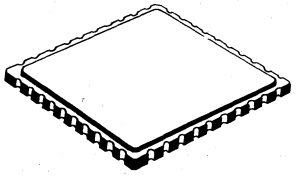
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	51.31	53.24	2.020	2.096
B	12.70	15.49	0.500	0.610
C	4.06	5.84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5^0	15^0	5^0	15^0
N	0.51	1.27	0.020	0.050

NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:
   ⌖ | Ø 0.25(0.010) Ⓜ | T | A Ⓜ
3. -T- IS SEATING PLANE.
4. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONS A AND B INCLUDE MENISCUS.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

40-PIN PACKAGES (Continued)

CHIP CARRIER
CASE 761-01





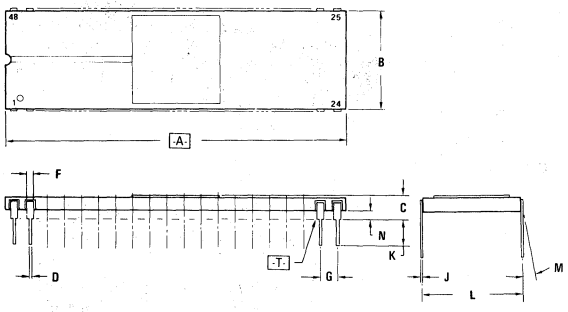
DIM	MILLIMETERS MIN	MAX	INCHES MIN	MAX
A	11.94	12.57	0.470	0.495
B	11.05	11.30	0.435	0.445
C	1.60	2.08	0.063	0.082
D	0.33	0.69	0.013	0.027
F	1.07	1.47	0.042	0.058
G	1.02 BSC		0.040 BSC	
H	0.84	1.19	0.033	0.047
N	1.27	1.79	0.050	0.070
R	11.94	12.57	0.470	0.495

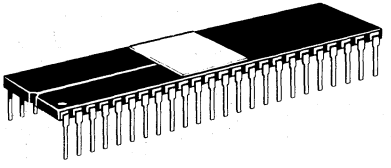
NOTES:
1. DIMENSIONS A & R ARE DATUMS.
2. -T- IS GAUGE PLANE.
3. POSITIONAL TOLERANCE FOR TERMINALS (D): 40 PLACES:
   ⌖ 0.25 (0.010) Ⓜ | T | A Ⓢ | R Ⓢ
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

4

48-PIN PACKAGES

CERAMIC PACKAGE
CASE 740-02





DIM	MILLIMETERS MIN	MAX	INCHES MIN	MAX
A	60.35	61.57	2.376	2.424
B	14.63	15.34	0.576	0.604
C	3.05	4.32	0.120	0.160
D	0.381	0.533	0.015	0.021
F	0.762	1.397	0.030	0.055
G	2.54 BSC		0.100 BSC	
J	0.203	0.330	0.008	0.013
K	2.54	4.19	0.100	0.165
L	14.99	15.65	0.590	0.616
M	0°	10°	0°	10°
N	1.016	1.524	0.040	0.060

NOTES:
1. DIMENSION -A- IS DATUM.
2. POSTIONAL TOLERANCE FOR LEADS:
   ⌖ Ø 0.25 (0.010) Ⓜ | T | A Ⓜ
3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

Technical Training

5

TECHNICAL TRAINING SYSTEM DESIGN

Since 1974 when Motorola first introduced the M6800 Family course around the United States, Motorola technical training courses have been among the most popular and effective methods for system designers to catch up or keep up with the microprocessor/microcomputer state-of-the-art.

Motorola technical training courses are scheduled throughout the world with courses in the United States, Canada, Mexico, Europe, and Asia. The schedule is advertised periodically, and information is always available from the training headquarters in Phoenix.

A special session of any Motorola technical training course may be held at your facility. This can be a standard course or a course designed to fulfill your particular needs.

The following is a list of course offerings. For more detailed course descriptions, course schedule in your area, or enrollment procedures, write: Motorola Technical Training, P.O. Box 2953, Mail Drop HW-68, Phoenix, Az. 85062. Or Call 602-244-7126, 602-962-2345, or 602-244-4945.

COURSE OFFERINGS

Basic M6800 Family Course — 4 Days (MTT1)

MTT1 is the original course of the M6800 Family, kept up to date and improved during the several years of its existence. It's designed to bring you up to speed in just four days, covering the background you'll need to design, develop, and debug an MC6800-based microcomputer system.

Basic M6801 Course — 4 Days (MTT2)

MTT2 is a beginning course on microprocessors based on the powerful MC6801 hardware and software. It is very similar to Course MTT1, but focuses on the MC6801 rather than the MC6800.

MC6809 Update — 2 Days (MTT3)

Course MTT3 is designed for the student who is very knowledgeable about the MC6800 microprocessor and wants to be equally capable with the MC6809.

High-Level Software — 4 Days (MTT4)

This high-level software course generates a working knowledge of the resident software packages available to users of EXORciser-based MDOS systems.

MC6801 Update — 2 Days (MTT5)

Course MTT5 is designed for the student who is very knowledgeable about the MC6800 microprocessor and wants to be equally capable with the MC6801.

M6805 Introductory Course — 3 Days (MTT6)

MTT6 is an introductory course on Motorola's M6805/M146805 Family of one-chip microcomputers/controllers.

Understanding Microprocessor Basics — 1 Day (MTT7)

This course is a one-day non-technical course designed to acquaint managers, secretaries, buyers, salesmen, and other non-designers with microprocessors. We cover the whys, whats, and hows of microcomputer systems. We'll give you the buzz words and use simplified examples to explain basic concepts. It's a good non-technical course. If you understand terms such as data bus, interrupt, multiplexing, mnemonics, etc., then this course isn't for you.

MC68000 16-Bit Microprocessor — 4 Days (MTT8)

The general features of the MC68000 such as pin functions, registers, addressing modes, and instruction set are covered. In addition, the unique features such as primitive instructions for high-level software, exception handling, and position independent machine code generation are discussed. The development tools used in the course include the Assembler, Editor, and the MC68000 ECB module. Two labs help provide experience with the hardware.

Designing With Micromodules — 2 Days (MTT9)

This 2-day course is designed to develop an understanding of the board-level computer system design approach for potential Micromodule users. The theme of the course is "learning the use of Micromodules through examples."

8-Bit Development Systems — 2 Days (MTT10)

This course is designed to prepare the student to understand and use the basic functions of both MC6800 EXORciser and MC6809 EXORciser II systems.

5

Basic MC6809 Course — 4 Days (MTT11)

MTT11 is a beginning course on microprocessors based on the powerful MC6809 hardware and software. It is very similar to Course MTT1, but focuses on the MC6809 rather than the MC6800.

Pascal — 4 Days (MTT12)

This course is designed to enable even the novice programmer to write well-constructed programs in Pascal. The first three days are for illustration of standard Pascal and structured programming as taught in a college-level course. The fourth day includes Motorola extensions and implementation for the MC6809 and MC68000. Each student has the opportunity to complete and execute several programs.

EXORmacs — 2 Days (MTT13)

This course aids the student in becoming familiar with EXORmacs. Included are the use of Utilities, Assemblers, Editors/Debuggers, and how to use Pascal on EXORmacs.

MPL — 4 Days (MTT14)

This course is designed to teach the student how to use the MPL Compiler for programming his or her applications. Upon completion of the course, the student will understand the (MC6800 or MC6809) MPL Compiler, the Macro Assembler, the Linking Loader, and MDOS, and will have written and executed programs which use these products.

EXORmacs Operating Systems — 4 Days (MTT15)

This course familiarizes the student with the multi-layered structure and operation of the EXORmacs operating system software. Use of RMS68K and VERSAdos on a target system is also discussed.

Virtual System Course — 4 Days (MTT16)

This course familiarizes the student with the MC68010 and various MC68000 peripheral chips. A vertical system example and design techniques used to implement it are presented.

Basic Macro-Cell Array & CAD Course — 3 Days (MTT17)

MTT17 is an introduction to designing with macro-cell arrays. Basic concepts and trade-offs between current technologies are discussed.

MCA-I CAD Course — 3 Days (MTT17B)

To familiarize the student with the Motorola Computer-Aided Design System used in designing ECL Macrocell arrays. Basic concepts and customer interface are discussed.

MCA-II CAD Course — 3 Days (MTT17C)

To familiarize the student with the Motorola Computer-Aided Design System used in designing CMOS Macrocell arrays. Basic concepts and customer interface are discussed.

MC68000 Operating System (UNIX*-like) — 4 Days (MTT18)

This course teaches the student how to use the Motorola UNIX*-like operating system and the C compiler.

Designing with VERSAmodules/VMEmodules — 4 Days (MTT19)

This course teaches the student about designing with board level products based around the VERSAbus and the VMEbus.

*UNIX is a trademark of Bell Labs

Memory Products

6

MOTOROLA

Memory Selector Guide

Motorola has developed a very broad range of reliable MOS and bipolar memories for virtually any digital data processing system application. And for those whose requirements go beyond individual components, Motorola also supplies Memory Systems and Micromodules.

New Motorola memories are being introduced continually. **This selector guide lists all those available as of November 1983.** For later releases, additional technical information or pricing, contact your nearest authorized Motorola distributor or Motorola sales office.

RAMs

MOS DYNAMIC RAMs

Organization	Part Number	Access Time (ns Max)	Power Supplies	No. of Pins
16384 × 1	MCM4116BP15	150	+12, ±5 V	16
16384 × 1	MCM4116BP20	200	+12, ±5 V	16
16384 × 1	MCM4116BP25	250	+12, ±5 V	16
16384 × 1	MCM4517P10	100	+5 V	16
16384 × 1	MCM4517P12	120	+5 V	16
16384 × 1	MCM4517P15	150	+5 V	16
16384 × 1	MCM4517P20	200	+5 V	16
65536 × 1	MCM6664AP15[1]	150	+5 V	16
65536 × 1	MCM6664AP20[1]	200	+5 V	16
65536 × 1	MCM6665AP15	150	+5 V	16
65536 × 1	MCM6665AP20	200	+5 V	16
65536 × 1	MCM6664BP15[1]*	150	+5 V	16
65536 × 1	MCM6664BP20[1]*	200	+5 V	16
65536 × 1	MCM6665BP15*	150	+5 V	16
65536 × 1	MCM6665BP20*	200	+5 V	16

CMOS STATIC RAMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
2048 × 8	MCM6116P12	120	24
2048 × 8	MCM6116P15	150	24
2048 × 8	MCM6116P20	200	24
4096 × 1	MCM6147P55	55	18
4096 × 1	MCM6147P70	70	18

Operating temperature ranges: 0°C to 70°C

* To be introduced.
(Not all speed selections shown)

[1] Motorola's innovative pin #1 refresh

[2] 300 mil package

MOS STATIC RAMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
128 × 8	MCM6810	450	24
128 × 8	MCM68A10	360	24
128 × 8	MCM68B10	250	24
1024 × 4	MCM2114P20	200	18
1024 × 4	MCM2114P25	250	18
1024 × 4	MCM2114P30	300	18
1024 × 4	MCM2114P45	450	18
1024 × 4	MCM21L14P20	200	18
1024 × 4	MCM21L14P25	250	18
1024 × 4	MCM21L14P30	300	18
1024 × 4	MCM21L14P45	450	18
2048 × 8	MCM2016HP45	45	24
2048 × 8	MCM2016HN45	45	24[2]
2048 × 8	MCM2016HY45	45	24[2]
2048 × 8	MCM2016HP55	55	24
2048 × 8	MCM2016HN55	55	24[2]
2048 × 8	MCM2016HY55	55	24[2]
2048 × 8	MCM2016HP70	70	24
2048 × 8	MCM2016HN70	70	24[2]
2048 × 8	MCM2016HY70	70	24[2]
16384 × 1	MCM2167HP35	35	20
16384 × 1	MCM2167HL35	35	20
16384 × 1	MCM2167HZ35	35	20
16384 × 1	MCM2167HP45	45	20
16384 × 1	MCM2167HL45	45	20
16384 × 1	MCM2167HZ45	45	20
16384 × 1	MCM2167HP70	70	20
16384 × 1	MCM2167HL70	70	20
16384 × 1	MCM2167HZ70	70	20

EPROMs

MOS EPROMs

Organization	Part Number	Access Time (ns max)	Power Supplies	No. of Pins
8192 × 8	MCM68764C	450	+5 V	24
8192 × 8	MCM68766C	450	+5 V	24
8192 × 8	MCM68766C35	350	+5 V	24

ROMs

MOS STATIC ROMs (+5 Volts)

Character Generators[3]

Organization	Part Number	Access Time (ns max)	No. of Pins
128 x (7 x 5)	MCM6670P	350	18
128 x (7 x 5)	MCM6674P	350	18
128 x (9 x 7)	MCM66700P	350	24
128 x (9 x 7)	MCM66710P	350	24
128 x (9 x 7)	MCM66714P	350	24
128 x (9 x 7)	MCM66720P	350	24
128 x (9 x 7)	MCM66730P	350	24
128 x (9 x 7)	MCM66734P	350	24
128 x (9 x 7)	MCM66740P	350	24
128 x (9 x 7)	MCM66750P	350	24
128 x (9 x 7)	MCM66760P	350	24
128 x (9 x 7)	MCM66770P	350	24
128 x (9 x 7)	MCM66780P	350	24
128 x (9 x 7)	MCM66790P	350	24

MOS Binary ROMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
2048 x 8	MCM68A316EP	350	24
2048 x 8	MCM68A316EP91[4]	350	24
4096 x 8	MCM68A332P	350	24
4096 x 8	MCM68A332P2[4]	350	24
8192 x 8	MCM68364P35	350	24
8192 x 8	MCM68364P35-3[4]	350	24
8192 x 8	MCM68364P25	250	24
8192 x 8	MCM68364P20	200	24
8192 x 8	MCM68365P25	250	24
8192 x 8	MCM68365P35	350	24
8192 x 8	MCM68366P25	250	24
8192 x 8	MCM68366P35	350	24
16384 x 8	MCM63128P15	150	28
16384 x 8	MCM63128P20	200	28
32768 x 8	MCM63256P15	150	28
32768 x 8	MCM63256P20	200	28

CMOS ROMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
256 x 4	MCM14524	1200	16
2048 x 8	MCM65516P43	430	18
2048 x 8	MCM65516P43M[8]	430	18
2048 x 8	MCM65516P55	550	18

Operating temperature ranges: 0°C to 70°C

*To be introduced.
(Not all speed selections shown)

[3]Character generators include shifted and unshifted characters, ASCII alphanumeric control, math, Japanese British, German, European and French symbols.

[4]Standard Patterns for MOS ROMs:
MCM68A316EP91 — Universal Code Converter and Character Generator
MCM68A332P2 — Sine/Cosine Look-Up Table
MCM68364P35-3 — Log/Antilog Look-Up Table
MCM65516P43M — MC146805 Monitor Program

Logic and Special Function Products

7

MOTOROLA

MC144110/1

Product Preview

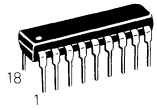
QUAD AND HEX D/A CONVERTERS

The MC144110 and MC144111 are hex and quad static D/A converters realized in CMOS technology. Each converter, featuring 6-bit resolution, consists of a 6-bit shift register, 6-bit latch, and a static D/A converter.

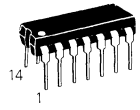
- 4/6 Direct R-2R Network Outputs
- 4/6 Emitter Follower Outputs
- MPU Compatible Input Levels
- Serial Data Input
- Data Cascade Output
- Wide Operating Voltage Range of 4.5 to 15 Vdc

CMOS
(LOW POWER COMPLEMENTARY MOS)

QUAD AND HEX
D/A CONVERTERS

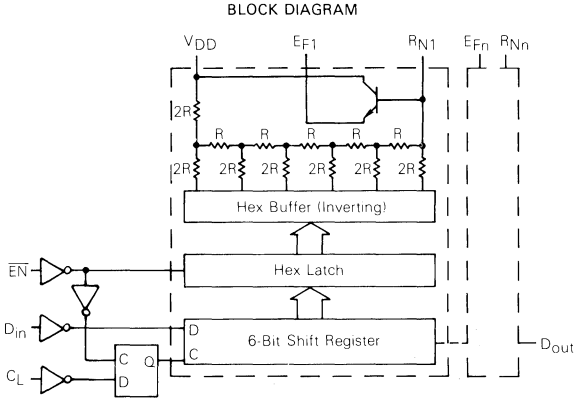


MC144110
P SUFFIX
PLASTIC PACKAGE
CASE 707-02



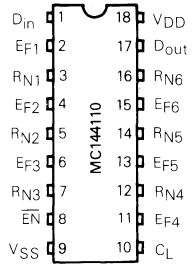
MC144111
P SUFFIX
PLASTIC PACKAGE
CASE 646

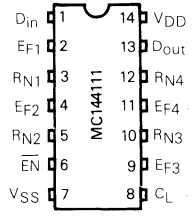
BLOCK DIAGRAM



Latch = Master Only
MSB = Shifted First
LSB = Shifted Last

PIN ASSIGNMENTS





7

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

MOTOROLA

MC145000
MC145001

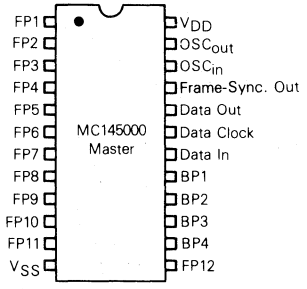
Advance Information

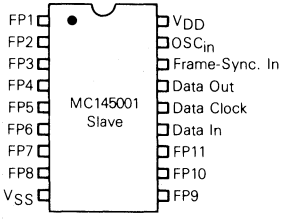
MULTIPLEXED LCD DRIVERS MASTER AND SLAVE

The MC145000 (Master) LCD Driver and the MC145001 (Slave) LCD Driver are CMOS devices designed to drive liquid crystal displays in a multiplexed-by-four configuration. The Master unit generates both frontplane and backplane waveforms, and is capable of independent operation. The Slave unit generates only frontplane waveforms, and is synchronized with the backplanes from the Master unit. Several Slave units may be cascaded from the Master unit to increase the number of LCD segments driven in the system. The maximum number of frontplanes is dependent upon the capacitive loading on the backplane drivers and the drive frequency. The devices use data from a microprocessor or other serial data and clock source to drive one LCD segment per bit.

- Microprocessor Compatibility
- Serial Data, Externally Clocked
- Multiplexing-By-Four
- Net dc Drive Component Less Than 50 mV
- Master Drives 48 LCD Segments
- Slave Provides Frontplane Drive for 44 LCD Segments
- Drives Segments Up to one Square Centimeter (0.155 Square Inches)
- Display Operating Frequency = 250 Hz Maximum
- Supply Voltage Range = 3 V to 6 V
- Latch Storage of Input Data
- Low Power Dissipation
- Logic Input Voltage Can Exceed V_{DD}
- Accomodates External Temperature Compensation
- 24-Pin DIP Configuration — Master
- 18-Pin DIP Configuration — Slave

PIN ASSIGNMENTS



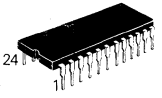


This is advance information and specifications are subject to change without notice.

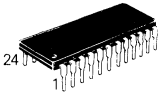
CMOS LSI

(LOW-POWER COMPLEMENTARY MOS)

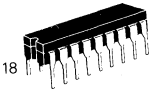
MULTIPLEXED LCD DRIVERS MASTER AND SLAVE



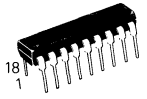
L SUFFIX
CERAMIC PACKAGE
CASE 623



P SUFFIX
PLASTIC PACKAGE
CASE 709



L SUFFIX
CERAMIC PACKAGE
CASE 726



P SUFFIX
PLASTIC PACKAGE
CASE 707

ORDERING INFORMATION

MC14XXXB

Suffix Denotes
L Ceramic Package
P Plastic Package

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the ranges $V_{SS} \leq V_{out} \leq V_{DD}$ and $V_{SS} \leq V_{in} \leq 15$ V

Unused inputs must always be tied to an appropriate logic voltage level.

7

MOTOROLA

MC145040
MC145041

Product Preview

ANALOG-TO-DIGITAL CONVERTER WITH SERIAL INTERFACE

The MC145040 and MC145041 are low-cost 8-bit A/D Converters with serial interface ports to provide communication with microprocessors and microcomputers. The converters operate from a single power supply with a guaranteed linearity of ± ½ LSB over the full temperature range. No external trimming is required.

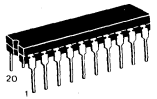
The MC145040 requires an external clock signal (A/D CLK) to operate the dynamic A/D conversion sequence. The MC145041 has an internal clock and an end-of-conversion signal (EOC) is provided.

- Conversion time ≤ 32 μs
- 11 Analog Input Channels with Internal Sample and Hold
- 0- to 5-Volt Analog Input Range with Single 5-Volt Supply
- Successive Approximation A/D Technique Uses All Capactive DAC Structure
- Ratiometric Conversion
- Separate V_{ref} and V_{AG} Pins for Noise Immunity
- V_{ref} and V_{AG} Adjustable for Reduced Input Range
- No External Trimming Required
- Direct Interface to Motorola SPI and National MICROWIRE™ Serial Data Ports
- TTL-Compatible Inputs May Be Driven With CMOS
- CMOS or TTL Compatible Outputs
- Very Low Reference Current Requirement (10 μA)
- Low Power Consumption (12 mW)
- Internal Test Mode for Self Test

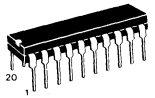
CMOS LSI

LOW-POWER COMPLEMENTARY MOS
SILICON GATE

ANALOG-TO-DIGITAL CONVERTER WITH SERIAL INTERFACE



L SUFFIX
CERAMIC PACKAGE
CASE 732-03



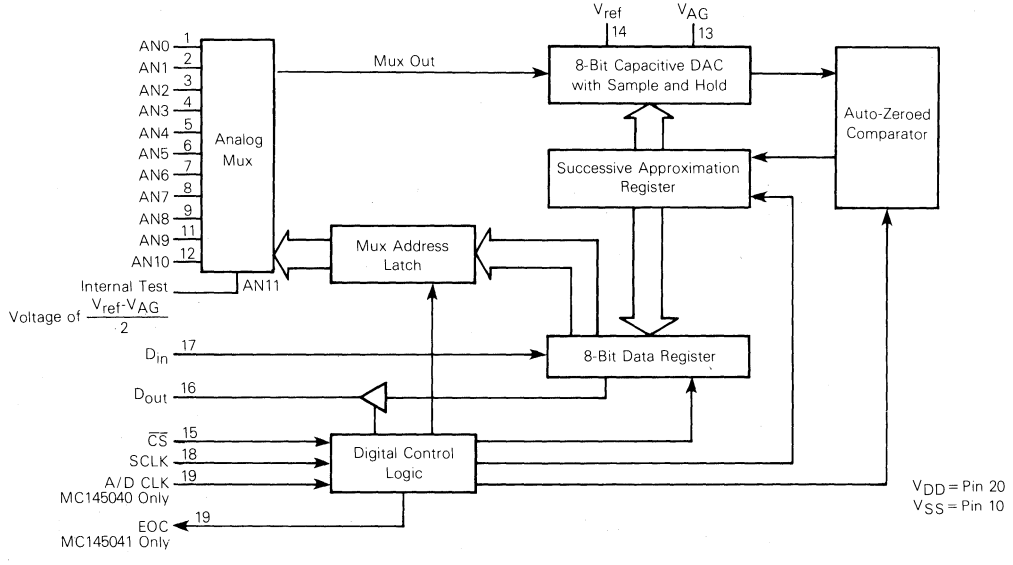
P SUFFIX
PLASTIC PACKAGE
CASE 738-02

ORDERING INFORMATION

MC14XXXX

Suffix	Denotes
L	Ceramic DIP
P	Plastic DIP
FN	Plastic Quad Package

BLOCK DIAGRAM



MICROWIRE™ is a trademark of National Semiconductor

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

7

MOTOROLA

MC145157
MC145158

Advance Information

CMOS LSI
(LOW-POWER COMPLEMENTARY MOS)

SERIAL INPUT PLL FREQUENCY SYNTHESIZER

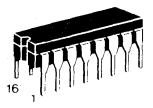
SERIAL INPUT PLL FREQUENCY SYNTHESIZERS

The MC145157 and MC145158 are part of a family of CMOS Phase Lock Loop frequency synthesizer devices from Motorola. These devices utilize silicon-gate CMOS technology to achieve the operating speeds necessary for high-frequency operation. The family includes devices having serial, parallel, and 4-bit data bus programmable inputs. Options include single- or dual-modulus capability, transmit/receive offsets, and a choice of phase detector types.

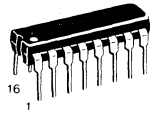
The MC145157 and MC145158 have fully programmable 14-bit reference counters, as well as fully programmable ÷ N (MC145157) and ÷ N/÷ A (MC145158) counters. The counters are programmed serially through a common data input and latched into the appropriate counter latch, according to the last data bit (control bit) entered.

When combined with a loop filter and VCO, these devices can provide all the remaining functions for a PLL frequency synthesizer operating up to the device's frequency limit. For higher VCO frequency operation, a down mixer or a fixed-divide prescaler can be used between the VCO and the PLL for the MC145157 and a dual-modulus prescaler for the MC145158.

- General Purpose Applications —
  CATV
  AM/FM
  Two-Way Radios
  TV Tuning
  Scanning Receivers
  Amateur Radio
- Low Power Drain
- 3.0. to 9.0 V Supply Range
- 30 MHz Typical Input Capability @ 5 V (f_{in} Input)
- Fully Programmable Reference and ÷ N Counters
- Reference Divider Range = 3 to 16383
- ÷ N Range = 3 to 16383 for the MC145157
- Dual Modulus Capability for the MC145158
- f_V and f_r Outputs
- Lock Detect Signal
- "Linearized" Digital Phase Detector
- Single Ended (Three-State) or Double-Ended Phase Detector Outputs
- ÷ N Range = 0 to 1023 for the MC145158
- ÷ A Range = 0 to 127 for the MC145158



L SUFFIX
CERAMIC PACKAGE
CASE 620



P SUFFIX
PLASTIC PACKAGE
CASE 648

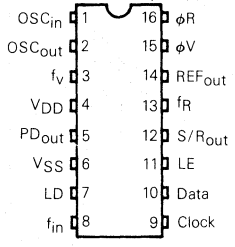
ORDERING INFORMATION

MC14XXXX

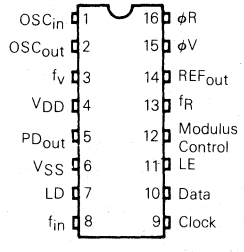
Suffix	Denotes
L	Ceramic Package
P	Plastic Package
C	Limited Operating Temperature Range

PIN ASSIGNMENT

MC145157



MC145158



7

This is advance information and specifications are subject to change without notice.

BUFFERS/INVERTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC04	Hex Inverter	LS04	*4069	LS/CMOS	14
HCT04	Hex Inverter with LSTTL-Compatible Inputs	LS04	*4069	LS/CMOS	14
HCU04	Hex Unbuffered Inverter	*LS04	4069	LS/CMOS	14
HC14	Hex Schmitt-Trigger Inverter	LS14	4584	LS/CMOS	14
HC125	Quad 3-State Noninverting Buffer	LS125		LS	14
HC126	Quad 3-State Noninverting Buffer	LS126		LS	14
HC240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver	LS240		LS	20
HCT240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS240		LS	20
HC241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS241		LS	20
HCT241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS241		LS	20
HC242	Quad 3-State Inverting Bus Transceiver	LS242		LS	14
HC243	Quad 3-State Noninverting Bus Transceiver	LS243		LS	14
HC244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS244		LS	20
HCT244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS244		LS	20
HC245	Octal 3-State Noninverting Bus Transceiver	LS245		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HC365	Hex 3-State Noninverting Buffer with Common Enables	LS365A		LS	16
HC366	Hex 3-State Inverting Buffer with Common Enables	LS366A		LS	16
HC367	Hex 3-State Noninverting Buffer with Separate 2-Bit and 4-Bit Sections	LS367A	*4503	LS/CMOS	16
HC368	Hex 3-State Inverting Buffer with Separate 2-Bit and 4-Bit Sections	LS368A		LS	16
HC540	Octal 3-State Inverting Buffer/Line Driver/Line Receiver	LS540		LS	20
HC541	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS541		LS	20
HC640	Octal 3-State Inverting Bus Transceiver	LS640		LS	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HC643	Octal 3-State Inverting and Noninverting Bus Transceiver	LS643		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HC4049	Hex Inverting Buffer/Logic-Level Down Converter		4049	CMOS	16
HC4050	Hex Noninverting Buffer/Logic-Level Down Converter		4050	CMOS	16

*Suggested alternative

BUFFERS

Device	HC 04	HCT 04	HCU 04	HC 14	HC 125	HC 126	HC 240	HCT 240	HC 241	HCT 241	HC 242	HC 243	HC 244	HCT 244
# Pins	14	14	14	14	14	14	20	20	20	20	14	14	20	20
Quad Device					•	•					•	•		
Hex Device	•	•	•	•										
Octal Device							•	•	•	•			•	•
Noninverting Outputs					•	•			•	•		•	•	•
Inverting Outputs	•	•	•	•			•	•			•			
Single Stage (unbuffered)			•											
Schmitt Trigger				•										
3-State Outputs					•	•	•	•	•	•	•	•	•	•
Common Output Enables								•	•	•	•	•	•	•
Active-Low Output Enables					•		••	••	•	•	•	•	••	••
Active-High Output Enables						•			•	•	•	•		
Separate 4-Bit Sections							•	•	•	•			•	•
Separate 2-Bit and 4-Bit Sections														
Transceiver											•	•		
Direction Control														
Logic-Level Down Converter														
LSTTL-Compatible Inputs		•						•		•				•

Device	HC 245	HCT 245	HC 365	HC 366	HC 367	HC 368	HC 540	HC 541	HC 640	HCT 640	HC 643	HCT 643	HC 4049	HC 4050
# Pins	20	20	16	16	16	16	20	20	20	20	20	20	16	16
Quad Device														
Hex Device			•	•	•	•							•	•
Octal Device	•	•					•	•	•	•	•	•		
Noninverting Outputs	•	•	•		•			•			•	•		•
Inverting Outputs				•		•	•		•	•	•	•	•	
Single Stage (unbuffered)														
Schmitt Trigger														
3-State Outputs	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Common Output Enables	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Active-Low Output Enables	•	•	••	••	••	••	••	••	•	•	•	•		
Active-High Output Enables														
Separate 4-Bit Sections														
Separate 2-Bit and 4-Bit Sections					•	•								
Transceiver	•	•							•	•	•	•		
Direction Control	•	•							•	•	•	•		
Logic-Level Down Converter													•	•
LSTTL-Compatible Inputs		•								•		•		

GATES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC00	Quad 2-Input NAND Gate	LS00	4011	LS	14
HCT00	Quad-2 Input NAND Gate with LSTTL-Compatible Inputs	LS00	4011	LS	14
HC02	Quad 2-Input NOR Gate	LS02	4001	LS	14
HC03	Quad 2-Input NAND Gate with Open-Drain Outputs	LS03	*4011	LS	14
HC08	Quad 2-Input AND Gate	LS08	4081	LS	14
HC10	Triple 3-Input NAND Gate	LS10	4023	LS	14
HC11	Triple 3-Input AND Gate	LS11	4073	LS	14
HC20	Dual 4-Input NAND Gate	LS20	4012	LS	14
HC27	Triple 3-Input NOR Gate	LS27	4025	LS	14
HC30	8-Input NAND Gate	LS30	4068	LS	14
HC32	Quad 2-Input OR Gate	LS32	4071	LS	14
HC51	2-Wide, 2-Input/2-Wide, 3-Input AND-OR-INVERT Gates	LS51	*4506	LS	14
☆HC58	2-Wide, 2-Input/2-Wide, 3-Input AND-OR Gates	*LS51	*4506		14
HC86	Quad 2-Input Exclusive OR Gate	LS86	4070	LS	14
HC132	Quad 2-Input Schmitt-Trigger NAND Gate	LS132	4093	LS	14
HC133	13-Input NAND Gate	LS133		LS	16
HC266	Quad 2-Input Exclusive NOR Gate	*LS266	4077	LS/CMOS	14
HC4002	Dual 4-Input NOR Gate	*LS25	4002	CMOS	14
HC4075	Triple 3-Input OR Gate		4075	CMOS	14
HC4078	8-Input NOR/OR Gate		4078	CMOS	14

* Suggested alternative

☆ High-Speed CMOS design only

Device	HC 00	HCT 00	HC 02	HC 03	HC 08	HC 10	HC 11	HC 20	HC 27	HC 30
#Pins	14	14	14	14	14	14	14	14	14	14
Single Device										•
Dual Device								•		
Triple Device						•	•		•	
Quad Device	•	•	•	•	•					
NAND	•	•		•		•		•		•
NOR			•						•	
AND					•		•			
OR										
Exclusive OR										
Exclusive NOR										
AND-OR-INVERT										
AND-OR										
2-Input	•	•	•	•	•					
3-Input						•	•		•	
4-Input								•		
8-Input										•
13-Input										
Schmitt Trigger Inputs										
LSTTL-Compatible Inputs			•							
Open-Drain Outputs				•						

Device	HC 32	HC 51	HC 58	HC 86	HC 132	HC 133	HC 266	HC 4002	HC 4075	HC 4078
#Pins	14	14	14	14	14	16	14	14	14	14
Single Device		•	•			•				•
Dual Device								•		
Triple Device									•	
Quad Device	•			•	•		•			
NAND					•	•				
NOR								•		•
AND										
OR	•								•	
Exclusive OR				•						
Exclusive NOR							•			
AND-OR-INVERT		•								
AND-OR			•							
2-Input	•	•	•	•	•		•			
3-Input		•	•						•	
4-Input								•		
8-Input										•
13-Input						•				
Schmitt Trigger Inputs					•					
LSTTL-Compatible Inputs										
Open-Drain Outputs										

SCHMITT TRIGGERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC14	Hex Schmitt-Trigger Inverter	LS14	4584	LS/CMOS	14
HC132	Quad 2-Input Schmitt-Trigger NAND Gate	LS132	4093	LS	14

BUS TRANSCEIVERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC242	Quad 3-State Inverting Bus Transceiver	LS242		LS	14
HC243	Quad 3-State Noninverting Bus Transceiver	LS243		LS	14
HC245	Octal 3-State Noninverting Bus Transceiver	LS245		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HC640	Octal 3-State Inverting Bus Transceiver	LS640		LS	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HC643	Octal 3-State Inverting and Noninverting Bus Transceiver	LS643		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HC646	Octal 3-State Noninverting Bus Transceiver and D-Type Flip-Flop	LS646		LS	24
HC648	Octal 3-State Inverting Bus Transceiver and D-Type Flip-Flop	LS648		LS	24

Device	HC 242	HC 243	HC 245	HCT 245	HC 640	HCT 640	HC 643	HCT 643	HC 646	HC 648
#Pins	14	14	20	20	20	20	20	20	24	24
Quad Device	•	•								
Octal Device			•	•	•	•	•	•	•	•
Buffer	•	•	•	•	•	•	•	•		
Storage Capability									•	•
Inverting Output	•				•	•	•	•	•	
Noninverting Output		•	•	•			•	•		•
Common Output Enables	•	•	•	•	•	•	•	•	•	•
Active-Low Output Enable	•	•	•	•	•	•	•	•	•	•
Active-High Output Enable	•	•								
Direction Control			•	•	•	•	•	•	•	•
LSTTL-Compatible Inputs				•		•		•		

7

LATCHES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC75	Dual 2-Bit Transparent Latch	LS75	*4042	LS	16
HC259	8-Bit Addressable Latch/1-of-8 Decoder	LS259	*4099	LS	16
HC373	Octal 3-State Noninverting D-Type Transparent Latch	LS373,LS573		LS373	20
HCT373	Octal 3-State Noninverting D-Type Transparent Latch with LSTTL-Compatible Inputs	LS373,LS573		LS373	20
HC533	Octal 3-State Inverting D-Type Transparent Latch	LS533		LS	20
HC563	Octal 3-State Inverting D-Type Transparent Latch	LS533			20
HC573	Octal 3-State Noninverting D-Type Transparent Latch	LS373,LS573		LS573	20

*Suggested alternative

Device	HC 75	HC 259	HC 373	HCT 373	HC 533	HC 563	HC 573
#Pins	16	16	20	20	20	20	20
Single Device		•					
Dual Device	•						
Octal Device			•	•	•	•	•
1-Bit			•	•	•	•	•
2-Bit	•						
8-Bit		•					
Transparent	•	•	•	•	•	•	•
Addressable		•					
Noninverting Outputs	•	•	•	•			•
Inverting Outputs	•				•	•	
Common Latch Enable			•	•	•	•	•
Active-Low Latch Enable	•	•	•	•	•	•	•
Active-Low Reset		•					
3-State Outputs			•	•	•	•	•
Common Output Enable; Active-Low			•	•	•	•	•
LSTTL-Compatible Inputs				•			

These devices are identical in function and are different in pinout only: HC373 and HC573
HC533 and HC563

7

FLIP-FLOPS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC73	Dual J-K Flip-Flop with Reset	LS73A, LS107A	*4027	LS73A	14
HC74	Dual D-Type Flip-Flop with Set and Reset	LS74A	4013	LS	14
HC76	Dual J-K Flip-Flop with Set and Reset	LS76A, LS112A	*4027	LS76A	16
HC107	Dual J-K Flip-Flop with Reset	LS73A, LS107A	*4027	LS107A	14
HC109	Dual J-$\overline{K}$ Flip-Flop with Set and Reset	LS109A	*4027	LS	16
HC112	Dual J-K Flip-Flop with Set and Reset	LS76A, LS112A	*4027	LS112A	16
HC113	Dual J-K Flip-Flop with Set	LS113A	*4027	LS	14
HC173	Quad 3-State D-Type Flip-Flop with Common Clock and Reset	LS173A	4076	LS/CMOS	16
HC174	Hex D-Type Flip-Flop with Common Clock and Reset	LS174	4174	LS/CMOS	16
HC175	Quad D-Type Flip-Flop with Common Clock and Reset	LS175	4175	LS/CMOS	16
HC273	Octal D-Type Flip-Flop with Common Clock and Reset	LS273		LS	20
HC374	Octal 3-State Noninverting D-Type Flip-Flop	LS374, LS574		LS374	20
HCT374	Octal 3-State Noninverting D-Type Flip-Flop with LSTTL-Compatible Inputs	LS374, LS574		LS374	20
HC534	Octal 3-State Inverting D-Type Flip-Flop	LS534		LS	20
HC564	Octal 3-State Inverting D-Type Flip-Flop	LS534			20
HC574	Octal 3-State Noninverting D-Type Flip-Flop	LS374, LS574		LS574	20
HC646	Octal 3-State Noninverting Bus Transceiver and D-Type Flip-Flop	LS646		LS	24
HC648	Octal 3-State Inverting Bus Transceiver and D-Type Flip-Flop	LS648		LS	24

*Suggested alternative

Device	HC 73	HC 74	HC 76	HC 107	HC 109	HC 112	HC 113	HC 173	HC 174
#Pins	14	14	16	14	16	16	14	16	16
Type	J-K	D	J-K	J-K	J-$\overline{K}$	J-K	J-K	D	D
Dual Device	•	•	•	•	•	•	•		
Quad Device								•	
Hex Device									•
Octal Device									
Common Clock								•	•
Negative-Transition Clocking	•		•	•		•	•		
Postive-Transition Clocking		•			•			•	•
Common, Active-Low Data Enables								••	
Noninverting Outputs	•	•	•	•	•	•	•	•	•
Inverting Outputs	•	•	•	•	•	•	•		
3-State Outputs								•	
Common, Active-Low Output Enables								••	
Common Reset								•	•
Active-Low Reset	•	•	•	•	•	•			•
Active-High Reset								•	
Active-Low Set		•	•		•	•	•		
Transceiver									
Direction Control									
LSTTL-Compatible Inputs									

Device	HC 175	HC 273	HC 374	HCT 374	HC 534	HC 564	HC 574	HC 646	HC 648
#Pins	16	20	20	20	20	20	20	24	24
Type	D	D	D	D	D	D	D	D	D
Dual Device									
Quad Device	•								
Hex Device									
Octal Device		•	•	•	•	•	•	•	•
Common Clock	•	•	•	•	•	•	•	•	•
Negative-Transition Clocking									
Positive-Transition Clocking	•	•	•	•	•	•	•	•	•
Common, Active-Low Data Enables									
Noninverting Outputs	•	•	•	•			•	•	
Inverting Outputs	•				•	•			•
3-State Outputs			•	•	•	•	•	•	•
Common, Active-Low Output Enables			•	•	•	•	•	•	•
Common Reset	•	•							
Active-Low Reset									
Active-High Reset	•	•							
Active-Low Set									
Transceiver								•	•
Direction Control								•	•
LSTTL-Compatible Inputs				•					

These devices are identical in function and are different in pinout only: HC73 and HC107
HC76 and HC112
HC374 and HC574
HC534 and HC564

7

DIGITAL DATA SELECTORS/MULTIPLEXERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC151	8-Input Data Selector/Multiplexer	LS151	*4512	LS	16
HC153	Dual 4-Input Data Selector/Multiplexer	LS153	4539	LS/CMOS	16
HC157	Quad 2-Input Noninverting Data Selector/Multiplexer	LS157	*4519	LS	16
HC158	Quad 2-Input Inverting Data Selector/Multiplexer	LS158	*4519	LS	16
HC251	8-Input Data Selector/Multiplexer with 3-State Outputs	LS251	*4512	LS	16
HC253	Dual 4-Input Data Selector/Multiplexer with 3-State Outputs	LS253	*4539	LS/CMOS	16
HC257	Quad 2-Input Data Selector/Multiplexer with 3-State Outputs	LS257	*4519	LS	16
HC298	Quad 2-Input Data Selector/Multiplexer with Output Latch	LS298		LS	16
HC354	8-Input Data Selector/Multiplexer with Data and Address Latches and with 3-State Outputs	LS354, *LS356	*4512	LS354	20
HC356	8-Input Data Selector/Multiplexer with Data and Address Latches and with 3-State Outputs	*LS354, LS356	*4512	LS356	20

*Suggested alternative

Device	HC 151	HC 153	HC 157	HC 158	HC 251	HC 253	HC 257	HC 298	HC 354	HC 356
#Pins	16	16	16	16	16	16	16	16	20	20
Description	One of 8 inputs is selected	One of 4 inputs is selected	One of two 4-bit words is selected	One of two 4-bit words is selected	One of 8 inputs is selected	One of 4 inputs is selected	One of two 4-bit words is selected	One of two 4-bit words is selected	One of 8 inputs is selected	One of 8 inputs is selected
Single Device	•				•				•	•
Dual Device		•				•				
Quad Device			•	•			•	•		
Data Latch with Active-Low Latch Enable									•	•
Common Address		•	•	•		•	•	•		
1-Bit Binary Address			•	•			•	•		
2-Bit Binary Address		•				•				
3-Bit Binary Address	•				•				•	•
Address Latch (Transparent)									•	
Address Latch (Non-transparent)										•
Active-Low Address Latch Enable									•	•
Output Latch with Active-Low Latch Clock								•		
Noninverting Output	•	•	•		•	•	•	•	•	•
Inverting Output	•			•	•				•	•
3-State Outputs					•	•	•		•	•
Common Output Enable			•	•			•			
Active-High Output Enable									•	•
Active-Low Output Enable	•	•	•	•	•	•	•		••	••

7

DECODERS/ DEMULTIPLEXERS/ DISPLAY DRIVERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC42	1-of-10 Decoder	LS42	*4028	LS	16
HC137	1-of-8 Decoder/Demultiplexer with Address Latch	LS137	*4028	LS	16
HC138	1-of-8 Decoder/Demultiplexer	LS138	*4028	LS	16
HCT138	1-of-8 Decoder/Demultiplexer with LSTTL-Compatible Inputs	LS138	*4028	LS	16
HC139	Dual 1-of-4 Decoder/Demultiplexer	LS139	4556	LS/CMOS	16
HC147	Decimal-to-BCD Priority Encoder	LS147		LS	16
HC154	1-of-16 Decoder/Demultiplexer	LS154, *LS159	*4515	LS154	24
HC237	1-of-8 Decoder/Demultiplexer with Address Latch	*LS137	*4028	LS	16
HC259	8-Bit Addressable Latch/1-of-8 Decoder	LS259	*4099	LS	16
HC4511	BCD-to-Seven-Segment Latch/Decoder/Display Driver	*LS47, *LS48, *LS49	4511	CMOS	16
HC4514	1-of-16 Decoder/Demultiplexer with Address Latch	*LS154, *LS159	4514, *4515	LS/CMOS	24
HC4543	BCD-to-Seven-Segment Latch/Decoder/Display Driver for Liquid-Crystal Displays	*LS47, *LS48, *LS49	4543	CMOS	16

* Suggested alternative

Device	HC42	HC137	HC138	HCT138	HC139	HC147
#Pins	16	16	16	16	16	16
Input Description	BCD Address	3-Bit Binary Address	3-Bit Binary Address	3-Bit Binary Address	2-Bit Binary Address	Any Combination of 9 Inputs
Output Description	One of 10	One of 8	One of 8	One of 8	One of 4	BCD Address of Highest Input
Single Device Dual Device	•	•	•	•	 •	•
Address Input Latch Active-High Latch Enable Active-Low Latch Enable		 • •				
Active-Low Inputs						•
Active-Low Outputs Active-High Outputs	•	•	•	•	•	•
Active-Low Output Enable Active-High Output Enable		• •	•• •	•• •	•	
Active-Low Reset						
Active-Low Blanking Input						
Active-Low Lamp-Test Input						
Phase Input (for LCD's)						
LSTTL-Compatible Inputs				•		

Device	HC154	HC237	HC259	HC4511	HC4514	HC4543
# Pins	24	16	16	16	24	16
Input Description	4-Bit Binary Address	3-Bit Binary Address	3-Bit Binary Address	BCD Data	4-Bit Binary Address	BCD Data
Output Description	One of 16	One of 8	One of 8	7-Segment Display	One of 16	7-Segment Display
Single Device Dual Device	•	•	•	•	•	•
Address Input Latch Active-High Latch Enable Active-Low Latch Enable		• •		• •	• •	• •
Active-Low Inputs						
Active-Low Outputs Active-High Outputs	• 	 •	 •	 •	 •	 •
Active-Low Output Enable Active-High Output Enable	••	• •	•		•	
Active-Low Reset			•			
Active-Low Blanking Input				•		•
Active-Low Lamp-Test Input				•		
Phase Input (for LCD's)						•
LSTTL-Compatible Inputs						

7

ANALOG SWITCHES/ MULTIPLEXERS/ DEMULTIPLEXERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC4016	Quad Analog Switch/Multiplexer/Demultiplexer		4016,4066	CMOS	14
HC4051	8-Channel Analog Multiplexer/Demultiplexer		4051	CMOS	16
HC4052	Dual 4-Channel Analog Multiplexer/Demultiplexer		4052	CMOS	16
HC4053	Triple 2-Channel Analog Multiplexer/Demultiplexer		4053	CMOS	16
HC4066	Quad Analog Switch/Multiplexer/Demultiplexer with Enhanced On-Resistance Linearity		4066,4016	CMOS	14
☆HC4316	Quad Analog Switch/Multiplexer/Demultiplexer with Separate Analog and Digital Power Supplies		*4016		16
☆HC4351	8-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4051		18
☆HC4352	Dual 4-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4052		18
☆HC4353	Triple 2-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4053		18

* Suggested alternative

☆High-Speed CMOS design only

Device	HC4016	HC4051	HC4052	HC4053	HC4066
#Pins	14	16	16	16	14
Description	4 Independently Controlled Switches	A 3-Bit Address Selects One of 8 Switches	A 2-Bit Address Selects One of 4 Switches	A 3-Bit Address Selects Varying Combinations of the 6 Switches	4 Independently Controlled Switches
Single Device		•			
Dual Device			•		
Triple Device				•	
Quad Device	•				•
1-to-1 Multiplexing	•				•
2-to-1 Multiplexing				•	
4-to-1 Multiplexing			•		
8-to-1 Multiplexing		•			
Active-High ON/OFF Control	•				•
Common Address Inputs			•	•	
2-Bit Binary Address			•		
3-Bit Binary Address		•		•	
Address Latch with Active-Low Latch Enable					
Common Switch Enable		•	•	•	
Active-Low Enable		•	•	•	
Active-High Enable					
Separate Analog and Control Reference Power Supplies		•	•	•	
Switched tubs (for R_{ON} and Prop. Delay Improvement)					•

Device	HC4316	HC4351	HC4352	HC4353
#Pins	16	18	18	18
Description	4 Independently Controlled Switches. (Has a separate Analog Lower Power Supply)	A 3-Bit Address Selects One of 8 Switches. (Has an Address Latch)	A 2-Bit Address Selects One of 4 Switches. (Has an Address Latch)	A 3-Bit Address Selects Varying Combinations of the 6 Switches. (Has an Address Latch)
Single Device		•		
Dual Device			•	
Triple Device				•
Quad Device	•			
1-to-1 Multiplexing	•			
2-to-1 Multiplexing				•
4-to-1 Multiplexing			•	
8-to-1 Multiplexing		•		
Active-High ON/OFF Control	•			
Common Address Inputs			•	•
2-Bit Binary Address			•	
3-Bit Binary Address		•		•
Address Latch with Active-Low Latch Enable		•	•	•
Common Switch Enable	•	••	••	••
Active-Low Enable	•	•	•	•
Active-High Enable		•	•	•
Separate Analog and Control Reference Power Supplies	•	•	•	•
Switched tubs (for R_{ON} and Prop. Delay Improvement)				

SHIFT REGISTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC164	8-Bit Serial-Input/Parallel-Output Shift Register	LS164	*4034	LS	14
HC165	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register	LS165	*4021	LS	16
HC166	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with Reset	LS166	*4021	LS	16
HC194	4-Bit Bidirectional Universal Shift Register	LS194A	4194	LS/CMOS	16
HC195	4-Bit Universal Shift Register	LS195A	*4035	LS	16
HC299	8-Bit Bidirectional Universal Shift Register with 3-State Parallel Outputs	LS299		LS	20
☆HC589	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with 3-State Output	*LS597			16
HC595	8-Bit Serial-Input/Serial- or Parallel-Output Shift Register with Latched 3-State Outputs	LS595	*4034	LS	16
HC597	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with Input Latch	LS597		LS	16
HC4015	Dual 4-Bit Serial-Input/Parallel-Output Shift Register		4015	CMOS	16

*Suggested alternative
☆High-Speed CMOS design only

Device	HC164	HC165	HC166	HC194	HC195	HC299	HC589	HC595	HC597	HC4015
#Pins	14	16	16	16	16	20	16	16	16	16
4-Bit Register				•	•					•
8-Bit Register	•	•	•			•	•	•	•	
Serial Data Input	•	•	•	•	•	••	•	•	•	•
Parallel Data Inputs		•	•	•	•	•	•		•	
Serial Output Only		•	•				•		•	
Parallel Outputs	•			•	•	•		•		•
Inverting Output		•			•					
Noninverting Output	•	•	•	•	•	•	•	•	•	•
Serial Shift/Parallel Load Control		•	•	•	•	•	•		•	
Shifts One Direction Only	•	•	•		•		•	•	•	•
Shifts Both Directions				•		•				
Positive-Transition Clocking	•	•	•	•	•	•	•	•	•	•
Active-High Clock Enable		•	•							
Input Data Enable	•									
Data Latch with Active-High Latch Clock							•		•	
Output Latch with Active-High Latch Clock								•		
3-State Outputs						•	•	•		
Active-Low Output Enable						••	•	•		
Active-High Reset										•
Active-Low Reset	•		•	•	•	•		•	•	

7

COUNTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC90	4-Stage Binary Ripple Counter with ÷2 and ÷5 Sections	LS90		LS	14
HC92	4-Stage Binary Ripple Counter with ÷2 and ÷6 Sections	LS92		LS	14
HC93	4-Stage Binary Ripple Counter with ÷2 and ÷8 Sections	LS93		LS	14
HC160	Presettable BCD Counter with Asynchronous Reset	LS160A	4160	LS/CMOS	16
HC161	Presettable 4-Bit Binary Counter with Asynchronous Reset	LS161A	4161	LS/CMOS	16
HC162	Presettable BCD Counter with Synchronous Reset	LS162A	4162	LS/CMOS	16
HC163	Presettable 4-Bit Binary Counter with Synchronous Reset	LS163A	4163	LS/CMOS	16
HC190	Presettable BCD Up/Down Counter	LS190	*4510	LS	16
HC191	Presettable 4-Bit Binary Up/Down Counter	LS191	*4516	LS	16
HC192	Presettable BCD Up/Down Counter with Reset	LS192	*4510	LS	16
HC193	Presettable 4-Bit Binary Up/Down Counter with Reset	LS193	*4516	LS	16
HC390	Dual 4-Stage Binary Ripple Counter with ÷2 and ÷5 Sections	LS390		LS	16
HC393	Dual 4-Stage Binary Ripple Counter	LS393	*4520	LS	14
HC4017	Decade Counter/Divider		4017	CMOS	16
HC4020	14-Stage Binary Ripple Counter		4020	CMOS	16
HC4024	7-Stage Binary Ripple Counter		4024	CMOS	14
HC4040	12-Stage Binary Ripple Counter		4040	CMOS	16
HC4060	14-Stage Binary Ripple Counter with Oscillator		4060	CMOS	16
HC4518	Dual BCD Counter		4518	CMOS	16
HC4520	Dual 4-Bit Binary Counter		4520	CMOS	16

*Suggested alternative

Device	HC 90	HC 92	HC 93	HC 160	HC 161	HC 162	HC 163	HC 190	HC 191	HC 192
#Pins	14	14	14	16	16	16	16	16	16	16
Single Device Dual Device	•	•	•	•	•	•	•	•	•	•
Ripple Counter Number of Ripple Counter Internal Stages Number of Stages with Available Outputs	• 4 4	• 4 4	• 4 4							
Count Up Count Down	•	•	•	•	•	•	•	• •	• •	• •
4-Bit Binary Counter BCD Counter Decimal Counter	•		•	•	•	•	•	•	•	•
Separate ÷2 Section Separate ÷5 Section Separate ÷6 Section Separate ÷8 Section	• •	• •	• •							
On-Chip Oscillator Capability										
Separate Count-Up and Count-Down Clocks										•
Count Up/Count Down Control Input								•	•	
Positive-Transition Clocking Negative-Transition Clocking Active-High Clock Enable Active-Low Clock Enable	•	•	•	•	•	•	•	•	•	•
Active-High Count Enable Active-Low Count Enable				••	••	••	••	• •	• •	
Active-High Set Active-High Reset	• •	•	•	•	•	•	•			•
4-Bit Binary Preset Data Inputs BCD Preset Data Inputs Active-Low Load Preset				• •	• •	• •	• •	• •	• •	• •
Carry Output Borrow Output Ripple Clock Output				•	•	•	•	• •	• •	• •

Device	HC 193	HC 390	HC 393	HC 4017	HC 4020	HC 4024	HC 4040	HC 4060	HC 4518	HC 4520
#Pins	16	16	14	16	16	14	16	16	16	16
Single Device	•			•	•	•	•	•		
Dual Device		•	•						•	•
Ripple Counter		•	•		•	•	•	•		
Number of Ripple Counter Internal Stages		4	4		14	7	12	14		
Number of Stages with Available Outputs		4	4		12	7	12	10		
Count Up	•	•	•	•	•	•	•	•	•	•
Count Down	•									
4-Bit Binary Counter	•		•							•
BCD Counter		•							•	
Decimal Counter				•						
Separate ÷2 Section		•								
Separate ÷5 Section		•								
Separate ÷6 Section										
Separate ÷8 Section										
On-Chip Oscillator Capability								•		
Separate Count-Up and Count-Down Clocks	•									
Count Up/Count Down Control Input										
Positive-Transition Clocking	•			•					•	•
Negative-Transition Clocking		•	•	•	•	•	•	•	•	•
Active-High Clock Enable				•					•	•
Active-Low Clock Enable				•					•	•
Active-High Count Enable										
Active-Low Count Enable										
Active-High Set										
Active-High Reset	•	•	•	•	•	•	•	•	•	•
4-Bit Binary Preset Data Inputs	•									
BCD Preset Data Inputs										
Active-Low Load Preset	•									
Carry Output	•									
Borrow Output	•									
Ripple Clock Output										

MONOSTABLE MULTIVIBRATORS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC123	Dual Retriggerable Monostable Multivibrator	LS123	*4538, *4528	LS	16
HC221	Dual Monostable Multivibrator	LS221	*4538, *4528	LS	16
HC423	Dual Retriggerable Monostable Multivibrator	LS423	*4538 *4528	LS	16
HC4538	Dual Precision Monostable Multivibrator (Retriggerable, Resettable)	*LS423	4538, 4528	CMOS	16

*Suggested alternative

Device	HC123	HC221	HC423	HC4538
#Pins	16	16	16	16
Dual Device	•	•	•	•
Precision Pulse Width				•
Retriggerable	•		•	•
Positive-Transition Trigger	•	•	•	•
Negative-Transition Trigger	•	•	•	•
Active-Low Trigger Enable	•	•	•	•
Active-High Trigger Enable	•	•	•	•
Active-Low Reset	•	•	•	•
Triggerable by Reset Pin	•	•		
Inverting Output	•	•	•	•
Noninverting Output	•	•	•	•

7

ARITHMETIC CIRCUITS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC85	4-Bit Magnitude Comparator	LS85	*4585	LS	16
HC181	4-Bit Arithmetic Logic Unit	LS181	4581	LS/CMOS	24
HC182	Carry Lookahead Generator	LS182	4582	LS/CMOS	16
HC280	9-Bit Odd/Even Parity Generator/Checker	LS280	*4531	LS	14
HC283	4-Bit Binary Full Adder with Fast Carry	LS283, LS83	4008	LS283	16
HC688	8-Bit Equality Comparator	LS688		LS	20
HCT688	8-Bit Equality Comparator with LSTTL-Compatible Inputs	LS688		LS	20

*Suggested alternative

MISCELLANEOUS DEVICES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC292	Programmable Frequency Divider/Digital Timer	LS292		LS	16
HC294	Programmable Frequency Divider/Digital Timer	LS294		LS	16
HC4046	Phase-Locked Loop	*LS297	4046	CMOS	16

*Suggested alternative

LSTTL INPUT-COMPATIBLE DEVICES

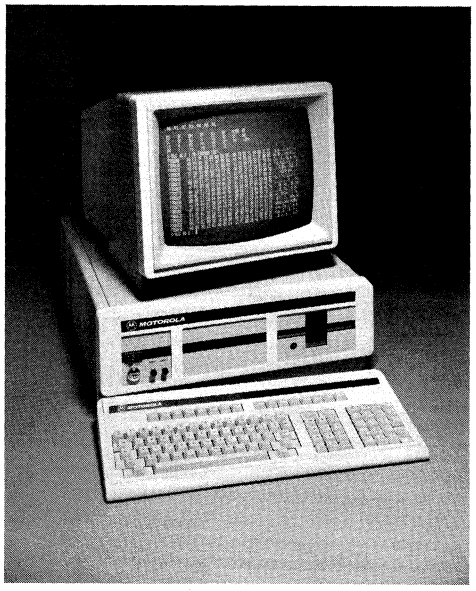
Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HCT00	Quad 2-Input NAND Gate with LSTTL-Compatible Inputs	LS00	4001	LS	14
HCT04	Hex Inverter with LSTTL-Compatible Inputs	LS04	*4069	LS/CMOS	14
HCT34	Hex Buffer with LSTTL-Compatible Inputs	LS07	*4050	LS	14
HCT138	1-of-8 Decoder/Demultiplexer with LSTTL-Compatible Inputs	LS138	*4028	LS	16
HCT240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS240		LS	20
HCT241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS241		LS	20
HCT244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS244		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HCT373	Octal 3-State Noninverting D-Type Transparent Latch with LSTTL-Compatible Inputs	LS373, LS573		LS373	20
HCT374	Octal 3-State Noninverting D-Type Flip-Flop with LSTTL-Compatible Inputs	LS374, LS574		LS374	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HCT688	8-Bit Equality Comparator with LSTTL-Compatible Inputs	LS688		LS	20

*Suggested alternative

Development Systems and Board-Level Products

8

VME/10 MICROCOMPUTER SYSTEM



The VME/10 Microcomputer System is a compact yet powerful desktop designer's workstation that can be used for developing advanced microprocessor-based systems using Motorola's 8-bit and 16-bit families of microprocessors, microcomputers, and peripheral components.

MAJOR BENEFITS

- Provides Efficient Design Support for M6800 and M68000 MPU Families
- Excellent Development Software Complement
- Customizable Through VMEbus and I/O Channel for End Applications
- Multi-mode Graphics Hardware with Both Monochrome and Color Options.

The VME/10 Microcomputer System combines the flexibility of a customizable workstation with the attributes of a powerful development support system that let the system integrator or OEM design an end product with the same hardware and software that can eventually constitute the end system itself. With appropriate interfaces and peripherals, these systems may be specialized designers' workstations, or perhaps front-end processors associated with larger external equipments such as factory automation systems or large complex medical diagnostic instruments. In addition to raw processing power, these small but capable systems have the flexibility for just the right I/O and performance improvement features for dedicated, user-defined systems.

BASIC DESIGN FEATURES

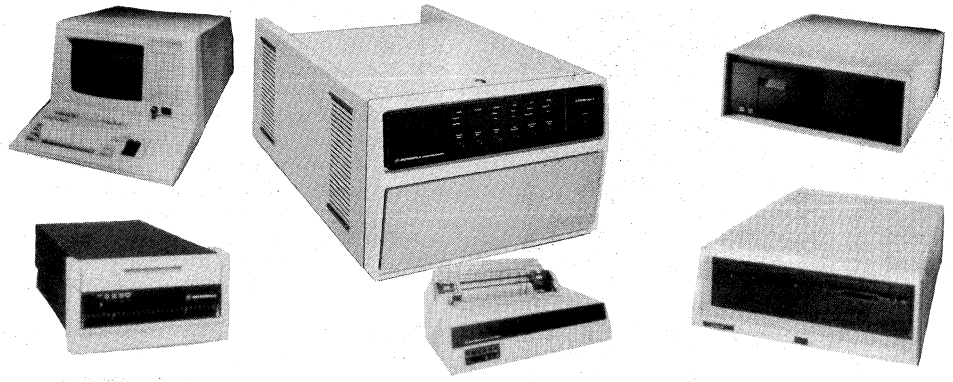
- MC68010 16/32-bit Microprocessor Unit
- MC68451 Memory Management Unit
- Industry-standard VMEbus interface with full bus arbitration logic and software controllable interrupter.
- I/O Channel Interface for adding off-board resources such as A/D converters, serial and parallel I/O ports, etc.
- 384K Byte Dynamic RAM (multiported between graphics controller and local bus, and VMEbus).
- 8K Byte Static RAM for storage of user-definable character sets and display attributes.
- Two 28-pin sockets for ROM/PROM/EPROM storage of up to 64K bytes for custom applications.
- Battery backed-up time-of-day clock with 50 bytes of CMOS RAM storage.
- 15″ video display having the following software controllable display formats:
  1. 25 lines by 80 characters — 8 x 10 characters with descenders (10 x 12 character field)
  2. 800 x 300 pixel for low resolution graphics
  3. 800 x 600 pixel for medium resolution graphics
  4. Pixel graphics with overlaid character displays
- Monochrome video display standard, with 7-level gray scaling (color optional).
- Detachable full ASCII keyboard with cursor control keys, numeric pad and 16 function keys.
- Mass Storage Subsystem providing both 5¼″ Floppy Disk and 5¼″ Winchester Disk Storage Units.

  Floppy Disk
  1 Mbyte Unformatted Capacity (655K Byte Formatted)

  Winchester Disk
  Choice of: (a) 6.38 Mbyte Unformatted Capacity (5 Mbyte Formatted)
  (b) 19.1 Mbyte Unformatted Capacity (15 Mbyte Formatted)
- Card cage options for feature expansion capability.
  Choice of: (a) Five I/O Channel Card Cage Slots (with 6.38 Mbyte Winchester option)
  (b) Five VMEbus Card Cage Slots with VMEbus backplane, plus four I/O Channel Slots (with 19.1 Mbyte Winchester option)
- Conformance to ergonomic standards applicable to video display and keyboard.
- VERSAdos Real-Time, Multitasking Operating System with M68000 Family Macro Assembler, plus tools and utilities.
- Capability of hosting hardware development tools
  — HDS-400 for M68000 Family 16/32-bit Emulation
  — HDS-200 for M6800 Family 8-bit Emulation
  — Bus State Analyzer for Logic Analysis Functions

EXORmacs

M68000 DEVELOPMENT SYSTEM



- Complete Development System for MC68000 MPU
- Up to Eight User Stations
- Multi-Processor Bus Arbitration
- Multi-Tasking Real-Time Operating System
- Resident Pascal High-Level Language
- Diagnostic Firmware
- Up to 192 Megabyte Fixed/Removable Hard Disk
- And Up To 2 Megabyte Dual Drive Floppy Disk
- Provisions For Future 32-Bit Microprocessors
- Optional Cross-Development Software for 8-Bit MPUs

The EXORmacs Development System is a state-of-the-art instrument for designing and developing advanced 16-bit microprocessor based systems using Motorola families of microprocessors, microcomputers, and peripheral components.

Coupled with the Motorola HDS-400 Microprocessor Hardware/Software Development Station it is also ideally suited for developing applications using the VERSAmodule and VMEmodule families of 16-bit board level application products and accessories.

Designed for flexibility and ease of use, the EXORmacs Development System takes advantage of the power and features of the MC68000 microprocessor unit (MPU). It reduces cost and development time by incorporating features which support 16-bit and future 32-bit microprocessor designs, as well as providing high-level language support through Pascal and FORTRAN. With an appropriate number of accessories, such as terminals, multiple-channel communications modules and hardware development stations, up to eight users may simultaneously develop and debug M68000 programs.

System Expansion Modules

Multichannel Communications
Module (MCCM) — M68KMCCM

VERSAbus Adapter Module — M68KVAM
VERSAbus RAM 128K Byte — M68KVM10-3
VERSAbus RAM 256K Byte — M68KVM11-1
VERSAbus RAM 512K Byte — M68KVM11-2
VERSAbus Extender — M68KEXTM
VERSAbus Wirewrap — M68KWW

EXORmacs Basic System Configurations

- Hardware Chassis — with Power Supply and 15-slot Backplane
  Resident Module Complement
  MC68000 MPU/MMU Module
  DEbug Module
  256K Dynamic RAM
  Disk Controller Module

- Software
  System V/68 Operating System Software
  M68000 System V/68 OS
  M68000 C Language Compiler Assembler and Linker
  Instrumentation Support Utilities
  VERSAdos Software Development Tools
  VERSAdos Operating System
  CRT Editor
  Macro Assembler
  Linkage Editor
  Symbolic Debug

- Peripherals
  EXORterm 155 Display Console
  Choice of Mass Storage:
  1 Megabyte Floppy Disk
  8/8 Megabyte Hard Disk
  25/25 Megabyte Hard Disk
  16/16 Megabyte Hard Disk
  16/80 Megabyte Hard Disk

VMC 68/2 Microcomputer

The VMC 68 Series is a high performance microcomputer system family intended for application by OEM's and system integrators starting from a product integration level formerly available only to the minicomputer user. The VMC 68 System Family will find wide application in industrial process control, automated testing, data acquisition, supervisory control, and many other factory and lab automation uses. The VMC 68 Series is based on the 16-bit M68000-based VERSAmodule Family of modular microcomputer products utilizing the industry and IEEE proposed VERSAbus standard system interconnect providing multiprocessing and intelligent peripheral controller architecture.

Also featured is the I/O channel which provides for the use of a broad selection of I/O modules for I/O flexibility.

VMC 68/2 System Features and Capabilities

Hardware-Only Package	Complete System Package
• MC68000-based VM02 Monoboard Microcomputer — Direct Addressing to 16 Megabytes — 128K Bytes Dual-Port RAM — Multiprocessor Architecture with System Controller Features — I/O Channel Interface for Functional Tailoring — Dual Multiprotocol RS-232C Serial Ports for System Flexibility • Dual 16-Bit Parallel Port I/Omodule — Centronics compatible Printer Interface — General Parallel I/O Applications • VERSAbug Firmware — Debug — Disk Bootstrap Load — Self-Test — Up/Downline Load • MC68120-based VM21 Universal Disk Controller — High-speed DMA data transfer to/from 1 or 2 SMD interface compatible disk drives AND Up to 4 EXORdisk II or III Floppy Disk Drives • VM11 Dynamic RAM Module with 256K Bytes of "global" RAM for program development and efficient multitasking system operation • 4 or 8-slot VERSAbus compatible VMC 68/2 Chassis — Power Fail/Restart Circuitry — 5 or 10 I/Omodule card slots for I/O Channel functional tailoring (Dual Parallel Port module occupies one of these slots) • 0°C to 50°C (32°F to 122°F) Operating Temperature Range • For 115 Vac 60 Hz Operation	In addition to all features of the Hardware-Only package: • MLD-16 Mass Storage Unit incorporating Disk Drive, SMD interface electronic module, and Disk Power Supply • 16 Megabyte (unformatted) 8-inch SMD interface compatible Disk Drive — 8M Bytes Fixed, plus 8M Bytes Removable Cartridge for storage and one-to-one System Backup — Embedded Servo Information to eliminate cartridge interchange problems and the need for head alignment — Simple Installation — Quiet Operation — High Performance — Disk Compartment sealed during operation — Exceptional Reliability (7500 Hour MTBF) — Long Service Life requiring no preventive maintenance in a benign environment • VERSAdos Real-Time Multitasking Operating System with Assembler and Utilities, including: — MC68000 Structured Macro Assembler — Text Editor, Linkage Editor, and Multitasking Debugger — System Diagnostics — System Generation (SYSGEN) capability for feature tailoring of the VERSAdos System • +10°C to +40°C (50°F to 104°F) Operating Temperature Range

Ordering Information

MVMC682-114	Four-slot VMC 68/2 Microcomputer System
MVMC682-118	Eight-slot VMC 68/2 Microcomputer System
MVMC682-114H	Four-slot VMC 68/2 Microcomputer System hardware-only package.
MVMC682-118H	Eight-slot VMC 68/2 Microcomputer System hardware-only package.

M68000 System Development Software

SYSTEM V/68 AND VERSAdos OPERATING SYSTEMS

System V/68

The System V/68 Operating System is the standard UNIX-derived Operating System for the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. In addition, a powerful command shell for interactive system controls and an extensive set of utility programs for many tasks, such as program development, text processing, electronic mail, and networking support are included.

Host Systems

The System V/68 Operating System is available as the host environment on Motorola development systems. The EXORmacs is a multiuser system capable of supporting up to eight users simultaneously. The VME/10 System is a single-user system. Hard disk is required for System V/68. Future Motorola Microsystems development systems will also be supported by the System V/68 Operating System.

Instrumentation Support

Communications support for the Motorola HDS-400 Hardware Development Station is included in System V/68. This provides customers with the full system development capability (both hardware and software) that they have come to expect from Motorola.

Languages

As an integral part of System V/68, C Language is offered. C Language has developed into one of the most popular commercial programming languages, and is used frequently in developing portable application software. System V/68 offers significant enhancements to C Language, along with several new language utilities. CXREF, a new cross reference program, and CFLOW, a new flow analysis program, are just two of the new utilities offered. System V/68 also includes a FORTRAN 77 compiler as well as an M68000 assembler and linker/loader.

Programmer's Workbench

The Programmer's Workbench utilities support the development of large software systems in a professional manner. They include the Source Code Control System (SCCS), which provides facilities to store, update and retrieve all versions of source code modules; YACC, which generates parsers; LEX, which builds lexical analyzers; and other utilities which enhance programmer productivity and the quality of work.

VERSAdos

The M68000 Real-time Operating System (VERSAdos) provides complete real-time, multitask support for the EXORmacs User. Features included in the VERSAdos are:

- Real-time multitasking executive
- Device independent I/O
- Floppy and hard disk support
- Sequential, random, and index sequential file capabilities.

CRT Text Editor

The EXORmacs CRT-oriented Text Editor runs under the supervision of the Operating System and provides the capability to create and modify source programs. The editor supports both command and cursor editing, utilizing the cursor, control characters and function keys of the EXORterm 155.

Structured Assembler

The M68000 Structured Macro Assembler translates source statements into relocatable machine code, assigns storage locations to instructions and data, performs auxiliary assembler actions designated by the programmer, and optionally produces a cross-reference listing. The M68000 resident assembler includes macro and conditional assembly capabilities plus certain control constructs that permit structured programming at the assembly language level.

Linkage Editor

The Linkage Editor provides the capability of merging two or more separately-compiled object units into a loadable object module file.

Symbolic Debug

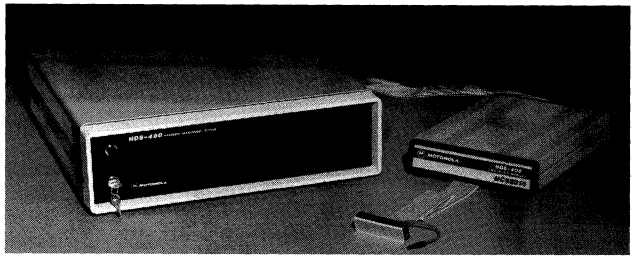
The SYMbug/A program is used to debug other programs, whose source code may have been written in Motorola-provided assembler language, for execution on the M68000. The language processors, in cooperation with the Linkage Editor, supply symbolic information to SYMbug/A. This permits the user to describe the debugging requirements to SYMbug in terms close to the language in which the source program was written.

Pascal Compiler (Optional) M68K0PASCALH

Pascal is a block structured high order language that promotes good programming technique, is self-documenting, and simplifies program writing.

FORTRAN Compiler (Optional) M68K0FORTRNH

Motorola's FORTRAN exceeds ANSI FORTRAN 77 subset language specification, providing real-time processing capabilities.



Control Station
M68KHDS400
M68KHDS400A

Family Board
M68KHDS16FB

Emulators
M68000HDS4
M68008HDS4-8
M68010HDS4-8

Software
M68KHDS4-XX

HDS-400 MICROPROCESSOR HARDWARE/SOFTWARE DEVELOPMENT STATION

Design Features

- 12.5 MHz Real-Time Emulation for MC68000 MPUs
- 10 MHz Real-Time Emulation with no Wait Cycles for MC68000 MPUs
- 8.0 MHz Emulation for MC68008 and MC68010 MPUs
- No User Target System Restrictions
- 32K bytes of 10 MHz No Wait Cycle Emulation RAM is Standard
- Emulation RAM Expandable to 64K, 128K or 256K bytes
- Full Symbolic Debug with EXORmacs and VME/10 Hosts
- Unrestricted User Memory Map
- One-Line Assembler/Disassembler
- Automatic Self-Test of Development Station Hardware
- M68KHDS400 Interfaces with EXORmacs Development System
- M68KHDS400A Interfaces with Motorola VME/10 and DEC VAX Hosts
- Compatible with Real-Time Bus State Analyzer

Major Benefits

- Reduces Development Costs
- Shortens Product Development Cycle
- Brings Product to Market Faster
- Versatility Protects Against Obsolescence

The HDS-400 Microprocessor Hardware/Software Development Station, in conjunction with a Motorola EXORmacs Development System or VME/10 Microcomputer System, or a DEC VAX Computer, provides a complete hardware/software development system for the Motorola M68000 family of microprocessors. It consists of a Control Station, with all the support circuitry for complete MPU emulation, and a separate Emulator Module with an internal microprocessor to match the particular MPU it is expected to emulate.

Two key capabilities of the HDS-400 make it very useful as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected MPU in the user's target system. By plugging the HDS-400 into the socket on the prototype hardware, it allows efficient testing and debugging of both hardware and software. The second capability is the rapid debug and integration of the target system for the production of prototypes. This is accomplished by the use of the powerful set of commands in the HDS-400. The user may execute the commands by either entering the command code and its parameters, or by sequentially depressing function keys which provide a "fill-in-the-blanks" format with parameters such as file name, address, data, etc. When a single function key or a combination of function keys is pressed, a command code is automatically generated and the command syntax is displayed by the system.

Typical System Configuration

The HDS-400 Hardware/Software Development Station includes a four-slot Control Station with a built-in 30 A power supply and an Emulator Module for the specific MPU which will be used in the target system. Emulators are available for the MC68000, MC68008, and MC68010 MPUs.

The HDS-400 has been partitioned with options and part numbers that give the user versatility in defining the development system configuration. The user may choose from three host computers EXORmacs, VME/10, or DEC VAX with a variety of operating systems. Each of the HDS-400 Control Stations is delivered pre-wired to accept the optional Emulation Memory Module and the Real-Time Bus State Analyzer (BSA). EXORterm 155 is required in HDS-400 systems hosted by the EXORmacs and the VAX. The VME/10 functions as both host and terminal to the HDS-400, eliminating the need for a separate terminal in VME/10-based systems.

System Performance

The HDS-400 Development Station, when substituted for the MPU chip in the target system being debugged, performs the functions of the microprocessor being emulated — exactly as the MPU would have performed were it still in the circuit being tested. The emulator provides the interfacing with the RAM, ROM, and I/O devices and operates at the same speed as the MPU. There are no restrictions on the use of emulation memory that are not imposed by the MPU itself, and the memory may be mapped to the target system or to the emulator module.

The standard 32K bytes of emulation RAM provided in the Family Interface Module may be expanded with one of three optional Emulation Memory Modules. The three memory expansion modules available increase the 10 MHz no wait cycle emulation RAM to 64K, 128K or 256K bytes.

Control Station
M68HDS201

Emulators
M6804P2HM
M6805P234HM
M6805RU23HM
M6805S2HM
M6805T2HM
M146805E2HM
M146805F2HM
M146805G2HM

HDS-200 MICROCOMPUTER HARDWARE/SOFTWARE DEVELOPMENT STATION

Design Features

- Real-time emulation for M6804/M6805/M146805 MCUs.
- Sixteen programmed breakpoints.
- Prioritized breakpoints.
- Line-by-line assembler/disassembler.
- Program trace commands.
- Commands displayed for operator HELP.
- Memory map display.
- Macro commands stored for re-use.
- Transparent mode for host communication.
- Emulates more than 20 MCUs.

Operating Features

- Compatible with EXORmacs, EXORciser and EXORset software development systems.
- Low cost.
- Stand alone operation — frees software development system for parallel use.
- Easy to use.
- Operates with any standard RS-232C terminal and most host systems.

The HDS-200 Hardware Development System, in conjunction with a Motorola EXORset, EXORmacs or EXORciser software development system, provides a complete hardware/software development system for the Motorola M6804/M6805/M146805 families of microprocessors. It consists of a Control Station, with all the functional circuitry to complete MCU emulation, and a separate Emulator Module with an internal microcomputer and memory capacity to match that of the particular MCU it is expected to emulate.

Two major factors contribute to the HDS-200's usefulness as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected MCU in the target system. By plugging the HDS-200 into the socket on the prototype hardware, it allows efficient testing of hardware as well as software debugging. The second factor is its powerful list of analysis commands. These easy-to-use, plain language commands enable the user to rapidly debug, integrate the target system and produce prototype systems.

Systems Development and Integration

The initial stages of developing an MCU-based system normally involve two parallel, rather independent, efforts. One is the hardware design — the other the software design. These efforts are frequently accomplished by two different teams of personnel, resulting in debugging problems that are often difficult and time-consuming. The HDS-200 simplifies this process because of its ability to bring the hardware and software development processes into intimate relationship with each other throughout the development cycle. Moreover, with the HDS-200 it becomes economically feasible to test alternate design approaches in order to determine the best solution.

The complete HDS-200 Hardware/Software Development System consists of three separate items — the HDS-200 Control Station, the Emulator Module, and an associated Firmware Cartridge.

HDS-200 Control Station

The station contains an internal power supply, logic circuits, clock and an MC6809 MPU. The MPU runs the monitor, controls the ports and interfaces with the emulators. It has two RS-232C communication ports for interconnecting with a host computer and a suitable terminal. Another cable connects the station to an associated outboard Emulator Module.

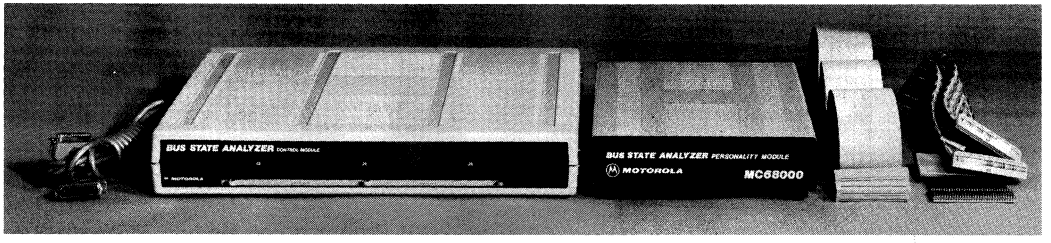
The Emulator Module

The module's output to the user's system is by a short, noisefree ribbon cable terminating in a plug to mate with the target hardware MCU socket. The emulator contains the target processor and various I/O interfaces to provide a compatible link between the Control Station and target hardware MCU/MPU socket.

Different modules are available for specific microprocessor family types. The various MCU Emulator Modules available include the M6804, M6805, M68705, and M146805 families. Each module comes with a matching Firmware Cartridge and an emulator cable/connector assembly.

The Firmware Cartridge

Paired with each Emulator Module is a small cartridge which is easily plugged into the HDS-200 station. This cartridge contains the necessary programs on ROM to enable the HDS-200 to adapt to the specific "personality" of the selected MCU type.



REAL-TIME BUS STATE ANALYZER

The Real-Time Bus State Analyzer (BSA) is a highly intelligent diagnostic tool that is designed specifically for use with microprocessors. It consists of a Control Module plus one of several "Personality Modules." The Control Module contains the analyzer hardware, control firmware, and I/O ports. The Personality Modules interface to selected MPU/MCU, EXORbus, or VERSAbus signals. The BSA stores data which appears on between 55 and 79 different lines.

In order to facilitate the gathering of pertinent data from the MPU/MCU or bus, a set of qualified triggering modes are provided. These modes can be broken into three categories: Continuous Trace Mode, Sequential Trigger Mode, and Window Trigger Mode.

CONTINUOUS TRACE MODE samples signals and stores signal information continuously on each occurrence of the clocking signal. It is primarily a default mode which the BSA automatically enters when power is first applied. There are no qualifications for the BSA to begin gathering information, so it will always be storing the signal states. This default mode is particularly useful when a sudden catastrophic failure occurs during a debugging session, before the user is able to configure the BSA. It is very likely that the events leading up to the failure will be stored in the BSA's trace store buffer.

SEQUENTIAL TRIGGER MODE requires that a series of events occur before the instrument triggers and starts to gather data; or conversely triggers and stops gathering data. Sequence Terms, as these events are called, must occur in order of specification, or triggering will not take place. A Sequence Reset Term can also be specified to reset the BSA and cause the instrument to begin looking for the Sequence Terms again. Sequential triggering will be most useful for debugging complex software, including loops, nested subroutines and complex branches.

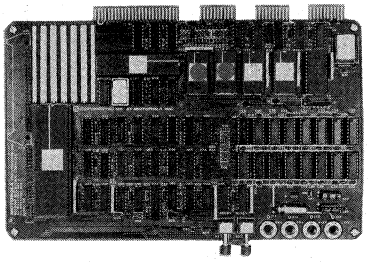
WINDOW TRIGGER MODE provides a means of causing signal states to be stored if address accesses occur inside or outside of a particular address range. Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map. Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating. It is also applicable for observing access violations in a multiple user environment.

SOFTWARE PERFORMANCE HISTOGRAMS are also provided to give an indication of the relative frequency of memory accesses within a particular memory range, with the exact range specified by the user. This histogram provides a means of determining where a program spends the greatest amount of time. The resulting information can then be used to compress inefficient code. A hardware performance histogram is provided to display the relative frequency of combinations of four user-selected signals within a user system.

In order to service these triggering modes and provide a complete set of operating features, an MC6809 microprocessor is located on the Control Module with local intelligence running from an operating system based in ROM. This operating system provides the data analysis and formatting functions for the operator including the interface to the hardware sampling the bus.

To reduce system redundancy, the terminal used by the operator to communicate with the development system will also serve to link him to the intelligence aboard the analyzer (it is a requirement of the CRT-based analyzer operating system that the terminal used be an EXORterm 155 Display Console). This communication will be achieved by means of a phantom or transparent serial link feeding from the terminal through the Bus State Analyzer control board and then to the normal terminal input channel of the development system. The logic onboard the BSA determines whether the information traveling over the link is destined for the development system, the Bus State Analyzer or the terminal. This allows the operating system or the user's software to run in the development system while analysis is being performed. Additionally, a means is provided for the BSA to operate in a stand-alone mode with only a terminal connected.

Part Number	Description
M68BSAC	BSA Control Module for use with BSA Personality Modules
M68BSACE	BSA Control Module with Enclosure
M68BSA1-1	BSA Personality Module for MC68000, MC68010 and MC68451
M68BSA2	BSA Personality Module for MC6800, MC6809, and MC6829
M68BSA3	BSA Personality Module for MC68008
M68BSA4	BSA Personality Module for MC6801 and MC68120
M68BSA5	BSA Personality Module for VERSABus
M68BSA6	BSA Personality Module for EXORbus



LOW COST MPU/MCU EVALUATION MODULES

A series of inexpensive evaluation modules are available for Motorola's line of microprocessors and microcomputers. Evaluation modules allow the user to prepare, debug, and run software in the resident microcomputer. Even though the cost is low, an onboard ROM contains extensive commands for controlling I/O and debug operations, including down-up load S-record transfers.

Memory, internal registers, and I/O registers may be displayed and modified. Program execution may be traced one step at a time or breakpoints may be inserted for program interruption. Circuitry and firmware are included to allow the MCU's EPROM to be programmed.

MC68000 Educational Computer Board MEX68KECB

The MC68000 Educational Computer Board (ECB) serves as an economical introduction to systems based on the M68000 family of microcomputer products.

The ECB is based around a 4 MHz MC68000 MPU. Also provided are 32K bytes of RAM, arranged as 16K x 16. The firmware is contained in two 8K by 8 ROMs, addressed as an 8K by 16 block of memory. Two RS-232C serial ports are implemented with MC6850 ACIA's and an MC14411 baud rate generator, allowing selection of data rates from 110 to 9600 baud.

One of the M68000 peripherals, the MC68230 Parallel Interface and Timer (PI/T), provides a Centronix-type parallel printer interface and an audio cassette interface. An audio cassette recorder may be used to store and retrieve user programs.

The ECB uses a terminal, interfaced via one RS-232C port. Also, a small wirewrap area is provided for system I/O modification and buffering.

MC6801 Evaluation Module MEX6801EVM

The MC6801 Microcomputer Evaluation Module is a completely self-contained microcomputer on a single printed circuit card, providing the user with the means of evaluating the MC6801 microcomputer. As configured, the MC6801 may be evaluated in the Single-Chip mode by attaching an RS-232C-compatible terminal to the serial port of the module. Thus, the minimum functioning system consists of only the MC6801 and an MC1488 and MC1489 (RS-232C interface).

In the Expanded mode, the customer may add an ACIA, PTM, 4K bytes RAM or 2K EPROM and a programmable gate array for address configuration.

MC68701 Programming Module MEX6801EV1

This module has the same features as the MC6801 module but is also populated with an MC68701, 2K bytes of RAM, a programmed gate array, and a DEbug monitor (PRObug) which also provides the programming capability for the MC68701 EPROM device. It, also, can be used to evaluate the MC6801 microcomputer.

MC68120 Evaluation Module M68120EVM

The M68120 Evaluation Module is designed to assist the potential user of an MC68120 Intelligent Peripheral Controller (IPC) chip in developing software, performing limited circuit emulation, and operating as a serially-linked design on an EXORbus compatible board format.

All data communications are accomplished via two RS-232C ports. Consequently, the Evaluation module can be operated in a stand-alone configuration with only power brought in on the EXORbus connector. An optional operating configuration allows the M68120 Evaluation Module to be plugged into an EXORciser II or an EXORmacs VERSAbus System via a VERSAbus Adapter Module (VAM). The dual-ported 128-byte RAM can then be mapped into a local map or system map.

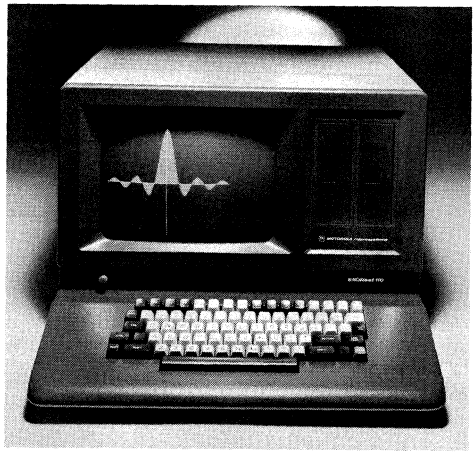
There is 4K of RAM populated on the board local bus along with decoding to permit an additional 4K RAM to be implemented in the user wirewrap area. The RAM allows user software development and debug for future programming of 2K, 4K, and 8K EPROMs to be inserted on the M68120 Evaluation Module. The Module has 64K bytes of address space on the local bus and 256 bytes of address space on the system bus.

MC68705 Evaluation Module M68705EVM MC1468705 Evaluation Module M1468705EVM

Operation of an MC68705 or MC1468705 is simulated by the resident MC6805 or MC146805 MCU. Data transfer within the EVM is controlled by the monitor ROM firmware. In turn, this ROM is controlled from an external RS-232C compatible user terminal. User object code may be down-loaded to the user program RAM via the host port; a cassette port is also provided for this purpose. The host and terminal port ACIAs are baud rate strap-selectable from 110 bps to 19.2 Kbps in eight steps.

The MCU parallel I/O ports allow the user to connect externally to the simulated MCU I/O lines. These lines are also used to control the MC68705 or MC1468705 MCUs on-chip EPROM programmer. This is accomplished by inserting the MCU into the programmer socket and executing the appropriate monitor commands.

EXORset 110



EXORset 110 Features

- MC6809 high performance microprocessor.
- Full ASCII Keyboard with 16 user-definable function keys.
- 12" CRT displaying 22 lines of 80 characters, or switchable to 16 lines of 40 characters and/or full graphics. 2K bytes of static RAM are included for CRT character refresh.
- 56K bytes of RAM and three sockets for up to 24K bytes of EPROM/ROM.
- Three card slots for EXORciser/Micromodule boards, four if no disk controller needed.
- Printer interface.
- Serial I/O port.
- EXORbug monitor/debug ROM included. An additional EPROM/ROM socket is available if user does not require EXORbug.
- Triple 16-bit programmable counter/timer included with input Gate and Clock signals and output signals available to the user.
- Meets FCC compliance for a Class A computing device.

A High Performance Processor

The EXORset controller is based on the new generation 16/8-bit microprocessor MC6809. The expanded instruction set, addressing modes, and architecture make execution of software particularly efficient and allow sophisticated programming techniques such as structured programming, position independent code, re-entrant routines and real-time operations. These capabilities make the 6809 microprocessor suitable for high-level language program development.

A CRT Display and Keyboard

The EXORset unit provides the user with a complete man/machine interface consisting of a full-size ASCII keyboard and 16 user-assigned function keys and a high resolution 12" CRT display capable of displaying 22 lines of 80 or 16 lines of 40 upper or lower case characters and simultaneously a full 320 x 256 dot graphic image.

Memory Flexibility

The EXORset controller allows for flexibility in the type and amount of memory to be used in the application. Three versions are available that provide optional amounts of mass storage: no floppy disk drives, with one double-sided mini-floppy disk drive for 160K bytes of mass storage and with two disk drives for 320K bytes of mass storage. All three versions include 2K bytes of dynamic RAM for CRT character refresh, 56K bytes of dynamic RAM and three strappable sockets that can be configured for 1K, 2K, 4K or 8K ROMs or EPROMs. A fourth socket, normally containing the 4K EXORbus firmware, can be configured for a user-designed monitor routine. The EXORset memory map is defined by PROMs, allowing the user to easily reconfigure the architecture of the system. Optional configurations information is available by contacting your local Motorola sales office.

On-Board Input/Output Ports

The EXORset unit provides three on-board input/output ports. An asynchronous serial communication port is provided with strap-selectable interface options of RS-232C, RS-422 or RS-423 and can be configured as a terminal or as a modem. The baud rate is software programmable from 110 to 19.2K baud. The user may also replace the asynchronous device with an SSDA device for synchronous communication application.

A 16-bit data plus four handshake control lines parallel input/output port is provided. This parallel port consists of a fully-buffered PIA device with a pinout that is compatible with a standard Centronics printer type interface. An optional adapter kit is available to interconnect this port to the industry standard optically-isolated solid-state relay mounting racks.

A triple 16-bit programmable counter/timer device is included, with each section's clock, gate and output signals available to the user. The output signals can be strapped to generate a system IRQ, FIRQ or NMI if required.

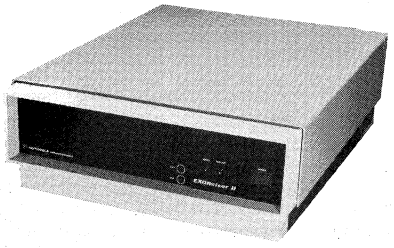
Add-On Input/Output Flexibility

The EXORset Controller has a four-slot card cage with bus connectors for installing additional EXORbus compatible modules available from Motorola as well as a number of other vendors. The Floppy Disk Controller Module occupies one of these four slots.

Development Systems

EXORciser For 8-Bit Prototype Development

M6800 EXOR
M6809 EXOR



The EXORciser is an expandable development system that allows development of any 8-bit Motorola microprocessor or microcomputer configuration, from the simplest to the most elaborate. It comes with an MPU Module that provides system timing and a DEbug Module that contains system firmware. Both MC68B00 or MC68B09 MPU versions are offered in the EXORciser Development System.

With optional accessories, the EXORciser design and diagnostic functions can be extended to other members of the Motorola family of microprocessors and microcomputers.

The EXORciser with a USE (User System Emulator) option can be used to test and evaluate equipment external to its chassis. By removing the microprocessing unit from the user's system and connecting the USE cable from the EXORciser into the MPU's socket, the EXORciser with its EXbug firmware can be used to debug and troubleshoot microprocessor systems.

The basic EXORciser consists of a rugged cabinet with a built-in power supply, and a prewired bus-oriented 14-slot Backplane with MPU and DEbug Modules. Together these elements form a development microcomputer, with the capability of adapting the unit to a specific design problem by adding optional I/O and memory modules. Additional Motorola memory modules for the EXORciser can be selected to suit varying system configurations; for example, to meet the increased memory requirements of sophisticated high order language based systems. The concept of add-on modules permits the user to match the functional requirements of the systems being developed. Using one slot each for a floppy disk and printer function, ten slots remain for memory and I/O expansion. The EXORciser is a system that is never out-of-date, being at all times upgradable when new and expanded microcomputer functions become available.

Accessories for EXORciser

PROM Programmer — M68PP5

The PROM Programmer is designed to program a variety of MOS PROMs, EPROMs and bipolar PROMs. It can verify data from the PROM, transfer data from the PROM to the development system RAM memory, and transfer blocks of data from one memory location to another. Programming time depends upon the PROM used.

The M68PP5 is a powerful new EEPROM/PROM/EPROM programmer, designed to provide all of the functionality of the M68PP3 and more. A powerful feature of the M68PP5 is that it does not require removal of the EXORset or EXORciser covers during operation. This is accomplished via the Remote Socket Module. This module can be conveniently positioned by the user for his needs. It also has many other new or improved features. Such features include: programming even or odd byte PROMs/EPROMs, commands to display and modify data, attach printer command to send all responses to a hard copy printer. The power of the M68PP5 is further enhanced by the increased list of standard devices which it can program.

Software on diskette for both M6800 and M6809-based systems is included with the PROM Programmer.

System Analyzer — MEX68SA2 (6800) / M6809SA (6809)

This unique instrument can be used to enhance the capabilities of the EXORciser as a design tool, or as an independent, portable, low-cost unit for field service of bus-compatible equipment.

In field service applications, the System Analyzer derives operating power and I/O signals directly from the system under test. It can stop the system at any point in its program, step through the program, change the contents of the system memory, and monitor and record the MPU's operation during a selected portion of the program without shutting down the operation.

In EXORciser applications, it complements the system's inherent program development capabilities. In conjunction with the EXORciser and USE, it offers a powerful combination of development and diagnostic tools available for microcomputer work.

MC6801 Development System — MEX6801

This product upgrades EXORciser and EXORset for development of MC6801-based systems. All three modes of MC6801 operation — single-chip, expanded multiplexed and expanded non-multiplexed — are supported by this system.

This system is fully compatible with all current supporting hardware and software and includes the USE function. It allows real-time emulation of the MC6801 application hardware and facilitates the debugging of software.

Resident System Software

8-Bit Assemblers, Editors and Monitors

M6800 and M6809 Development System Software Package

Supplied with the Motorola floppy disk subsystem, EXORdisk, is a basic software development package consisting of the Motorola Disk Operating System (MDOS), CRT Editor, Macro Assembler and Linking Loader.

M6800/01/09 UP/Down Load **M6800UPDWNLD, M6809UPDWNLD**

Permits a user to download software developed in a host system into an EXORciser or Micromodule; alternatively, memory-to-memory uploads are permitted between EXORcisers, and a memory-to-file upload to an EXORmacs; provided in both a 2K PROM and two 1K PROMS.

8-Bit High Level Languages and Cross Assemblers

M6800/M6809 MPL Compiler **M68MPLR020M/ M6809MPL**

A high-level, user-oriented system programming language for the MC6800 and MC6809 MPU's, MPL is a block-structured language with features chosen for applicability to the microprocessor environment. This compiler is designed to operate in an EXORterm or EXORciser floppy disk-based environment with MDOS.

M6800/M6809 FORTRAN **M68FTNR012M/ M6809FORTRN**

Resident FORTRAN is a high-level programming language widely used for scientific and engineering problem solving. This FORTRAN Compiler, which is a subset of the ANSI standard FORTRAN IV, translates the source program into a relocatable object module. The Linking Loader converts the relocatable object code into an executable object file.

M6800 Real-Time FORTRAN Compiler **M68RTFR02M**

This FORTRAN Compiler enables the user to write real-time software in a high-level language for use in M6800-based Micromodule systems. It also contains an execution-time operating system, allowing several queues of tasks to be performed, with an ability to respond to real-time interrupts and to generate delays.

M6800 Resident BASIC Interpreter **M68BASR010M**

The Resident BASIC Interpreter provides another problem-solving tool to the M6800 microcomputer family of products. BASIC is high-level programming language widely-used for education, general-purpose, and certain business-related applications. Decimal arithmetic, string variables and arrays, string functions, and printer output are several of the features.

M6809 BASIC-M Interactive Compiler **M6809BASMR**

The BASIC-M Interactive Compiler provides an extension over standard BASIC in two major directions. It improves considerably the capabilities of the BASIC programming language and generates executable codes that can be used independently of the compiler itself. The compiler is available for M6809-based EXORciser.

M6809 Pascal **M6809PASCLC**

M6809 Pascal produces relocatable object modules that may be linked with other separately compiled Pascal modules and/or assembly language modules. The object code is position independent, re-entrant and ROMable. Both a Compiler which produces a relocatable object module, and an Interpreter version are available for M6809-based EXORciser systems.

EXORciser CRT Editor **M68EDITM**

EDITM is a memory resident record key oriented text editor that can do CRT editing on a line-by-line basis using only the left and right cursor functions and a few easy to remember control key sequences. EDITM can edit up thru a 132 character line, can be run under CHAIN command control, has error recovery procedures, and is USER configurable for different CRTs and default conditions. 6800 and 6809 versions of EDITM are included. 32K RAM minimum.

EXORset/EXORciser Document Processor **M68XDOC/ M68MDOC**

DOC is a powerful text processing program. Any editor may be used to imbed the DOC processor commands with DOC interprets and formats in the output text. Among the many features are: file concatenation (book chapters), multiple file input (form letter/address file), automatic table of contents generation, automatic page numbering, left/right/center text justification, conditional text, and multiple line spacing. 24K RAM minimum.

Peripherals

Use of appropriate peripheral devices can generate savings by affording faster program development. Each Motorola peripheral is supplied with the necessary circuitry to perform the necessary development system interface function.

EXORterm 155 — M68SXD10155A

EXORterm 155 is a video terminal which facilitates the exchange of data between the user and the development system via a high quality video interface in combination with keyboard entry and a serial communications link using speeds up to 9600 baud.

EXORterm 155 uses LSI components of the M6800 family to provide control of the display attributes, communication facility, terminal switch/indicator control, and keyboard inputs. The keyboard provides cursor control keys and special keys to invoke functions unique to the EXORciser and EXORmacs Operating Systems. These keys can also be used by the designer for special routines. An additional Text Edit mode feature permits multiuser editing.

EXORterm 155 may be connected for either RS-232C or 20/60 mA current loop operation. Like the EXORterm Development System, this display console contains a high-quality CRT with a full 1920-character screen and 7 x 9 ASCII characters.

The EXORdisk — M68DSK2, M68DSK3, M68KFD1102

The EXORdisk is a dual floppy disk storage system with its own package of development software. EXORdisk is designed to support either MDOS (the EXORciser Disk Operating System) or VERSAdos (the EXORmacs Disk Operating System). It facilitates high-speed data transfers through fast headsettling time and logical sector arrangement. An interface card connects this storage system to the EXORmacs, EXORciser or EXORterm Development Systems.

EXORdisk is available in various storage capacities. EXORdisk II offers 512K bytes of storage. It is a single-sided/single-density dual drive system with up to 256K bytes of memory per diskette. EXORdisk III is a double-sided/single-density dual drive system with total storage of 1 million bytes. An expansion unit is available for EXORdisk III which adds one additional disk drive and interconnecting cable to increase storage to 2 million bytes.

Hard Disk — M68KHDS16-1, M68KHDS32-1, M68KHDS50-1, M68KHDS96-1

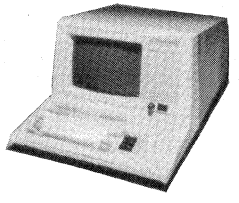
The longer, more complex programs written for advanced 16-bit processors like the MC68000 make much higher data transfer speed and larger storage capacity a necessity. New Hard Disk systems offer the EXORmacs user a choice of high speed mass storage.

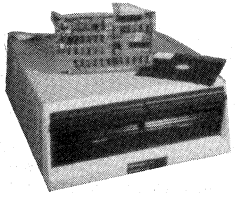
For multiuser operation in the EXORmacs system, Hard Disk is required to provide rapid storage and retrieval for a large number of files. Hard Disk storage greatly enhances and increases file access performance over a floppy disk-based system.

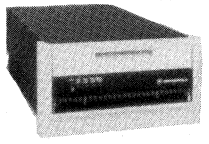
Dot-Matrix Printers — M68SP702C10, MPRINT703

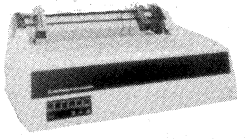
Motorola dot-matrix printers are equipped with an interface module and/or an interconnection cable assembly that specifically adapts them to the various Motorola microcomputer development systems, including the EXORmacs, EXORciser, EXORterm and EXORset. In addition these interface accessories permit the printers to be used with Motorola Micromodules to provide more complete availability of microcomputer system components. Printer specifications are as follows:

FUNCTION	703	702
Print Speed (cps)	180	120
Lines-per-Minute (80 characters)	90	65
Bidirectional Printing	Yes	Yes
Dot-Matrix	7 x 7	7 x 7
ASCII Character Set	96	64
Tractor Feed	Yes	Yes
Condensed Print (10–16.5 cpi)	—	—









8

REFERENCE GUIDE: Selection by MPU/MCU Supported

PRODUCT CATEGORY: EXORmacs (68000 only)

Type Number	Description
M68KVM10-3	VERSAbus RAM 128K Bytes
M68KVM11-1	VERSAbus RAM 256K Bytes
M68KVM11-2	VERSAbus RAM 512K Bytes
M68KHDD16-1	16 MB Hard Disk
M68KHDD32-1	32 MB Hard Disk
M68KHDD50-1	50 MB Hard Disk
M68KHDD96-1	96 MB Hard Disk
M68KMCCM	Multichannel Communications Module
M68KEXTM	VERSAbus Extender Module
M68KFD1102	EXORdisk III for EXORmacs
M68KVAM	VERSAbus Adapter Module
M68KWW	VERSAbus Wirewrap Module
M68K703LP1	EXORmacs Printer 703, 110 V
M68KRDS1	EXORmacs Remote Development Station with USE
M68KRDS2	EXORmacs Remote Development Station without USE
M68KMACSRK	EXORmacs Rack Mount Kit

PRODUCT CATEGORY: Systems Products

Type Number	6800	6801	6809	6805	68000	68010	Description
M68K101-1						X	VME/10 Microcomputer System — 5 MB
M68K102B1						X	VME/10 Microcomputer System — 15 MB
MVMC682-114					X		VMC 68/2 Microcomputer System
M6809SET110	X	X	X	X			EXORset Microcomputer System

PRODUCT CATEGORY: Instrumentation

Type Number	68000	68008	68010	68120	6804	6805	146805	68705	6800 Family	Description
M68KHDS400	X	X	X							HDS-400 Control Station
M68KHDS16FB	X	X	X							HDS-400 Personality Module
M68000HDS4	X									MC68000 Emulator Module
M68008HDS4-8		X								MC68008 Emulator Module
M68010HDS4-8			X							MC68010 Emulator Module
M68HDS201					X	X	X	X		HDS-200 Control Station
M6804P2HM					X					MC6804P2 Emulator Module
M6805P234HM						X		X		MC6805P2,P4,P6, MC68705P3,P5 Emulator Module
MC6805RU234HM						X		X		MC6805R2,R3,U2,U3, MC68705R3,U3 Emulator Module
M6805S2HM						X				MC6805S2 Emulator Module
M6805T2HM						X				MC6805T2 Emulator Module
M146805E2HM							X			MC146805E2 Emulator Module
M146805F2HM							X			MC146805F2, M1468705F2 Emulator Module
M146805G2HM						X		X		MC146805G2, M1468705G2 Emulator Module
M68BSAC	X									Bus State Analyzer Control Module
M68BSA1-1	X									MC68000 BSA Personality module
M68BSA2									X	M6800 Family Personality Module
M68BSA4				X					X	MC6801, MC68121 Personality Module
M68BSA5	X	X	X							VERSAbus Personality Module
M68BSA6									X	EXORbus Personality Module
MEX6801EVM									X	MC6801 Evaluation Module
M68120EVM				X						MC68120 Evaluation Module
M68705EVM							X			MC68705 Evaluation Module
M1468705EVM							X			MC1468705 Evaluation Module

8

PRODUCT CATEGORY: EXORciser

Type Number	141000	68000	6801	6809	6805	6800/02	Description
MEX68IC2				X	X	X	I/O Interconnect Cable (Use with MEX6821-2)
MEX68RK2		X		X	X	X	Rack Mounting Kit EXORciser I & II
MEX68RR		X		X	X	X	EPROM/RAM Module
MEX68SA						X	System Analyzer
MEX68SA2						X	System Analyzer II
MEX68USEC						X	User System Evaluator
MEX68USM				X		X	Universal Support Module
MEX68WW				X	X	X	Wirewrap Module
MEX68XT3				X	X	X	Extender Module
MEX6801EVM			X				Evaluation Module
MEX6801EVM1			X				68701 Programming Module
MEX6801			X				Development System
MEX6802-46						X	MC6802/46 Support Module
MEX6808-22		X	X	X	X	X	8K Static RAM Module with Parity
MEX6809KT				X			6809 Upgrade for EXORciser or EXORterm
MEX6812-1		X	X	X	X	X	2K Static RAM Module
MEX6816-1HR		X	X	X	X	X	16K Dynamic RAM Module with Hidden Refresh
MEX6816-22D		X	X	X	X	X	16K Dynamic RAM Module with Parity
MEX6816-22S		X	X	X	X	X	16K Static RAM Module with Parity
MEX6821-2		X	X	X	X	X	Input/Output II Module
MEX6832-1HR		X	X	X	X	X	32K Dynamic RAM Module with Hidden Refresh
MEX6832-22		X	X	X	X	X	32K Dynamic RAM Module with Parity
MEX6845		X	X	X	X	X	MC6845 CRT Controller Module
MEX6848-1HR		X	X	X	X	X	48K Dynamic RAM Module with Hidden Refresh
MEX6848-22		X	X	X	X	X	48K Dynamic RAM Module with Parity
MEX6850		X	X	X	X	X	ACIA Module
MEX6850-2		X	X	X	X	X	ACIA/SSDA Module
MEX6854		X	X	X	X	X	MC6854 ADLC Support Module
MEX6864-1HR		X	X	X	X	X	64K Dynamic RAM Module with Hidden Refresh
MEX6864-22		X	X	X	X	X	64 Dynamic RAM Memory with Parity
MEX68488		X	X	X	X	X	MC68488 GPIA Support Module
M68BASR010M						X	Resident BASIC Interpreter on 6800 MDOS Diskette
M68FTNR012M						X	Resident FORTRAN Compiler and Linking Loader on 6800 MDOS Diskette
M68MPLR020M						X	Resident MPL Compiler on 6800 MDOS Diskette
M68PANEL220		X	X	X	X	X	6809 Front Panel Conv. of EXORterm 200
M68PP3		X	X	X		X	PROM Programmer III
M68PP3-1		X	X	X		X	Personality Module & Software for PPIII to allow Programming of MCM2532 and MCM68764
M68RTFR02M						X	Resident Real-Time FORTRAN Compiler on MDOS Diskette for 6800
M6800DOWNLD			X			X	6800/6801 Down-Line-Load ROM
M6800EXOR						X	M6800 EXORciser II Development
M6800EXORU						X	M6800 EXORciser II USE Development System 110 V
M6800SMDOS						X	6800 CRT Editor/Macro Assembler with MDOS
M6800XASMBL1			X			X	6800/6801 Cross Macro Assembler
M6805MASC01M					X		6805 Cross Macro Assembler and Linking Loader on MDOS Diskette
M6809DOWNLD				X			6809 Down-Line-Load ROM
M6809EXOR				X			M6809 EXORciser II Development System 110 V
M6809FORTRN				X			6809 Resident FORTRAN Compiler
M6809MASC01M				X			6809 Cross Macro Assembler and Linking Loader on MDOS Diskette
M6809MPL				X			6809 Resident MPL Compiler on MDOS Diskette
M6809PASCLC				X			6809 Resident PASCAL Compiler
M6809SA				X			System Analyzer II
M6809SMDOS				X			6809 CRT Editor/Macro Assembler with MDOS
M6809USE				X			User System Evaluator
M6833		X	X	X	X	X	Blank Diskettes (SS/SD)
M6834		X	X	X	X	X	Blank Diskette (DS/SD)

PRODUCT CATEGORY: PERIPHERALS

Type Number	68000	6809	6805	6802	6800	Description
M68DSK2		X	X	X	X	EXORdisk II 110 V
M68DSK3		X	X	X	X	EXORdisk III 110 V
M68SFDRK3		X	X	X	X	Rack Mounting Kit, EXORdisk II and III
M68SFDU1102E		X	X	X	X	EXORdisk IIIE Expansion Unit, 110 V
M68SP702C10		X	X	X	X	Microsystems Printer 702, 110 V
MPRINT703	X	X	X	X	X	Microsystems Printer 703, 110 V
M68SXD10155A	X	X	X	X	X	EXORterm 155
M68KHDS32-1	X					32MB Hard Disk
M68KHDS96-1	X					96MB Hard Disk
M68KHDE32-1	X					32MB Hard Disk Expansion
M68KHDE96-1	X					96MB Hard Disk Expansion
M68CART	X					Hard Disk Cartridge

PRODUCT CATEGORY: VMEmodules (68000 family)

Type Number	Description
MVME101	68000 Monoboard Microcomputer
MVME110	68000 Monoboard Microcomputer with I/O Channel Interface
MVME200/201	64K and 256K Byte Dynamic Memory
MVME210	Static RAM/ROM Board
MVME300/310	GPIB Controller Modules
MVME310	Universal Intelligent Peripheral Controller
MVME315	Intelligent DMA SASI Interface and Floppy Disk Controller
MVME930	VMEbus Extender Board
MVME931	VMEbus Wirewrap Board

PRODUCT CATEGORY: VERSAmodules (68000 family)

Type Number	Description
M68K0RMS68K	M68000 Real-Time Multitasking, Software (Object) on EXORmacs Diskette
M68KVM01A1	68000 16-Bit Monoboard Microcomputer, 32K RAM
M68KVM01A2	68000 16-Bit Monoboard Microcomputer, 64K RAM
M68KVM02	68000 16-Bit Monoboard Microcomputer, 128K RAM
M68KVM03	68010 16-Bit Monoboard Microcomputer, 10 MHz, 256K RAM
M68KVMCC1	4-Slot Card Cage
M68KVMCH1-1	VERSAmodule System Chassis, 15 Amps-5 Vdc, 110 V
M68KVM10-3	128K Byte Dynamic RAM Module
M68KVM11	256/512K Byte Dynamic RAM
M68KVM20	Floppy Disk Controller Module
M68KVM21	Universal Disk Controller
M68KVM30	4-Channel Serial Communication Module
M68KVM60	Universal Intelligent Peripheral Controller Module
M68KVBUG	VERSAbug Debug Monitor Firmware Package

PRODUCT CATEGORY: MICROMODULES

Type Number	6809	6802	6800	Description
M68BASRC1			X	Resident BASIC Interpreter ROM Set (MINIBUG II-Based)
M68BASRC2			X	Resident BASIC Interpreter ROM SEt (MICRObug-Based)
M68BASRM2			X	Resident BASIC Interpreter Module (Micromodules)
M68EAB1			X	Resident Editor/Assembler and BASIC Interpreter Module (MINIBUG II-Based)
M68EAB2			X	Resident Editor/Assembler and BASIC Interpreter Module (Micromodules)
M68MMCC05	X	X	X	Card Cage, 5-Card
M68MMCC10	X	X	X	Card Cage, 10-Card
M68MMFLC1	X	X	X	Front Load Chassis, 14 Card, 110 V
M68MMFLK	X	X	X	Rack Mounting Slide Kit, FLC
M68MMLC1	X	X	X	Long Chassis, 10-Card, 110 V
M68MMLK	X	X	X	Rack Mounting Kit, Long Chassis
M68MMPS1-1	X	X	X	Micromodule, EXORciser, EXORterm, DC Power Supply, 110 V
M68MM01A2			X	Monoboard Microcomputer (with four 2K x 8 EPROM/ROM Sockets)
M68MM01B1A		X		Monoboard Microcomputer
M68MM01D			X	Monoboard Microcomputer
M68MM03	X	X	X	32/32 Input/Output Module
M68MM03-1	X	X	X	32/32 Input/Output Module (with 4.7K Termination Option)
M68MM03-2	X	X	X	32/32 Input/Output Module (with 330/220 Termination Option)
M68MM04A	X	X	X	16 Socket EPROM, ROM or RAM Module
M68MM05A	X	X	X	8-Channel, 12-Bit Differential Input A/D Module
M68MM05B	X	X	X	16-Channel, 12-Bit Single Ended Input A/D Module
M68MM05C	X	X	X	Quad 12-Bit D/A Module
M68MM07	X	X	X	Quad Communication Module
M68MM08A			X	MICRObug Module-Consisting of MICRObug ROM (Use with MM01A2)
M68MM09	X	X	X	4K CMOS RAM with Battery Backup
M68MM10B	X	X	X	Power Fail Detect Module with Battery Backed-up CMOS Time-of-Day Clock/Calendar
M68MM11	X	X	X	RS-232C to TTY Adapter Module
M68MM12		X	X	GPIB Listener/Talker/Controller Module (with 6800 Firmware)
M68MM12-1	X			GPIB Listener/Talker/Controller Module (with 6809 Firmware)
M68MM12A	X	X	X	GPIB Listener/Talker Module
M68MM13A	X	X	X	Digital-Output (Contact Closure) Module — 16 Outputs
M68MM13B	X	X	X	Digital-Output (Contact Closures) Modules — 32 Outputs
M68MM13C	X	X	X	Optically Isolated Digital Input Module-24 Voltage Inputs
M68MM13D	X	X	X	Optically Isolated Digital Input Module-24 Contact Closure Inputs
M68MM14	X	X	X	2 MHz Hardware Arithmetic Processor Unit
M68MM14A	X	X	X	3 MHz Hardware Arithmetic Processor Unit
M68MM15A	X	X	X	High-Level A/D Module 16 Channel
M68MM15A1	X	X	X	High-Level A/D Module 32 Channel
M68MM15B	X	X	X	Low-Level A/D Module
M68MM15CV4	X	X	X	High-Level Voltage D/A Module 4 Channel
M68MM15CI4	X	X	X	Current D/A Module 4 Channel
M68MM16	X	X	X	Combo ROM, RAM and I/O (Parallel and Serial) (1 or 2 MHz)
M68MM17	X			6809 Monoboard Microcomputer
M68MM19A	X			6809 Monoboard Microcomputer (2 MHz) (For new designs use MM19A1, up to 32K EPROM)
M68MM19SB	X			SUPERbug Firmware ROM
M68MMI/OC	X	X	X	Parallel I/O Adapter Set
M68XEARC1		X	X	Resident Editor/Assembler ROM Set (MINIbug II/MICRObug-Based)

VMEmodules

VMEmodules from Motorola incorporate the high performance MC68000, the internationally accepted Eurocard format, the defacto industry standard 16-bit VMEbus, and the new and flexible I/O Channel, all combined in the most versatile and latest state-of-the-art approach to the modular systems concept.

The MC68000 MPU

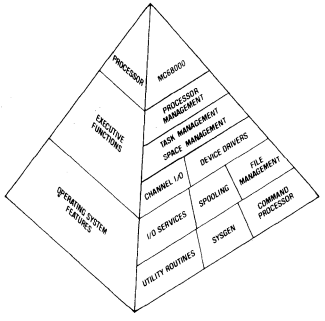
You've seen the benchmarks, and the results — MC68000 has emerged as the acknowledged microprocessor leader in the 16/32 bit performance class. Its architecture is designed for optimal support of the latest high-level languages, and it directly addresses 16 Megabytes of memory (instead of one Megabyte for most of the competition). Its 32-bit internal features mean easy growth to full 32-bit capability as your needs grow into the future. VMEmodule products put the MC68000 MPU to work in a modular structure that has achieved worldwide acceptance and support, both by users and manufacturers of microcomputer subsystems.

Worldwide Standard Package: Eurocard

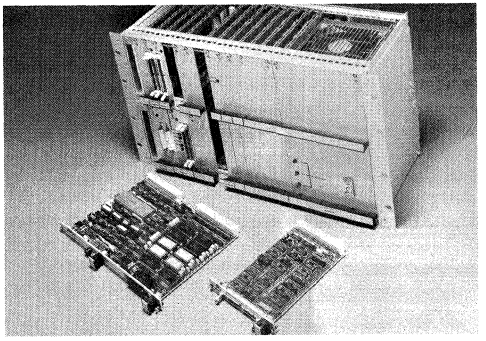
Developed as a de facto standard in Europe, the Eurocard mechanical format is rapidly gaining worldwide acceptance of modular applications in a broad range of laboratory and industrial automation environments. And for good reason — the Single and Double Eurocard circuit boards and card cages in the VMEmodule product line offer a convenient size, plus pin-and-socket bus connectors to give you an extra margin of confidence of reliability in the more severe application environments.

Multiprocessing 16/32 Bit VME Bus

The VME bus doesn't lock you into today's technology. It has the inherent power and capabilities to adapt to any number or types of popular processors for true multiprocessing applications; and, you can use as many bus masters as you need.



BASIC-M, I/Omodule, RMS68K, VMEbus, VERSAbug, VERSAdos, VERSAmodule and VMEmodule are trademarks of Motorola Inc.



With the VME bus, you can mix 8, 16, and 32-bit processors in the VME backplane. It operates asynchronously at high speed, and provides 7 interrupt plus 4 bus arbitration priority levels to allow total flexibility.

I/O Channel Expands Capabilities

The VMEmodule system architecture supports the I/O Channel feature described elsewhere in this publication. Briefly, the I/O Channel is a buffered extension of the on-board processor bus, allowing the system to be easily custom-tailored with the addition of input/output functions in small modular amounts both within and external to the VME card cage. The I/O Channel promotes efficient system utilization by allowing I/O transfers to proceed at rates up to 2 megabytes per second, independently of other on-going activity in the higher-speed VMEbus system interconnect.

Powered by High Performance Software

VMEmodule products are designed for demanding lab and industrial automation environments where quick, accurate response to multiple random events is essential — and Motorola's RMS68K Real-Time Multitasking Executive software for the VMEmodule Monoboard Microcomputer provides the nucleus around which complete real-time applications can be built. For those applications where large data files and mass storage resources must be handled efficiently, there's the full-featured VERSAdos Operating System. Standard device drivers are provided with both VERSAdos and RMS68K for interfaces and devices supplied by Motorola, and both systems make provisions for easy addition of user-supplied device drivers. Both the RMS68K Executive and the full VERSAdos System are rapidly emerging as the standard real-time system structure for MC68000-based applications.

To provide diversified programming capabilities for VMEmodule-based projects, Motorola supplies not only an advanced Structured Macro Assembler, but also efficient Pascal and FORTRAN Compilers.

VMEmodules

And to offer streamlined debugging capabilities, the VMEbug Debug Monitor firmware is available either in ROM or on disk for use with the VMEmodule Monoboard Microcomputer.

Modular Subsystems elevate the starting point for microcomputer system design from the "components" level to the board level. And, just as there are variations in microprocessors for different end-use requirements, there are families of modular subsystems to best serve these varying demands. Thus, the VMEmodule family joins the existing Motorola Micromodule 8-bit family and the VERSAmodule 16-bit family of modular microsystem products to let the user tailor his system to his specific needs.

VMEmodules provide a degree of performance and flexibility that bridges the gap between the lower-level 8-bit processing tasks (the Micromodule domain) and the highend computation and memory-intensive challenges that are the domain of the physically larger and more complex 16/32-bit VERSAmodules. This spectrum of microsystem products offers the most cost effective solution to complex systems — perhaps distributed control systems — with the right performance elements at each processing node of the system.

The Intangible Extras —

When you select Motorola microsystem products for your system design, you get not just the hardware and software, but a host of built-in benefits of almost equal importance. Among these:

- A field-proven line of thoroughly tested products that assure highly reliable system operation.
- A time-tested set of support tools and documentation that simplify system design and operation.
- A nationwide field-sales and service network that offers design and applications support before, during and after the sale.

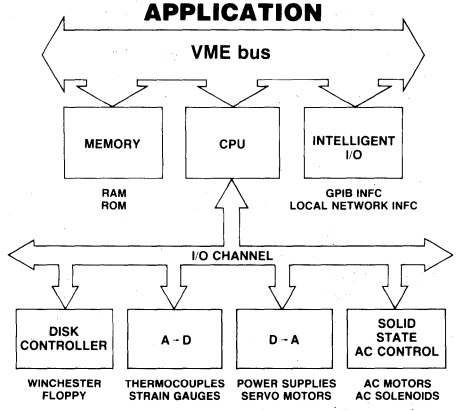
A mature training program at various levels that offers group training at specified locations as well as in the customer's own establishment.

A product line that continues to expand to take full advantage of new developments for increasing capabilities, improving performance and allowing more efficient operation.

Multiple Sources of VME Compatible Products — Worldwide

Development of the VME bus structure represents the combined technical efforts of Motorola and a number of other major international electronics companies. The initial announcement in Europe met with very positive reactions from potential users and vendors the world over, with the result that the original participants are being joined by increasig numbers of companies planning to supply such products. These sources are united through the activities of the VME Bus Manufacturers Group, which meets four times a year in technical forum to help assure the user community of a high degree of technical compatibility between products, and to make available to the public a comprehensive list of suppliers.

TYPICAL VMEmodule APPLICATION



VMEmodule Line*

VMEmodules — VMEbus compatible, Double Eurocard Format.

MVME101 — MC68000 Monoboard Microcomputer with two serial ports and one parallel port on board.

MVME110 — MC68000 Monoboard Microcomputer with I/O Channel support for extended I/O functions.

MVME200/201 — 64K byte and 256K byte Dynamic RAM Modules with data parity check.

MVME210 — Static RAM/ROM Board providing up to 128K bytes storage capacity.

MVME300 — GPIB Controller meeting full IEEE 488-1978 standard.

MVME310 — Universal Intelligent Peripheral Controller with 35% of board area in wirewrap for customer applications.

MVME315 — Intelligent DMA SASI interface and floppy disk controller.

MVME930 — VMEbus Extender Board

MVME931 — VMEbus Wirewrap Board

*See also the list of I/O modules on another page in this catalog for additional I/O functional elements supporting the VMEmodule line.

Software

MVMEBUG — Debugging Packages for VMEmodule Monoboard Microcomputer with single-line Assembler/Disassembler.

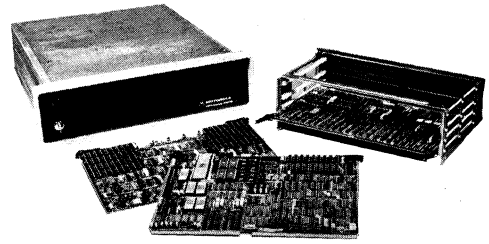
M68KORMS68K — M68000 Real-Time Multitasking Executive provides task scheduling and synchronization for any number of tasks.

M68KOVDOS — OEM VERSAdos Operating System is a real-time multitasking MC68000 based system oriented to hard disk operation.

Packaging

VMEmodule and I/Omodule Card Cages, Chassis, Power Supplies and Backplanes.

VERSAmodules



VERSAmodule circuit boards are microcomputer building blocks from Motorola, based on the state-of-the-art 16-bit MC68000 Microprocessor. They are part of a family of modular building block products that provide the system designer ready-to-run hardware and software. VERSAmodule building blocks drastically reduce the total cost of bringing together a fully configured custom microcomputer-based system . . . by saving development time, engineering talent, and money as well.

With VERSAmodule products, you minimize the risks of design limitations and system obsolescence while keeping your system tied to the leading edge of technology. Your system is built around the most advanced 16-bit microcomputer available today . . . incorporating sophisticated architectural features to enhance system performance. The full range of available software products and applications development tools assure early system completion. And Motorola's experienced support staff is available to help, any time.

Use Today's Most Advanced 16-Bit Microcomputers

The VERSAmodule Monoboard Microcomputers (VM01A and VM02) are the most powerful and versatile 16-bit single-board microcomputers available. They achieve a higher degree of computing power, memory capacity and tailorability by combining the MC68000 MPU with other on-board features. Such on-board features as I/O Channel interface, VERSAbus interface, bus arbitration logic, dual port RAM, multiprotocol serial I/O, parallel I/O, programmable timer/counters, and RAM with battery back-up capability enable these VERSAmodule Monoboards to handle applications ranging from those using a single processor through those requiring complex multiprocessing structures.

VERSAbus Architecture Enhances System Performance

VERSAmodule boards are interconnected in a system using the VERSAbus interconnect standard. The high-speed VERSAbus interconnect is characterized by asynchronous operation supporting direct memory addressing and true multiprocessor operation. Unlike other popular bus structures, VERSAbus architecture does not limit the number or types of processors that can be used in multiprocessing applications. The number of "bus masters" or main processor boards is limited only by the number of card slots in the particular VERSAbus backplane being used. Furthermore, several lines within the VERSAbus structure enhance system reliability and integrity by providing for efficient self-diagnosis . . . resulting in minimum system downtime.

Cost-Effective I/O Channel Increases System Flexibility

The I/O Channel is an advanced architectural feature of VERSAmodule Systems that allows greater system flexibility and low incremental cost for I/O expansion. The I/O Channel has a 12-bit address bus, 8-bit bidirectional data bus, 4K Bytes of memory-mapped I/O, and a data transfer rate of up to 2 Megabytes per second.

VERSAdos Real-Time Disk Operating System

The VERSAdos Operating System Software employs modular design of its major programs to allow easy addition of user functions with minimal cost. It contains a file management package and additional device-independent I/O support. The VERSAdos System is available with software drivers for both floppy and hard disk storage, and incorporates redundant safeguards against system failures. Optimum processor and memory utilization are achieved through true multitasking and dynamic memory allocation/deallocation.

RMS68K Real-Time Multitasking Executive

For real-time applications that do not require auxiliary mass storage (disk), and efficient Real-Time Executive may provide all the required systems functions.

The RMS68K Real-Time Multitasking Executive provides the nucleus around which real-time applications can be built. It allows a wide variety of application systems without large expenditures for complex real-time and multitask control functions. RMS68K is ROMable, meaning that the executable code for your entire system could be placed in ROM. In addition, the RMS68K System customizes your system by allowing you to add your own device drivers and select only those functions that you need. Compatibility with VERSAdos and debug software packages helps reduce the cost of software maintenance over the life of your system.

VERSAbug Debugging Packages

The VERSAbug debug package provides a powerful evaluation and debugging tool for VERSAmodule Systems. It permits full-speed execution of system and user-developed programs operated in a VERSAmodule Monoboard Microcomputer environment under complete operator control.

VERSAbug software is available as a system debug monitor, in a pre-configured EPROM resident package, or as source and relocatable object modules, packaged on diskette or cartridge disk, allowing you to easily create your own application-specific version in a matter of hours. In either package, VERSAbug software gives you a powerful tool for reducing system development and continuing maintenance costs.

VERSAmodules

Complete Your System . . . On Schedule

With VERSAmodule products, the lion's share of your system's hardware design, debug, assembly and test is done for you. The mature operating system software is already developed and debugged, too. You can begin developing your applications software immediately, in order to respond faster to customer requirements, penetrate fast moving market windows, or automate a critical activity sooner. The result . . . higher profitability.

Use Your Resources Efficiently

Since your costly and often limited technical resources are not needed to design or debug the basic computer system hardware, you can concentrate on the value-added areas of applications software and any unique hardware requirements. In other words, you apply your scarce resources to the area you know best . . . your application.

Lower Your Non-Recurring Costs

The rising costs to design, develop and debug basic system hardware are reduced by using VERSAmodule products. But the cost savings don't stop here. The powerful applications development tools supporting the VERSAmodule family greatly facilitate the development and debugging of your applications software and any specialized hardware. This allows you to get it right the first time . . . avoiding costly redesigns and project delays.

VERSAmodule CIRCUIT BOARDS

Monoboard Microcomputers

VM01A Monoboard Microcomputer — MC68000 MPU, 32/64K Byte RAM, Sockets for 64K Byte ROM, four parallel I/O ports, two serial I/O ports.

VM02 Monoboard Microcomputer — MC68000 MPU, 128K Byte dual-port RAM, Sockets for 64K Byte ROM, two Multiprotocol serial I/O ports. I/O Channel Interface.

VM03 Monoboard Microcomputer — MC68010 MPU at 10 MHz, MC68451 MMU, 256K DRAM, Sockets for 64K Byte ROM, two Multiprotocol serial I/O ports and I/O Channel Interface.

Memory Modules

VM10 Random Access Memory — 128K Byte Dynamic RAM, Byte Parity, 16-bit data/word length.

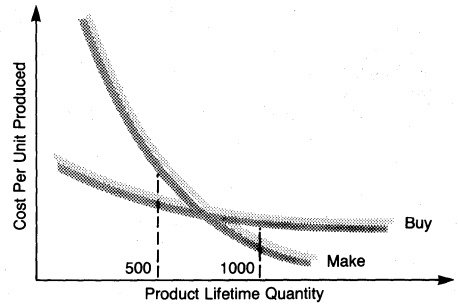
VM11 Random Access Memory — 256/512K Dynamic RAM, Error detection and correction, 16/32-bit data word length.

VM80 Combination ROM/RAM/I-O — 0/128K Byte Dynamic RAM, Sockets for 256K Byte ROM, six parallel I/O ports, two Multiprotocol serial I/O ports.

Controllers

VM20 Floppy Disk Controller — Up to 4 floppy disks, 2M Byte formatted floppy capacity, On-board IPC with data buffer.

VM60 Universal Intelligent Peripheral Controller — 4K Bytes on-board RAM, Up to 32K Bytes ROM, DMA data transfers, Wire wrap area for custom user interface.



VM21 Universal Disk Controller — UP to 4 floppy disks, Up to 2 SMD compatible hard disks, Up to 516 M Byte formatted disk capacity, On-board IPC with data buffer.

VM30 Multi-Channel Communications Module — four RS-232C serial I/O ports, One parallel printer port, ON-board IPC with data buffer.

Support

RSC1 Remote Serial Conversion Module — RS-232C to RS-449 or multidrop port, Synchronous or asynchronous operation, Half or full duplex, Eurocard form factor.

System Packaging and Accessories — 5¼ inch Chassis, Stand-Alone Card Cage, Power Supplies, Cabling Options, I/Omodule Card Cage, Mass Storage Enclosure, Industrial Card Cage System Package, VERSAbus Adapter Module.

Addition I/O

All of the I/Omodules described under I/Omodules in this catalog are compatible with the I/O Channel on VERSAmodule 02, thus extending many additional I/O and control functions to the VERSAmodule product family.

FUTURE VERSAmodule PRODUCTS

Motorola currently offers more than 20 individual hardware and software products in the VERSAmodule and I/Omodule product lines. But beyond these, Motorola engineers are at work planning and designing future products to ensure continual expansion of the VERSAmodule product line. New hardware and software products will incorporate the latest technology in easy-to-use building-block form. Future family members will include higher-performance single board computers, higher-density memory modules, and new intelligent device controllers . . . all of which take advantage of advancements in LSI technology. I/Omodule products will expand the offering of popular industry interfaces and new software will bring advanced tools like applications-oriented languages and multiprocessor capability for the VERSAdos Operating System. Other announcements from Motorola, plus those from independent vendors offering VERSAbus-compatible products, will assure an even broader selection of useful products in the future.

I/Omodules

The I/O Channel is a new system architectural concept supported in Motorola Microsystem products which allows modular I/O expansion on the local processor bus.

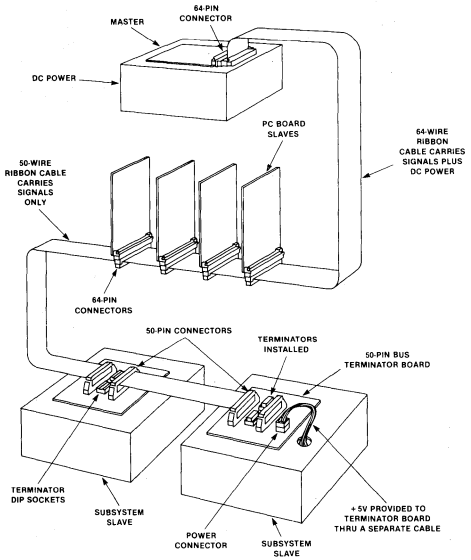
This frees the system bus to handle simultaneous high-speed data exchange and multiprocessor access requirements while permitting most lower speed system I/O activity to take place through the local I/O Channel. Thus, the advanced I/O Channel architecture affords great flexibility in I/O intensive applications such as high speed data acquisition and distributed control.

More than a dozen defined I/Omodule products already support the Motorola modular product line. The family will grow into the future with additional offerings from Motorola, and with a variety of I/O Channel compatible products from other vendors. Should you desire to design custom I/O Channel modules for your specific needs, that task is made easier by a comprehensive I/O Channel Specification Manual available from Motorola. (M68RIO1/D1)

The I/O Channel provides the following features:

- 12-bit address bus
- 8-bit bidirectional data bus
- Asynchronous operation
- Up to 2 megabyte transfer rate
- Four interrupt lines
- Reset line
- 4 MHz free running clock line

The figure below illustrates how a system might be configured using a ribbon cable bus I/O Channel. The bus master is typically a computer, but may also include a DMA controller for transferring blocks of data to or from a slave device at high speed.



I/Omodule Product Line

A. I/Omodules — I/O Channel Compatible, Single Eurocard Format.

MVME400 — Dual Channel RS-232C Serial Port providing two independent, full-duplex serial input/output ports.

MVME410 — Dual Channel 16-bit Parallel Port, four independent 8-bit ports jumper or software configurable as inputs or outputs.

MVME420 — SASI™ Peripheral Adapter provides interface to SA1400 Shugart Associates SASI Bus.

MVME435 — Buffered 9-Track Magnetic Tape Adapter to interface industry standard 800/1600 BPI, ½" Magnetic Tape Formatter.

MVME600 — Analog Input Module with 16 channel single-ended or 8 channel differential operation.

MVME605 — Analog Output Module with 4 independent channels and 12-bit resolution.

MVME610/615/616 — Opto Isolated 120V/240V AC Input/Output modules with eight independent I/O Channels.

MVME620/625 — Opto Isolated 3VDC Input/Output modules with eight channels and isolation to 2500 Volts.

MVME932 — I/O Channel Extender Board.

MVME933 — I/O Channel Wirewrap Board.

MVME935 — I/O Channel Extender Board which converts DIN connector to 50-pin dual row header.

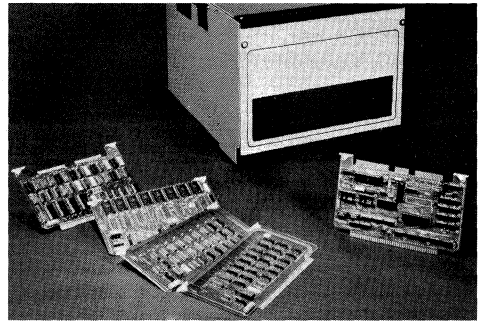
B. I/Omodules — I/O Channel Compatible, Non-Eurocard Format.

M68RWIN1-1, M68RWIN1-2 — Winchester Disk Controller for 5½" or 8" Winchester and Floppy Disk drive combinations.

M68RI01 — Remote Input/Output Module provides parallel I/O oprations and will accept up to 16 compatible solid state relay input and output modules.

M68RAD1 — Remote Intelligent Analog-to-Digital Conversion Module controlled by an on-board Intelligent Peripheral Controller.

Micromodules



The Motorola line of Micromodules offers a selection of modular subsystems that permits a high degree of end-product customization. It is supplemented by a sophisticated library of development software with high-level language interface to simplify man-machine interaction. An array of packaging accessories provides the proper physical environment for the system assembly.

Utilizing Motorola's extensive family of 8-bit MPU-compatible chips, Micromodules are tailored to meet the performance objectives of most industrial automation and data acquisition applications. They are priced to compete favorably with in-house development and manufacturing costs and, in many instances, they represent the most cost-effective means for rapid, reliable system implementation (or even for prototyping chip-implemented systems.)

The Modular Building Blocks

The Micromodule Family is based on a selection of differently configured *single-board microcomputers*. These vary in capabilities and applications as a result of differences in on-board microprocessors, as well as memory and I/O content. For some requirements, a single monoboard microcomputer module, supplemented by a suitable enclosure, a power supply and your applications program, will adequately serve your total needs. For other more demanding applications, the Family offers a wide range of expansion modules which tailor the system to your ultimate requirements.

Software Support

To ease programming load and allow programmers to concentrate on the end product application, incorporate the **M6809 Real-time Multitasking System (RMS09)** as the executive kernal around which a real-time applications system can be built. RMS09 is a flexible collection of systems routines from which the user can customize or 'sysgen' supervisor routines and interrupt handling routines tailored as simple or as complex as the application system requires.

Also available for MC6809-based systems is SUPERbug, a high performance monitor which also provides the facility for linking relocatable modular software routines that can be independently written and executed from EPROM, ROM or RAM. For MC6800-based Micromodules, there is MICRObug Monitor, with system software and hardware debugging capability. Also available are Editor/Assembler and Basic Interpreter packages.

MONOBOARD MICROCOMPUTER SELECTION GUIDE

Part No.	Parallel I/O	Serial I/O			Memory		Options
		RS-232C	RS-422	20mA	ROM Capacity	RAM (Bytes)	
MC6800/MC6802 Based, 1 MHz Clock Rate							
M68MM01B	1 PIA 1 PTM				To 4K**		Low Cost, Self-contained Not Expandable
M68MM01	3 PIAs				To 4K**	1K	
M68MM01A2	2 PIAs	1 ACIA		*	To 8K**	1K	
M68MM01B1A	1 PIA 1 PTM	1 ACIA		*	To 4K**	384	Cassette I/O
M68MM01D	Printer Port 1 PTM	1 ACIA	(Opt) +	*	To 10K**		Use 2K RAMs in ROM Sockets
MC6809-Based; Clock Rate 1 MHz, except M68MM19A1-2MHz							
M68MM17	1 PIA 1 PTM	2 ACIA		*	To 64K**	To 64K**	Use RAMs in ROM Sockets
M68MM19-1 19A1	1 PIA 1 PTM	1 ACIA or SSDA	(Opt) +	*	To 32K**	2K	Replace ACIA With SSDA +

NOTES:
+ = Option requires minor board modifications
* = Option requires addition of Micromodule MM11 (RS-232C to 20-mA Current-Loop Adapter)
** = User supplied

D-A Converters—All inputs are TTL compatible.

Part No.	No. of Channels	Input	Analog Output Range		Comments
			Voltage	Current	
M68MM05C	4	12-Bit Binary	0-5, 0-10 ± 2.5, ± 5 ± 10		Output Voltage Range option is strap selectable.
M68MM15CV	1 to 4*	12-Bit Binary or two's complement	0-5, 0-10, ± 5, ± 10		Input Code and Output Voltage Range Options are strap selectable.
M68MM15CI	1 to 4*	12-Bit Binary or two's complement	0-5, 0-10, ± 5, ± 10	4 to 20mA	Voltage or Current output device with strap selectable current or voltage range options.

*Add suffix 1 through 4 to part number to denote number of channels required.

A-D Converters

Part No.	No. of Channels		Resolution	Input Voltage	Comments
	Diff.	S.E.	No. of Bits	(full scale, dc)	
High Level					
M68MM15A	8	16	12	0-5 Vdc, 0-10 Vdc, ± 5 Vdc, ± 10 Vdc.	V_{in} is strap selectable
M68MM15A1	16	32	12	same as above	
M68MM05A	8		12	± 10mV to ± 10V	
M68MM05B		16	12	same as above	
Low Level					
M68MM15B	1		15 plus sign	± 25 mV, ± 55 mV, ± 80 mV	Expandable to 16 channels with Expander Circuits
M68MM15BEX	1 to 4			Channel Expander for above	

Micromodules

EXORbus-Compatible Memory Modules

	Capacity (K Bytes)	Features
RAM-Static, NMOS		
M68MM06	2	Clock Speed—1 MHz
MEX6808-22	8	
MEX6816-22S	16	With Parity, Clock speed = 1 or 2 MHz (with stretched Phase 2)
RAM-Static, CMOS		
M68MM09	4	Clock speed = 1.5 or 2 MHz (with stretched phase 2) On-board ckt. for user-installed parity.
M68MM21	8	Optional parity.
M68MM21-1	16	Optional parity.
RAM-Dynamic, with parity		
MEX6816-22D	16	Jumper selectable 1-, 1.5-,
MEX6832-22	32	or 2- MHz speed;
MEX6848-22	48	Row-addressable in 16K byte blocks.
MEX6864-22	64	
RAM-Dynamic; with hidden refresh; clock speed = 1 MHz; all with parity.		
MEX6816-1HR	16	
MEX6832-1HR	32	Organized into independently
MEX6848-1HR	48	addressable rows of 16K bytes each.
MEX6864-1HR	64	

Unpopulated Modules—User supplies chip set

	Number Of Sockets	EPROM/ROM CAPACITY (Memory Chip Number)	OPTION RAM CAPACITY (Memory Chip Number)
MEX68RR	20	16K (MCM68708/6830)	512 (MCM6810)
M68MM04	16	16K (MCM68708/6830)	0
M68MM04A	16	64K (1K, 2K or 4K capacity)	8K (1K or 2K capacity)

Serial-Format Digital I/O

ACIA Modules — MEX6850, 50-2

Offers both TTY and RS-232C data terminal interface, with eight switch-selectable baud rates between 110 and 9600 baud. MEX6850 operates at 1MHz and is configured with Modem output; 6850-2, at 2MHz, is configured with 20 mA TTY output.

Quad Serial I/O — M68MM07

Supplied with four MC6850 ACIAs, or with user supplied MC6852 SSDAs for either asynchronous or synchronous operation. Strap options permit RS-232C, RS-422, RS-423 or 20mA interface and baud-rate selection for each of the four ports.

RS-232C to TTY Adapter — M68MM11

Converts RS-232C output to 20 mA TTY operation.

8 Channel Serial I/O Module — M68MM18

Provides eight asynchronous RS-232C channels. Each channel is strap selectable to baud rates from 75 to 115K BPS. Memory location is strap selectable in a block of eight channels.

GPIB Modules

Provide interface between various MPUs and the IEEE STD 488-1978 interface bus. MM12A provides Listener/Talker functions for sending and receiving data bytes, requesting service and responding to parallel and serial polls. MM12 and 12-1 add the controller function that permits the system to send commands and conduct serial and parallel polls.

Listener/Talker for MC6800-type systems **M68MM12A**

Listener/Talker/Controller for MC6800-type systems **M68MM12**

Listener/Talker/Controller for MC6809-type systems **M68MM12-1**

Memory-I/O-Timer Expansion Module — M68MM16-1, 2, 3

Provides functional expansion of Monoboard MM01 (version 16-1), or MM19 (version 16-2), and can be used as MM19 Emulator in an EXORset Development System (version 16-3). Includes asynchronous serial data port with strap-selectable RS-232C, RS-422, or RS-423 interface, parallel interface port with 16 data lines and 2K of static RAM, four control lines, three 16-bit programmable counter/timers, and four sockets for user installed, single 5-volt-supply MOS or bipolar memories.

Parallel-Format Digital I/O

Universal PIA-Controlled I/O — MEX6820, 21-2

Contains two MC6820 Peripheral Interface Adapters (PIA's) for a total of four separate 8-bit I/O ports for peripheral interfacing.

32-In/32-Out Expansion Module — M68MM03

Contains 32 bits of parallel input and 32 bits of parallel output in four continguous 8-bit bytes. Used for simultaneous transfer of 4 bytes of informtion between an MPU and an external system to speed up the data transfer cycle.

16/32-Channel Relay Output — M68MM13A, B

Contains 16 (MM13A) or 32 (MM13B) on-board reed relay output channels to isolate the microcomputer from the system(s) being controlled.

24-Channel Optically Isolated Input Modules — M68MM13C, D

Provides three byte-oriented (8-bit) input channels that have high electrical isolation between microcomputer and equipment being monitored. Input voltages in excess of 17 volts are read as logical "1"; 4 volts or less represent logical "0." MM13D provides an on-board wetting source for applications requiring switch and relay inputs.

Quad Parallel Interface Adapter — M68MM22

Utilizes four PIAs in a versatile buffered I/O configuration that allows up to 64 high-voltage (200 Vdc or 280 Vac) or high-current (to 3A) signals to be monitored or controlled.

Packaging Hardware

Part No.	Description
M68MMCC05	5-Card Cardcage
M68MMCC10	10-Card Cardcage
M68MMFLC1	Front Load Chassis, 14 Card, 110 Vac
M68MMLC1	Long Chassis, 10 Card, 110 Vac
M68MMSC1	Short Chassis, 5-Card, 110 Vac
M68MMPS-1	Power Supply, 110 Vac

MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.