MOTOROLA

8-BIT MICROPROCESSORS DATA MANUAL

1983

This book is intended to provide the design engineer with the technical data needed to completely and successfully design an 8-bit microprocessor or microcomputer based system. The data sheets for Motorola's microprocessor, microcomputer, and peripheral components are included, as well as information about technical training.

A companion manual covering the Motorola family of 16-bit Microprocessors is also available.

Printed in Switzerland -1983

TABLE OF CONTENTS

ALPHANUMERICAL INDEX OF DATA SHEETS

ALPHANUMERICAL INDEX (continued)

Device	Description	Page
MC14442	CMOS LSI Microprocessor-Compatible A/D Converter	4-722
MC68008	Reduced Data Bus MC68000	4-732
MC68488	General Purpose Interface Adapter	4-736
MC68652/2652	8-Bit MOS Multi-Protocal Communications Controller	4-755
MC68653/2653	8-Bit MOS Multi-Protocal Communications Controller	4-776
MC68701	EPROM Microcomputer/Microprocessor	4-797
MC68705P3	8-Bit HMOS 1,8k EPROM Single-Ship Microcomputer	4-836
MC68705P5	(See Note 1)	
MC68705R3	8-Bit HMOS 3,7k EPROM Single-Ship Microcomputer with A/D	4-861
MC68705U3	8-Bit HMOS 3,7k EPROM Single-Ship Microcomputer	4-889
MC146805E2	8-Bit CMOS Microprocessor with RAM and I/O	4-912
MC146805F2	8-Bit CMOS 1k Single-Ship Microcomputer	4-947
MC146805G2	8-Bit CMOS 2k Single-Ship Microcomputer	4-971
MC146818	CMOS Real-Time Clock Plus RAM	4-995
MC146823	CMOS Parallel Interface	4-1015
MC1468705F2	8-Bit CMOS Microcomputer EPROM	4-1026
MC1468705G2	8-Bit EPROM Microcomputer Unit (MCU)	4-1028
MC68HC04P2	8-Bit HCMOS Microcomputer Unit (MCU)	4-1047
MC68HC04P3	8-Bit CMOS Microcomputer EPROM	4-1049
MC68HC05C4	8-Bit CMOS Microcomputer EPROM	4-1050
MC68HC09E	8-Bit HCMOS Microprocessor Unit	4-1053
MCM6810/A10/B10	8-Bit Static Random Access Memory	4-1055
MC68661A/MC2661A	8-Bit MOS	4-1059

Note 1: At the time of print this data sheet was not yet available. This device is similar to MC68705P3 with added security feature on chip.

Motorola's 8-Bit Microprocessor/Microcomputer Families

Motorola's 8-Bit Microprocessor and Microcomputer Families

Serving as the "heart" of every microcomputer system is a microprocessor. Start with the chip set that precisely meets your design objective. Motorola manufacturers the industry's most complete selection of sold-state microcomputer components to give you 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.

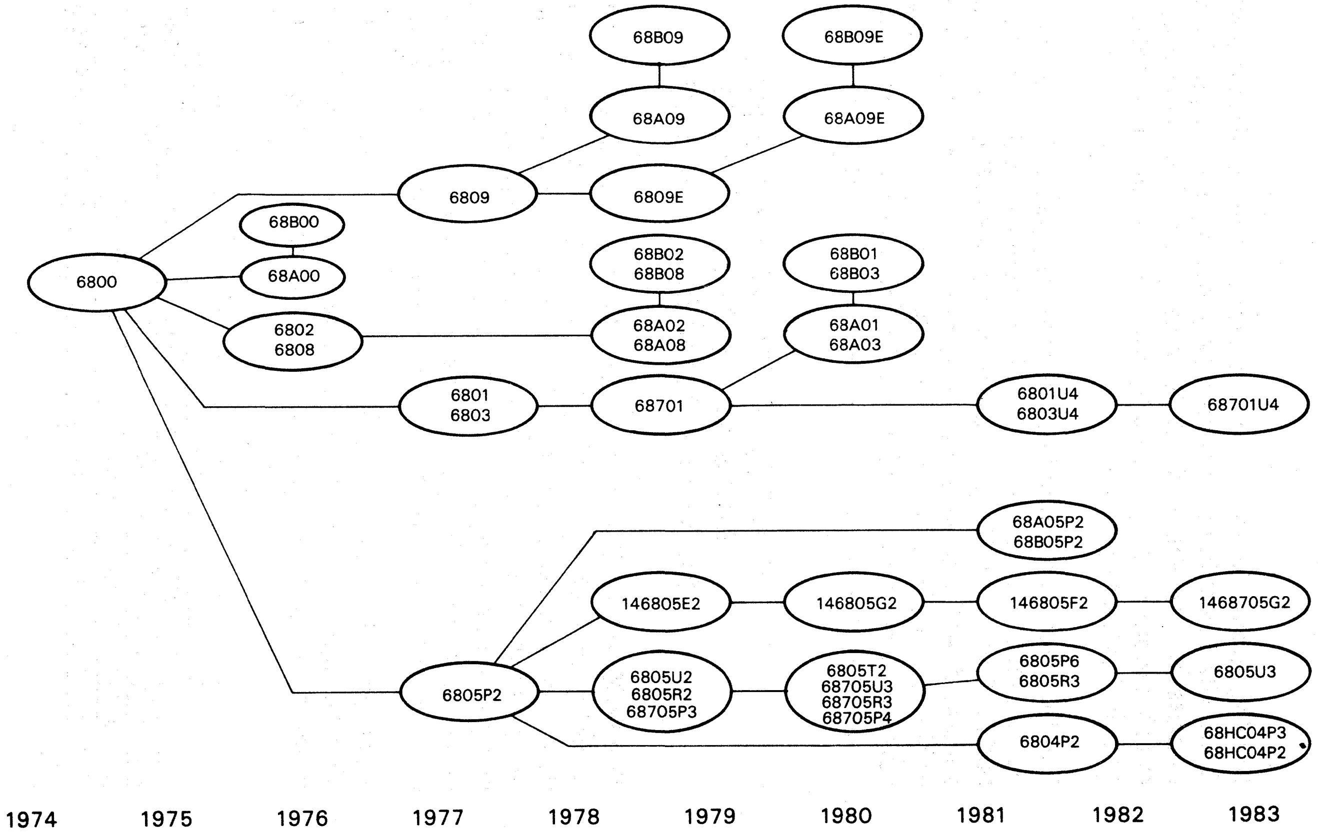
8-BIT MICROPROCESSORS (MPUs)
MC6800 – MC6802 – MC6803
MC6809 – MC6809E – MC146805E2

The MC6800 MPU was the first of the M6800 MPU Family and still remains a highly cost-effective processor for a great many process-control and data-communications applications. Seventy-two instructions and six different addressing modes give it powerful capability, and a full range of compatible peripheral chips offer the widest possible latitude in system implementation. After years of field experience, the MC6800 has earned an enviable reputation as one of the easiest-to-use processors available.

Moreover, to tailor the system to your specific needs at the lowest cost, the MC6800 (and its peripherals) is available in three different packages, three different temperature ranges, and three speed ranges, as follows:

	1.5 MHz			1.5 MHz		2 MHz
	0 to 70°C	−40 to +85°C	−55 to +125°C	0 to 70°C	−40 to 85°C	0 to 70°C
Plastic	MC6800P	MC6800CP	–	MC68A00P	MC68A00CP	MC68B00P
Cerdip	MC6800S	MC6800CS	MC6800BQCS	–	–	–
Ceramic	MC6800L	MC6800CL	–	MC68A00L	MC68A00CL	MC68B00L

FIGURE 1-1. – GENEALOGY OF THE COHESIVE
M6800 MICROPROCESSOR/MICROCOMPUTER 8-BIT FAMILY



The MC6803 MPU is the microprocessor version of the MC6801 single-chip microcomputer. The MC6803 accomodates applications where external ROM is present. With 13 parallel input/output lines, a 16-bit timer, and a serial communications interface the MC6803 offers a great deal of freedom in system needs. One of the most desirable attributes of the multi-generation MC6803 is its compatibility with existing software and hardware. The MC6803 easily meets this goal by being thoroughly integrated into the total M6800 family of components. In addition, since the MC6803 is an HMOS device, it requires only a single +5 volt power supply and interfaces with both TTL and MOS peripherals. The concept of an integrated family of devices is predicated on continuity in both design and development. As a member of the M6800 family, the MC6803 shares many of the attributes of the basic MC6800 MPU. For example, the MC6803 encompasses the full MC6800 instruction set, yet new instructions have been incorporated for even greater system capability and ease of programming. Many MC6803 instructions execute in fewer cycles than on the MC6800. More and faster instructions increase throughput and reduce software conversion and development time. Some of the features of the MC6803 are:

- Expanded MC6800 Instruction Set
- Full Duplex Serial Communications Interface
- Upward MC6800 Source and Object Code Compatibility
- 16-Bit Timer with Three Modes
- 16-Bit Multiplexed Address Bus Providing 64K-Byte Memory Space
- 128 Bytes of On-Chip RAM (64 Bytes Retainable with Battery Backup)
- 13 Parallel I/O Lines
- Internal Clock (Divide-by-Four)
- TTL-Compatible Inputs and Outputs
- Interrupt Capability (Maskable and Non-Maskable)

The MC6809 microprocessor, with five internal 16-bit registers, offers up to five times higher performance than the MC6800, yet, due to the 8-bit bus is fully compatible with all M6800 bus-oriented supplementary circuits and peripherals. Here's how the MC6809 stacks up:

Architectural Improvements:

- Additional 16-Bit Index and Stack Registers
- Direct Page Register
- Increased Addressing Modes

- 16-Bit Operations and 16-Bit Accumulator
- 8 x 8 Multiplier
- Fast Interrupt

Software Improvements:

- Designed for efficient handling of high-level languages, including Pascal, Basic, MPL, Cobol, and Fortran.
- Position-independent coding and reentrant-programming capability encourage development of "canned software," with modular program interchangeability.
- Structural, high subroutined code enhanced by two 16-bit index registers and program counter usable for indexing.
- Multi-task and multi-processor organization.
- Stack-oriented compiler instructions with both user and hardware stack registers available.

Although the MC6809 is compatible with the extensive existing M6800 Family, Motorola is designing even more peripherals to enhance systems designed with the MC6809. These new peripherals (e.g., the MC6829 Memory Management Unit, the MC6839 Floating Point ROM) allow an MC6809 user to realize the full potential of the process.

The MC6809 is a logical step for applications that crowd the capacity limits of today's conventional 8-bit processor — yet, hardware and software (upward) compatibility with existing M6800 processors protects previous software investment.

The MC6809E includes all the features of the MC6809 plus external clocking to provide the flexibility required in a multi-processor system.

The MC146805E2 initiates the CMOS side of Motorola's microprocessor family. Battery-oriented and noise sensitive applications have long sought an M6800 MPU implemented in CMOS. The MC146805E2 includes an 8-bit optimized processor the equal of the MC6800 in speed and performance, plus on-chip RAM, timer, parallel I/O ports, and clock oscillator. Complete CMOS systems are assembled using the MC146823 Parallel Interface, MC146818 Real-Time Clock plus RAM, MCM65516 CMOS 2K ROM, and many MSI and SSI support parts. The MC146805E2 also serves as a ROM-less prototype device for the CMOS and HMOS M6805 Family single-chip MCUs.

The processor has sixty-one basic instructions that are similar to those of the popular MC6800 microprocessor, plus some unique enhancements. A complete set of bit-manipulation and test instructions allow any bit in RAM or any I/O pin to be individually set or cleared or tested as a conditional branch, all with a single instruction. The table look-up indexing modes have also been enhanced and made more ROM efficient.

The very low power requirement of static CMOS make the MC146805E2 family of processors and peripherals extremely attractive for those applications where power is a major consideration (portable instruments, telecommunications, point-of-sale terminals, remote instrumentation, industrial control, appliance controllers, etc.). The operating voltage range is from 3 to 6 volts, while current usage ranges from microamps upward depending upon frequency, voltage, standby modes, and operating duty cycle. Other MC146805E2 features include:

- Expansion Bus Addressing 8K Bytes of Memory
- 112 Bytes of RAM
- 16 Bidirectional I/O Lines in Addition to the Bus
- 2 Program Initiated Low-Power Standby Modes
- Timer/Counter:
  - 8-Bit Programmable Counter
  - 7-Bit Software-Selectable Prescaler
  - External Timer Input
  - Maskable Timer Interrupt
- Maskable External Interrupt
- 40-Pin Package
- Fully Static Operation for Lower Power Needs
- Oscillator Frequency to 5 MHz at 5 V
- Compatible ROM Available — MCM65516 (2K x 8)

8-BIT MICROPROCESSORS FEATURES MATRIX

Device	Tech	Pins	RAM 8X	I/O Lines	Special I/O	Mnem Inst[1]	Ext Addr	Data Size	Clock	Timer
MC6800	NMOS	40	–	–	–	72	64K	8	No	–
MC6802	NMOS	40	128	–	–	72	64K	8	Yes	–
MC6802NS	NMOS	40	128	–	–	72	64K	8	Yes	–
MC6803	HMOS	40	128	13	Serial	82	64K	8	Yes	16-Bit
MC6809	HMOS	40	–	–	–	59	64K[2]	8	Yes	–
MC6809E	HMOS	40	–	–	–	59	64K[2]	8	No	–
MC146805E2	CMOS	40	112	16	–	61	8K	8	Yes	8-Bit + Prescaler

NOTES:

[1] Some Mnemonic Instructions can have many Opcode Instructions. As a result a Microprocessor normally has many more Opcode Instructions than Mnemonic Instructions. For instance the MC6809 has 59 Mnemonic Instructions and 1464 Opcode Instructions.

[2] Two megabytes when used with the MC6829 Memory Management Unit.

8-BIT MICROPROCESSORS SELECTOR GUIDE

TECHNOLOGY

HMOS/NMOS	Page	CMOS	Page
MC6800	4-32	MC146805E2	4-912
MC6802	4-145		
MC6802NS	4-145		
MC6803	4-166		
MC6809	4-327		
MC6809E	4-359		

PROCESSING POWER

8-Bit	Page
MC6800	4-32
MC6802	4-145
MC6802NS	4-145
MC6803	4-166
MC6809	4-327
MC146805E2	4-912

FUNCTIONAL BLOCKS

MPU	Page	MPU with On-Chip RAM	Page
MC6800	4-32	MC6802	4-145
MC6809	4-327	MC6802NS	4-145
MC6809E	4-359	MC6803	4-166
		MC146805E2	4-912

LANGUAGE ORIENTATION

Low-Level Language	Page	High-level Language	Page
MC6800	4-32	MC6809	4-327
MC6802	4-145	MC6809E	4-359
MC6802NS	4-145		
MC6803	4-166		
MC146805E2	4-912		

SINGLE-CHIP MICROCOMPUTERS (MCUs)
THE M6801 – M6805 – M3870 – M6804 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 serially 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 and M6804 Families include a number of memory and package sizes with various special I/O functions, in HMOS, CMOS and HCMOS.

PERFORMANCE – Processor performance, or program efficiency, for the application is an important single-chip MCU selection criteria. The M6801 Familiy is the throughput leader with 16-bit data operations, binary multiply, 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 offers a very successful 8-bit architecture, while the M6804 Family offers 8-bit performance and architecture at a 4-bit price.

TECHNOLOGY – The very high production volumes of high-density NMOS (HMOS) permit low cost single-ship solutions. High density CMOS (HCMOS) 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 and M6804 Families make HCMOS 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.

ROM SIZE – The mask ROM capacities of the present single-chip MCUs range from 1K bytes for the M6805 and M6804 Families up to 4K bytes on one 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. Architectures of the M6804 and MC3870 class offer short one and two byte instructions. The M6801, M6804 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 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 and the M3870 at 64 bytes and 112 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 smaller (and lower cost) 28-pin packages. All four MCU Families include 40-pin versions, while the M6805 and M6804 Families also have 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 other 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 five bus structure modes for off-chip expansion. The five bus modes permit the number of bus pins to be optimized for the amount of address space needed off-chip.

INTERRUPTS – When an application program must synchronize to two or more external events, interrupt hardware in some form is usually necessary. The M6801, M6804 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, may measure external values, may count external events, and may 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 and M6804 Families timers consist of a programmable 8-bit counter and a selectable 7-bit prescaler. The MC3870 timer is 8-bit 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 four families are fully supported on the EXORset, EXORciser and EXORmacs development systems. 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 M6805 Family.

The M6805 and M6804 Families have evolved from the M6800 Family. These families include similar programmable bidirection 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 and M6804 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 reduces 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 M6805 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, 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.

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 HCMOS RAM. Program memory is provided by the completely bus compatible MC65516 2K HCMOS 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 (MC14597, MC14598 and MC14599), remote I/O (MC14469) and frequency synthesizers (MC14156 and MC145144).

POWER SAVINGS – Energy efficiency is, of course, the chief HCMOS attraction. HCMOS 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 and M6804 Families are static in design, low-speed operating current is extremely low.

STATIC DESIGN – The clock of a static HCMOS microprocessor may be at any frequency below the specified maximum. HCMOS 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 work load 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 Family HCMOS 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 cost 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 HCMOS 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 MATRIX

		M6805 Family											
		MC6805P2	MC6805P4	MC6805P6	MC6805U2	MC6805U3	MC6805R2	MC6805R3	MC6805T2	MC68705P3	MC68705P	MC68705U3	MC68705R3
Processor	Bits	8 Bits											
	Instruction Set	Control Optimization of MC6800											
	Registers	2 General Purpose and 3 Special Registers											
	Addressing Modes	10 Addressing Modes											
	Basic Inst Types	59 Basic Instruction Types											
	Total Instructions	207 Total Instructions											
	μs/Avg Inst	4.9 to 5.4 μs per Average Instruction (1 MHz)											
	Subroutines	13 Subroutine Levels											
Technology		HMOS											
Memory	Mask ROM	1.1K ROM	1.1K ROM	1.8K ROM	2K ROM	3.8K ROM	2K ROM	3.8K ROM	2.5K ROM	–	–	–	–
	EPROM	–	–	–	–	–	–	–	–	1.8K EPROM	1.8K EPROM	3.8K EPROM	3.8K EPROM
	RAM Bytes	64 RAM	112 RAM	64 RAM	64 RAM	112 RAM	64 RAM	112 RAM	64 RAM	112 RAM	112 RAM	112 RAM	112 RAM
Package Size		28 Pins	28 Pins	28 Pins	40 Pins	40 Pins	40 Pins	40 Pins	28 Pins	28 Pins	28 Pins	40 Pins	40 Pins
Input/ Output Pins	Inputs	–	–	–	8 Inputs	8 Inputs	2 to 5 Inputs	2 to 5 Inputs	–	–	–	8 Inputs	2 to 5 Inputs
	Outputs	–	–	–	–	–	–	–	–	–	–	–	–
	Mask Bidir	–	–	–	–	–	–	–	–	–	–	–	–
	Prog Bidir	20 I/O	20 I/O	20 I/O	24 I/O	24 I/O	24 I/O	24 I/O	19 I/O	20 I/O	20 I/O	24 I/O	24 I/O
	Spec. Func	–	–	–	–	–	1 to 4 Analog	1 to 4 Analog	2 Special	–	–	–	1 to 4 Analog
Expansion Bus		–	–	–	–	–	–	–	–	–	–	–	–
Special Function I/O	Display Decoder	–	–	–	–	–	–	–	–	–	–	–	–
	High Current Drive	10 mA Drive on 8 Pins											
	Analog Inputs	–	–	–	–	–	8-Bit A/D	–	–	–	–	–	8-Bit A/D
	Serial I/O	Shift Register I/O with Bit Manipulation Instructions											
	Freq Synth	–	–	–	–	–	–	–	Freq Synth	–	–	–	–
Standby RAM		–	Stby. RAM	–	–	–	–	–	–	–	–	–	–
Timer	Prescale Bits	7 Prescaler Bits											
	Counter Bits	8-Bit Counter											
	Timer Functions	1 Timer Function at a Time											
Interrupts	Timer Interrupt	Timer Interrupt											
	External IRQ		1 Ext IRQ		2 Ext IRQs				1 Ext IRQ	1 Ext IRQ	1 Ext IRQ	2 Ext IRQs	
	Serial I/O IRQ		–	–	–	–			–	–	–	–	–
Development Support	ICs	EPROM and ROM-Less Versions											
	Dev System	EXORciser®											
	Emulation	User System Emulator											
	Assembler	Macro Assembler											
	HL Language	–											

SINGLE-CHIP MICROCOMPUTER FAMILIES SELECTOR GUIDE

TECHNOLOGY

HMOS/NMOS	Page	HMOS/NMOS	Page
MC6801	4-61	MC68705P3	4-836
MC6803	4-166	MC68705P5	4-836
MC6801U4	4-102	MC68705R3	4-861
MC6803U4	4-102	MC68705U3	4-889
MC68701	4-797		
MC6805P2	4-200		
MC6805P4	4-223		
MC6805P6	4-1078		
MC6805R2	4-247	**CMOS**	**Page**
MC6805R3	4-272		
MC6805T2	4-275	MC146805E2	4-912
MC6805U2	4-302	MC146805F2	4-947
MC6805U3	4-325	MC146805G2	4-971

PROCESSING POWER

8-Bit	Page	8-Bit	Page
MC6801	4-61	MC68705P3	4-836
MC6803	4-166	MC68705P5	4-836
MC6801U4	4-102	MC68705R3	4-861
MC6803U4	4-102	MC68705U3	4-889
MC68701	4-797	MC146805E2	4-912
MC6805P2	4-200	MC146805F2	4-947
MC6805P4	4-223	MC146805G2	4-971
MC6805P6	4-1078		
MC6805R2	4-247		
MC6805R3	4-272		
MC6805T2	4-275		
MC6805U2	4-302		
MC6805U3	4-325		

SINGLE-CHIP MICROCOMPUTER FAMILIES FEATURES MATRIX

M6805 Family (continued)					MC6804 Family			M6801 Familiy			M6801 Family		
MC146805G2	MC146805F2	MC146805E2	MC1468705G2	MC3870	MC6804P2	MC68HC04P2	MC68HC04P3	MC6801	MC68701	MC6803	MC6801U4	MC68701U4	MC6803U4
8 Bits Control Optimization of MC6800 2 General Purpose and 3 Special Registers 10 Addressing Modes 61 Basic Inst Types 209 Total Instructions 3.9 to 4.0 μs/Avg Inst (1 MHz) 29 Levels	13 Levels	29 Levels	29 Levels	8 Bits F8 7 Registers 5 Addr Modes 54 Basic Inst 76 Total Inst 4.7 μs/Inst 1 Level	8 Bits Optimization of MC6805 3 General Purpose and 3 Special Registers 10 Addressing Modes 43 Basic Inst Types + 2 CMOS 245 Total Instructions 8 to 16 μs/Avg Inst (1 MHz) 4 Subroutine Levels			8 Bits Super Set of MC6800 2 General Purpose, 4 Special Registers 7 Addressing Modes 75 Basic Inst Types 219 Total Instructions 3.7 μs/Avg Inst (1 MHz) Indefinite Levels			8 Bits Super Set of MC6800 2 General Purpose, 4 Special Registers 7 Addressing Modes 75 Basic Inst Types 219 Total Instructions 3.7 μs/Avg Inst (1 MHz) Indefinite Levels		
CMOS				NMOS	HMOS	HCMOS	HCMOS	HMOS			HMOS		
2K ROM — 112 RAM	1K ROM — 64 RAM	No ROM — 112 RAM	— 2K EPROM 112 RAM	2K ROM — 64 RAM	1K ROM — 32 RAM	1K ROM — 32 RAM	2K ROM — 128 RAM	2K ROM — 128 RAM	— 2K EPROM 128 RAM	No ROM — 128 RAM	4K ROM — 192 RAM	— 4K EPROM 192 RAM	No ROM — 192 RAM
40 Pins	28 Pins	40 Pins	40 Pins	40 Pins	28 Pins	28 Pins	28 Pins	40 Pins	40 Pins	40 Pins	40 Pins	40 Pins	40 Pins
— — — 32 I/O —	4 Inputs — — 16 I/O —	— — — 16 I/O —	— — — 32 I/O —	— — 32 I/O — —	— — — 20 I/O —	— — — 20 I/O —	— — — 20 I/O —	— — — 24 to 31 I/O 0 to 7 Special Funct Pins	— — —	— — —	— — — 24 to 31 I/O 0 to 10 Special Funct Pins	— — —	— — —
—	—	8K Addr	—	—	—	—	—	64K Addressability			64K Addressability		
— 10 mA, 4 Pins — Shift Register I/O with Bit Manipulation Instructions —	— — — —	— — — —	— 10 mA, 4 Pins — —	— — — — —	— — — — —	— — — — —	— — — — —	— — — 8-Bit UART + Bit Rate Gen. —	— — — —	— — — —	— — — 8-Bit UART + Bit Rate Gen. —	— — — —	— — — —
—	—	—	—	—	Standby RAM			—	—	—	—	—	—
7 Prescaler Bits 8-Bit Counter 1 Timer Function at a Time				+ 200 Prescale 8-Bit Counter 1 Function	7 Prescaler Bits 8-Bit Counter 1 Time Function at a Time			— 16-Bit Timer 3 Simultaneous Timer Functions	—	—	— 16-Bit Timer 6 Simultaneous Timer Functions	—	—
Timer Interrupt 1 Ext IRQ —	—	—	—	Timer IRQ or 1 Ext IRQ —	1 Ext IRQ			3 Timer Interrupts 2 Ext Interrupts 2 Serial I/O IRQs			6 Timer Interrupts 2 Ext Interrupts 2 Serial I/O IRQs		
EPROM and ROM-Less Versions EXORciser® User System Emulator Macro Assembler —				— EXORciser® USE Assembler —	EPROM Version EXORciser® User System Emulator Macro Assembler —			EPROM and ROM-Less Versions EXORciser® User System Emulator Macro Assembler Fortran, Basic, MPL			EPROM and ROM-Less Versions EXORciser® User System Emulator Macro Assembler Fortran, Basic, MPL		

FUNCTIONAL BLOCKS

MPU with On-Chip RAM	Page
MC6803	4-166
MC6803U4	4-102
MC146805E2	4-912

MCU	Page
MC6801	4-61
MC6801U4	4-102
MC68701	4-797
MC6805P2	4-200
MC6805P4	4-223

MCU	Page
MC6805P6	4-1078
MC6805R2	4-247
MC6805R3	4-272
MC6805T2	4-275
MC6805U2	4-302
MC6805U3	4-325
MC68705P3	4-836
MC68705P5	4-836
MC68705R3	4-861
MC68705U3	4-889
MC146805F2	4-947
MC146805G2	4-971

LANGUAGE ORIENTATION

Low-Level Language	Page
MC6801	4-61
MC6803	4-166
MC6801U4	4-102
MC6803U4	4-102
MC68701	4-797
MC6805P2	4-200
MC6805P4	4-223
MC6805P6	4-1078
MC6805R2	4-247
MC6805R3	4-272
MC6805T2	4-275
MC6805U2	4-302

Low-Level Language	Page
MC6805U3	4-325
MC68705P3	4-836
MC68705P5	4-836
MC68705R3	4-861
MC68705U3	4-889
MC146805E2	4-912
MC146805F2	4-947
MC146805G2	4-971

PERIPHERAL AND INTERFACE COMPONENTS

Motorola manufactures and is continuing in new design efforts to provide you with an extensive selection of efficient, cost effective peripheral and interface components.

PERIPHERAL AND INTERFACE COMPONENTS SELECTOR GUIDE

FOR NMOS/HMOS MICROPROCESSOR SYSTEMS

Page

For MC6800 Two-Phase Clock Generation

For Parallel-Oriented Applications

For Serial Applications

For Complex Peripheral Control

For Television Display

PERIPHERAL AND INTERFACE COMPONENTS SELECTOR GUIDE
(continued)

FOR SYSTEM EXPANSION

FOR CMOS SYSTEMS

The Motorola M6800 Generic Bus Concept and Use

The M6800 Generic Bus Concept and Use

M6800 – M6801 – M6802 – M6809 – M6800 FAMILY PERIPHERALS

After more than 5 years of experience shipping many millions of microprocessors and peripherals, Motorola has collected, coordinated, and improved the bus timing parameters for these 8-bit devices. The smaller geometries and reduced capacitances obtained by introduction of optical reductions of existing mask sets, the use of new process techniques such as HMOS I and HMOS II, and the natural improvement in product yield that comes with experience have allowed Motorola to improve many performance parameters.

The new enhanced peripheral bus timing specifications allow their use in even wider ranges of applications and yet maintain complete compatibility with existing systems. This section provides a discussion about:

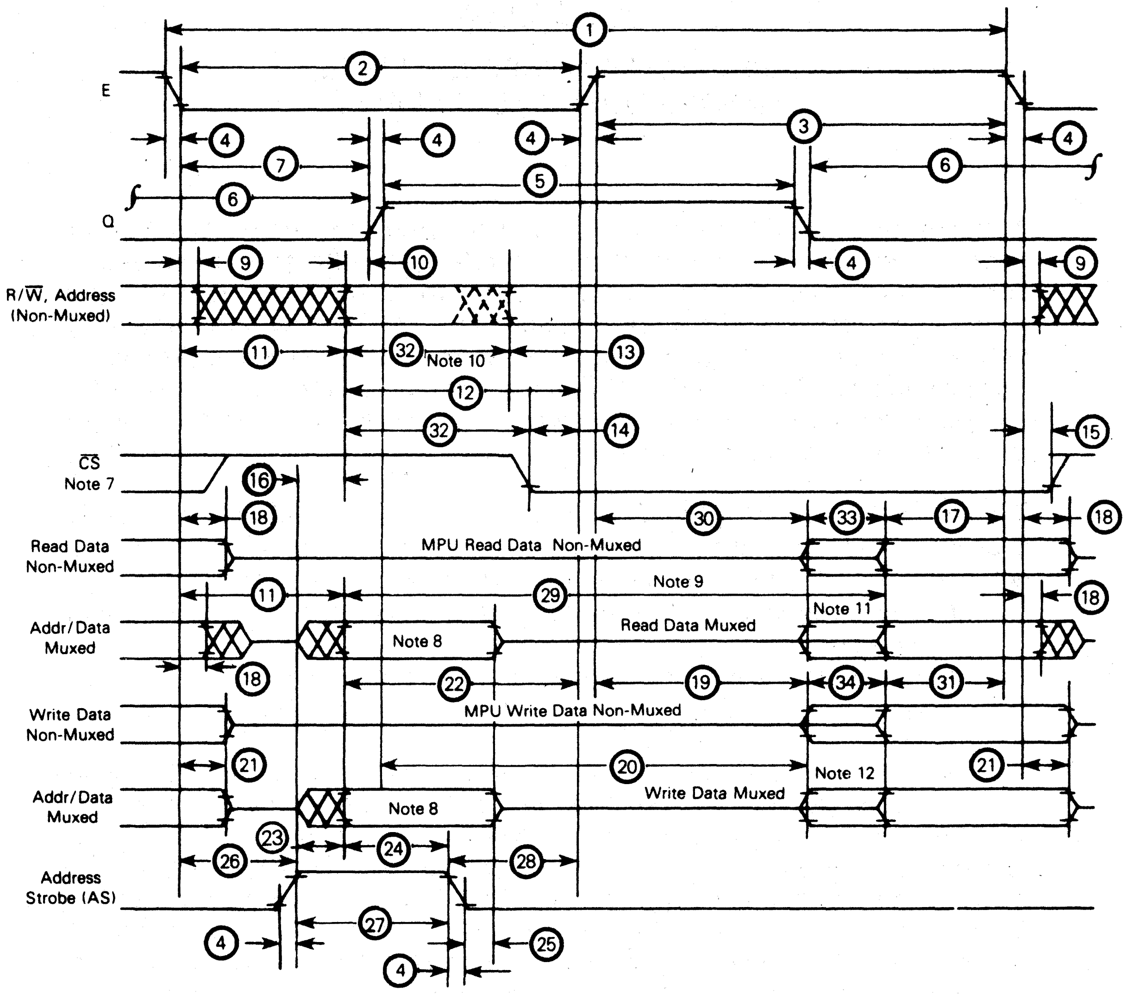
- The Generic Bus concept for both 8-bit NMOS/HMOS and CMOS devices.
- A complete set of bus timing for all of the 8-bit microprocessors (except the MC6800) and peripherals in one table (grouped by speed).
- A set of equations for calculating worst-case bus timing. These appear as notes under appropriate Generic Bus timing diagram.

The bus timing diagram shown in Figure 2-1 illustrates the waveforms needed for or generated by all but two of Motorola's mid-range family of 8-bit microprocessors and peripherals. (The MC6800, although it began the family, is not shown, due to the nature of the $\phi 1$, $\phi 2$, and DBE input signals.) A generic bus timing characteristics table gives a side-by-side comparison of major microprocessor types within the M6800 family, along with the input and output specifications common to all bus peripherals. These tables are shown as Tables 2-1 through 2-3.

A *subset* of the generic bus timing diagram and characteristics table appears in each data sheet (except the MC6800 and MCM6810) showing only the signals and identification numbers appropriate to that part.

A standard bus specification such as this allows a comparison of multiplexed and non-multiplexed processors; all timings associated with a particular system are shown together, which makes worst-case design easier.

FIGURE 2-1. NMOS/HMOS GENERIC BUS TIMING DIAGRAM



NOTES:

1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
4. For MC6800, write data is referenced to DBE, not E; see M6800 pages.
5. For MC6800, address delay is referenced to $\phi 1$, not E; see M6800 pages.
6. Clock pulse rise and fall time for MC6800 measured to $V_{CC} - 0.6$ V; see M6800 pages.
7. CS and CS̄ on MC6810 have same timing.
8. Address valid on the occurrence of the latest of 11, 12, 16, 22, or 23.
9. Usable access time is computed by: 1 − (4 + 11 + 17), or 12 + 4 − 17, see note 8 (except for MC6809; for MC6809, by 1 − 4 − 7 max + 10 − 17).
10. Usable address buffer time is computed by: 2 − (11 + 13); see note 8 (except for MC6809; for MC6809, by: 2 − 7 max + 10 − 13).
11. Usable read data buffer time is computed by: 3 − (17 + 30).
12. Usable write data buffer time is computed by: 3 − (19 + 31), except for MC6809; for MC6809, by: 1 − (4 + 7 max + 4 + 20 + 31).

TABLE 2-1. NMOS/HMOS GENERIC BUS TIMING CHARACTERISTICS FOR 1.0 MHz OPERATION

Ident. Number	Characteristics	Symbol	6800		6801		6802		6809		6821/ 6859		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	1.0	2.0	1.0	10	1.0	10	1.0	10	μs
2	Pulse Width, E Low (See Note 6)	PW_{EL}	405	9500	430	1000	450	5000	430	5000	430	9500	ns
3	Pulse Width, E High (See Note 6)	PW_{EH}	450	9500	450	1000	450	9500	450	9500	450	9500	ns
4	Clock Rise and Fall Time (See Note 6)	t_r, t_f	–	100	–	25	–	25	–	25	–	25	ns
5	Pulse Width, Q High	PW_{QH}	–	–	–	–	–	–	430	5000	–	–	ns
6	Pulse Width, Q Low	PW_{QL}	–	–	–	–	–	–	450	9500	–	–	ns
7	Delay Time, E to Q Rise*	t_{AVQ}	–	–	–	–	–	–	200	250	–	–	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	10	–	10	–	ns
10	Address Valid Time to Q Rise*	t_{AQ}	–	–	–	–	–	–	50	–	–	–	ns
11	Address Delay from E Low (See Note 5)	t_{AD}	–	270	–	–	–	–	–	–	–	–	ns
12	Non-Muxed Address Valid Time to E* (MPU)	t_{AV}	–	–	200	–	160	–	–	–	–	–	ns
13	Addres Setup Time Before E (Periph.)	t_{AS}	–	–	–	–	–	–	–	–	80	–	ns
14	Chip Select Setup Time Before E	t_{CS}	–	–	–	–	–	–	–	–	80	–	ns
15	Chip Select Hold Time	t_{CH}	–	–	–	–	–	–	–	–	10	–	ns
16	Non-Muxed Address Delay Time from AS	t_{AD}	–	–	–	–	–	–	–	–	–	–	ns
17	Read Data Setup Time	t_{DSR}	100	–	80	–	100	–	80	–	–	–	ns
18	Read Data Hold Time	T_{DHR}	10	–	10	–	10	–	10	–	20	50	ns
19	Write Data Delay Time (See Note 4)	t_{DDW}	–	225	–	225	–	225	–	–	–	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	–	–	–	–	–	–	200	–	–	ns
21	Write Data Hold Time (See Note 4)	t_{DHW}	10	–	20	–	30	–	30	–	10	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	–	–	200	–	–	–	–	–	–	–	ns
23	Muxed Address Delay Time from AS	t_{ADAS}	–	–	–	–	–	–	–	–	–	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	–	–	60	–	–	–	–	–	–	–	ns
25	Muxed Address Hold Time	t_{AHL}	–	–	20	–	–	–	–	–	–	–	ns
26	Delay Time, AS to E Rise*	t_{ASD}	–	–	90	–	–	–	–	–	–	–	ns
27	Pulse Width, AS High*	PW_{ASH}	–	–	220	–	–	–	–	–	–	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	–	–	90	–	–	–	–	–	–	–	ns
29	Usable Access Time* (See Note 9)	t_{ACC}	605	–	570	–	605	–	695	–	–	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	–	–	–	–	–	–	–	–	290	ns
31	Peripheral Input Data Setup Time	t_{DSW}	–	–	–	–	–	–	–	–	165	–	ns
32	Buffer Logic Delay Time Address, $\overline{CS}$, R/$\overline{W}$ (See Note 10)	t_{BDA}	55	–	120	–	100	–	150	–	–	–	ns
33	Buffer Delay Time, Read Data (See Note 11)	t_{BDR}	60	–	80	–	60	–	80	–	–	–	ns
34	Buffer Delay Time, Write Data (See Note 12)	t_{BDW}	60	–	60	–	60	–	365	–	–	–	ns

*At specified cycle time

TABLE 2-2. NMOS/HMOS GENERIC BUS TIMING CHARACTERISTICS FOR 1.5 MHz OPERATION

Ident. Number	Characteristics	Symbol	68A00		68A01		68A02		68A09		68A21/ 68A59		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	0.667	10	0.667	2.0	0.667	10	0.667	10	0.667	10	µs
2	Pulse Width, E Low (See Note 6)	PW_{EL}	230	9500	300	1000	280	5000	280	5000	280	9500	ns
3	Pulse Width, E High (See Note 6)	PW_{EH}	280	9500	300	1000	280	9700	280	9700	280	9500	
4	Clock Rise and Fall Time (See Note 6)	t_r, t_f	–	100	–	25	–	25	–	25	–	25	ns
5	Pulse Width, Q High	PW_{QH}	–	–	–	–	–	–	280	5000	–	–	ns
6	Pulse Width, Q Low	PW_{QL}	–	–	–	–	–	–	280	9700	–	–	ns
7	Delay Time, E to Q Rise*	t_{AVQ}	–	–	–	–	–	–	130	165	–	–	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	10	–	10	–	ns
10	Address Valid Time to Q Rise*	t_{AQ}	–	–	–	–	–	–	25	–	–	–	ns
11	Address Delay from E Low (See Note 5)	t_{AD}	–	180	–	–	–	–	–	–	–	–	ns
12	Non-Muxed Address Valid Time to E*(MPU)	t_{AV}	–	–	115	–	100	–	–	–	–	–	ns
13	Address Setup Time Before E (Periph.)	t_{AS}	–	–	–	–	–	–	–	–	60	–	ns
14	Chip Select Setup Time Before E	t_{CS}	–	–	–	–	–	–	–	–	60	–	ns
15	Chip Select Hold Time	t_{CH}	–	–	–	–	–	–	–	–	10	–	ns
16	Non-Muxed Address Delay Time from AS	t_{AD}	–	–	–	–	–	–	–	–	–	–	ns
17	Read Data Setup Time	t_{DSR}	60	–	60	–	70	–	60	–	–	–	ns
18	Read Data Hold Time	T_{DHR}	10	–	10	–	10	–	10	–	20	50	ns
19	Write Data Delay Time (See Note 4)	t_{DDW}	–	200	–	170	–	170	–	–	–	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	–	–	–	–	–	–	140	–	–	ns
21	Write Data Hold Time (See Note 4)	t_{DHW}	10	–	20	–	20	–	30	–	10	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	–	–	115	–	–	–	–	–	–	–	ns
23	Muxed Address Delay Time from AS	t_{ADAS}	–	–	–	–	–	–	–	–	–	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	–	–	40	–	–	–	–	–	–	–	ns
25	Muxed Address Hold Time	t_{AHL}	–	–	20	–	–	–	–	–	–	–	ns
26	Delay Time, AS to E Rise*	t_{ASD}	–	–	60	–	–	–	–	–	–	–	ns
27	Pulse Width, AS High*	PW_{ASH}	–	–	140	–	–	–	–	–	–	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	–	–	60	–	–	–	–	–	–	–	ns
29	Usable Access Time* (See Note 9)	t_{ACC}	400	–	345	–	310	–	440	–	–	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	–	–	–	–	–	–	–	–	180	ns
31	Peripheral Input Data Setup Time	t_{DSW}	–	–	–	–	–	–	–	–	80	–	ns
32	Buffer Logic Delay Time Address, $\overline{CS}$, R/$\overline{W}$ (See Note 10)	t_{BDA}	40	–	55	–	40	–	80	–	–	–	ns
33	Buffer Delay Time, Read Data (See Note 11)	t_{BDR}	30	–	40	–	20	–	30	–	–	–	ns
34	Buffer Delay Time, Write Data (See Note 12)	t_{BDW}	0	–	50	–	30	–	230	–	–	–	ns

*At specified cycle time

TABLE 2-3. NMOS/HMOS GENERIC BUS TIMING CHARACTERISTICS FOR 2.0 MHz OPERATION

Ident. Number	Characteristics	Symbol	68B00		68B01		68B02		68B09		68B21/ 68B59		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	0.5	10	0.5	2.0	0.5	10	0.5	10	0.5	10	μs
2	Pulse Width, E Low (See Note 6)	PW_{EL}	210	9500	210	1000	210	5000	210	5000	210	9700	ns
3	Pulse Width, E High (See Note 6)	PW_{EH}	220	950	220	1000	220	9700	220	9700	220	9700	ns
4	Clock Rise and Fall Time (See Note 6)	t_r, t_f	–	100	–	20	–	20	–	20	–	20	ns
5	Pulse Width, Q High	PW_{QH}	–	–	–	–	–	–	210	5000	–	–	ns
6	Pulse Width, Q Low	PW_{QL}	–	–	–	–	–	–	220	9700	–	–	ns
7	Delay Time, E to Q Rise*	t_{AVQ}	–	–	–	–	–	–	80	125	–	–	ns
9	Address Hold	t_{AH}	10	–	10	–	10	–	10	–	10	–	ns
10	Address Valid to Q Rise*	t_{AQ}	–	–	–	–	–	–	15	–	–	–	ns
11	Address Delay from E Low (See Note 5)	t_{AD}	–	150	–	–	–	–	–	–	–	–	ns
12	Non-Muxed Address Valid Time to E*(MPU)	t_{AV}	–	–	70	–	50	–	–	–	–	–	ns
13	Address Setup Time Before E (Periph.)	t_{AS}	–	–	–	–	–	–	–	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	–	–	–	–	–	–	–	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	–	–	–	–	–	–	–	–	10	–	ns
16	Non-Muxed Address Delay Time from AS	t_{AD}	–	–	–	–	–	–	–	–	–	–	ns
17	Read Data Setup Time	t_{DSR}	40	–	40	–	60	–	40	–	–	–	ns
18	Read Data Hold Time	T_{DHR}	10	–	10	–	10	–	10	–	20	50	ns
19	Write Data Delay Time (See Note 4)	t_{DDW}	–	160	–	120	–	160	–	–	–	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	–	–	–	–	–	–	110	–	–	ns
21	Write Data Hold Time (See Note 4)	t_{DHW}	10	–	10	–	20	–	30	–	10	–	ns
22	Muxed Address Valid Time to E Rise*	t_{AVM}	–	–	80	–	–	–	–	–	–	–	ns
23	Muxed Address Delay Time from AS	t_{ADAS}	–	–	–	–	–	–	–	–	–	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	–	–	20	–	–	–	–	–	–	–	ns
25	Muxed Address Hold Time	t_{AHL}	–	–	10	–	–	–	–	–	–	–	ns
26	Delay Time, AS to E Rise*	t_{ASD}	–	–	45	–	–	–	–	–	–	–	ns
27	Pulse Width, AS High*	PW_{ASH}	–	–	110	–	–	–	–	–	–	–	ns
28	Delay Time, AS to E Rise*	t_{ASED}	–	–	45	–	–	–	–	–	–	–	ns
29	Usable Access Time* (See Note 9)	t_{ACC}	290	–	260	–	235	–	330	–	–	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	–	–	–	–	–	–	–	–	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	–	–	–	–	–	–	–	–	60	–	ns
32	Buffer Logic Delay Time Address, $\overline{CS}$, R/$\overline{W}$ (See Note 10)	t_{BDA}	20	–	30	–	10	–	60	–	–	–	ns
33	Buffer Delay Time, Read Data (See Note 11)	t_{BDR}	30	–	30	–	15	–	30	–	–	–	ns
34	Buffer Delay Time, Write Data (See Note 12)	t_{BDW}	0	–	40	–	0	–	160	–	–	–	ns

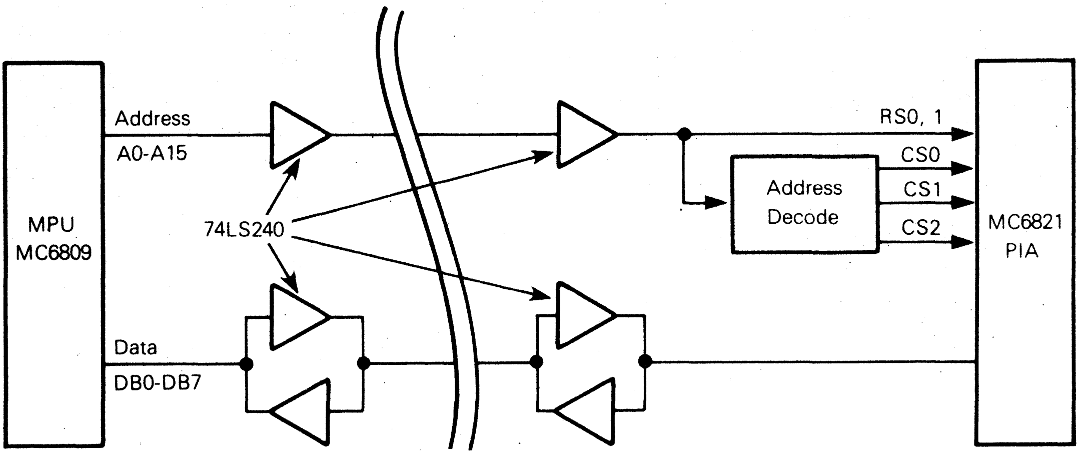
*At specified cycle time

DIAGRAM/TABLE USE

As an example of the use of the timing diagram, consider the simple bus connections of an MC6809 executing a read bus cycle with a peripheral such as the MC6821 shown in Figure 2-2. The bus cycle, identified as #1 in Figure 2-1 begins when $\bar{E}$ (#4) falls. The address output of the MC6809 is valid prior to the rising edge of Q by time #10.

The MC6821, as well as all other bus peripherals, requires that the register select inputs, derived from the address bus, be valid prior to the rising edge of $\bar{E}$ by time #13. Similarly, address decoding for assertion of chip select(s) must be valid by time #14. The time available for address buffers and decoding logic is given by #32 and can be calculated from the equation in note 10.

FIGURE 2-2. SIMPLE BUS CONNECTION — EXAMPLE



At 1 MHz, the characteristics table for the MC6809 shows that 150 nanoseconds are available for the combination of the two 74LS240 address buffers in series with the address decoding logic shown in Figure 2-2. Standard LS-series buffers and gates may easily be used in such a case. At 2 MHz, the MC6809 provides 60 nanoseconds for address decoding logic. Characteristic #32 on the 2 MHz table also illustrates the dramatic improvement in technology attendant with the introduction of the MC6801 and MC6809 when compared with the MC6800 and MC6802. Schottky buffering is easily done in a 2 MHz system with either of these newer processors; the MC6809 even allows ample time for use of LS-series buffers at 2 MHz.

To continue with the example, data access occurs and output buffers are enabled within the MC6821 for the duration of $\overline{E}$ (#3) when chip select is asserted. The peripheral output data will be valid by time #30. Propagation through any data bus buffer must be done quickly to ensure valid data to the MC6809 on or before the read data setup time (#17). This read buffer delay time is #33. The overall time between the existence of a valid address output from the microprocessor and the required input data valid (read setup time) is called the usable access time and is characteristic #29. This is the time provided by the microprocessor at any given bus rate, for address buffers, address decoders, ROM/RAM access time, and data bus buffers.

Table 2-4 shows a comparison of the different buffering arrangements possible in the various speed ranges for the MC6809 used in the example, and also for the MC6802 in a similar system. The buffers and gates listed are a representative sampling of those available. At 2 MHz, the MC68B02 does not provide any buffer delay time for a write operation. As a result, when operating at 2 MHz, the MC68B02 is limited to small unbuffered systems.

TABLE 2-4. BUFFER TIME EXAMPLES

MPU	Available Address Buffer/Decode Time (ns)	Available Buffers and Decode (ns)	Margin (ns)	Available Data Buffer Time (ns)		Available Buffers (ns)	Margin (ns)	
				R	W		R	W
MC6809	150	MC6888/8T28 +7440 (10+10+22)	108	80	365	MC6880/ 8T26 (14+14)	52	337
MC68A09	80	MC6888/8T28 MC+74LS40 (10+10+24)	36	30	230	MC6880/ 8T26 (14+4)	2	202
MC68B09	60	MC6888/8T28 MC+74LS40 (10+10+24)	16	30	160	MC6880/ 8T26 (14+14)	2	132
MC6802	100	MC74LS240 MC+74LS40 (14+18+24)	44	60	60	MC74LS240	28	28
MC68A02	40	MC74LS240 MC+74LS40 (7+7+24)	2	20	30	74S240 (7+7)	6	16
MC68B02	10	74S40=(6)	4	15	0	–	15	0

In the case of multiplexed-bus parts such as the MC6801 and MC68120, additional consideration must be given to the multiplexed address output. The address is defined to be valid, in the case of the MC6801, prior to the fall of address strobe by time #24. (In the MC68120, address is valid after the rise of address strobe by time #23.) This is sufficient time to allow capture of the address by a transparent latch, such as the 74LS373. The propagation time required by this latch, and the location of address strobe within the E low time, reduces the time available for address decoding.

When this approach is used with processors that include an on-chip oscillator (e.g., MC6801, MC6802, MC6809), specify address valid by xx (#10, #12, or #24) nanoseconds before $\overline{E}$ or $\overline{Q}$ rise or $\overline{AS}$ fall.

When this approach is used with processors that require external $\overline{E}$ or $\overline{AS}$ input (e.g., MC6809E or MC68120), specify address valid by xx (#11 or #23) nanoseconds after $\overline{E}$ fall or $\overline{AS}$ rise.

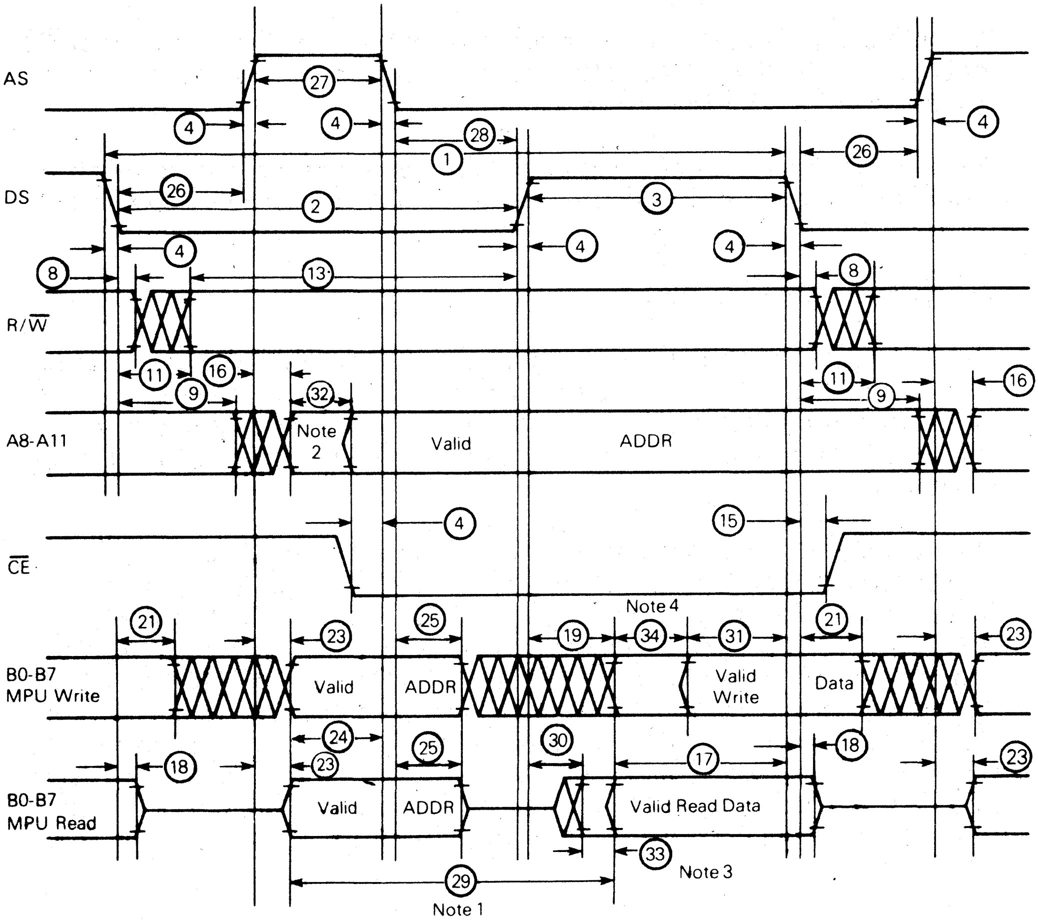
Do not consider the address output of any microprocessor valid until all appropriate parameters (#11, #12, #16, #22, or #23) have been considered.

THE CMOS GENERIC BUS

The timing diagrams shown in Figure 2-3 and the timing characteristics given in Table 2-5 may be considered as a special case of the generic bus. This information pertains to Motorola's rapidly expanding family of CMOS microprocessors and peripherals, beginning with the MC146805E2 and the MC146818.

The identification numbers shown are consistent with those in Figure 2-1 for the NMOS/HMOS generic bus. The waveforms have been drawn to indicate the input and output relationships of the MC146805E2 and MC146818. The characteristic table shows, in addition to the MC146805E2 and the MC146818, timing characteristics for Motorola's CMOS ROM, the MCM65516, available in two speed ranges. Both the MCM65516 and the MC146818 contain the MOTEL circuit, allowing direct application to both the Motorola generic bus and the competitive 8085-type bus. Always refer to the latest data sheet or other technical documentation for up-to-date characteristics for all Motorola microcomponents.

FIGURE 2-3. CMOS GENERIC BUS TIMING DIAGRAM



NOTES:

1. Available access time:
   mux = 24 (MPU) + 28 + 3 − 17
   non-mux = 27 + 28 + 3 − 17 − 16
2. Buffer delay (address):
   mux = 24 (MPU) − 24 (peripheral)
   non-mux = 27 − 16 − 24 (peripheral)
3. Buffer delay (data read) = 3 − 30 − 17
4. Buffer delay (data write) = 3 − 19 − 31
5. V_{Low} = 0.5 V, V_{High} = 2.0 V for V_{DD} = 3 V;
   V_{Low} = 0.8 V, V_{High} = V_{DD} − 2.0 V for V_{DD} = 5 V ± 10%.

TABLE 2-5. CMOS GENERIC BUS TIMING CHARACTERISTICS FOR 1.0 MHz OPERATION

Ident. Number	Characteristic	Symbol	MC146805E2		MC146818		MC65516-43		MC65516-55		Unit
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Bus Cycle Time	t_{cyc}	1000	DC	953	DC	580	DC	725	–	ns
2	Pulse Width, DS Low	PW_{DSL}	560	–	300	–	–	–	–	–	ns
3	Pulse Width, DS High	PW_{DSH}	375	–	325	–	–	–	–	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	30	–	30	–	–	–	–	ns
8	$R/\overline{W}$ Hold Time	t_{RWH}	10	–	–	–	–	–	–	–	ns
9	Non-Muxed Address Hold Time	t_{AH}	100	–	–	–	50**	–	80	–	ns
11	$R/\overline{W}$ Delay Time from DS	t_{RWD}	–	300	–	–	–	–	–	–	ns
13	$R/\overline{W}$ Setup Time to DS	t_{RWS}	–	195	–	–	–	–	–	–	ns
14	Chip Enable Setup Time to AS Fall	t_{CES}	–	–	55	–	50	–	50	–	ns
15	Chip Enable Hold Time	t_{CEH}	–	–	0	–	50**	–	80	–	ns
16	Non-Muxed Address Delay Time from AS	t_{AD}	0	100	–	–	–	–	–	–	ns
17	Read Data Setup Time	t_{DSR}	115	–	–	–	–	–	–	–	ns
18	Read Data Hold Time	t_{DHR}	0	140	10	100	0	160	0	160	ns
19	Write Data Delay Time	t_{DDW}	–	120	–	–	–	–	–	–	ns
21	Write Data Hold Time	t_{DHW}	55	–	0	–	–	–	–	–	ns
23	Muxed Address Delay Time from AS	t_{ADAS}	0	120	–	–	–	–	–	–	ns
24	Muxed Address Valid Time to AS Fall*	t_{ASL}	55	–	55	–	50	–	50	–	ms
25	Muxed Address Hold Time	t_{AHL}	60	180	20	–	50	–	80	–	ns
26	Delay Time, DS to AS Rise*	t_{DSD}	160	–	50	–	50	–	50	–	ns
27	Pulse Width, AS High*	PW_{ASH}	175	–	100	–	150	–	175	–	ns
28	Delay Time, AS to DS Rise*	t_{ASDSD}	160	–	90	–	100	–	160	–	ns
29	Usable Access Time*(See Note 1)	t_{ACC}	475	–	–	–	–	430	–	550	ns
30	Peripheral Output Data Delay Time	t_{DDR}	260	–	20	240	175	–	200	–	ns
31	Peripheral Input Data Setup Time	t_{DSW}	–	–	220	–	–	–	–	–	ns
32	Buffer Logic Delay Time Address, CS, $R/\overline{W}$ (See Note 2)	t_{BDA}	–	–	–	–	5	–	–	–	ns
33	Buffer Delay Time, Read Data (See Note 3)	t_{BDR}	20	–	–	–	–	–	–	–	ns
34	Buffer Delay Time, Write Data (See Note 4)	t_{BDW}	35	–	–	–	–	–	–	–	ns

*At specified cycle time.

** Hold times for MCM65516 are from AS fall and are easily met by MC146805E2.

Reliability

MOTOROLA INC.

RELIABILITY REPORT
8238

1982
MICROPROCESSOR FAMILY RELIABILITY

SEPTEMBER 1982

Motorola: the Reliability Name in Semiconductors

Test results contained herein are for information only. This report does not alter Motorola's standard warranty or product specifications.

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 1981. Included in the report are summary results of dynamic life testing for plastic and ceramic packaged devices, and moisture performance and thermal 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 60% confidence limit (see Appendix A). Tables 1 and 2 summarize the 1981 dynamic life test data for MOS Microprocessors.

Table 1. – Summary of Dynamic Life Test Results

Technology	Device Type	Wafer Lots	Test Devices	125°C Device Hours	70°C Equivalent Device Hours	Failures		Failure Rate %/1000 Hours	
						Catas- trophic	Total	Catastrophic	Total
NMOS	MC6800	3	135	129,672	6.4×10^6	1	1	0.030	0.030
	MC6802	5	225	247,728	8.4×10^6	0	0	0.011	0.011
	MC6810	6	270	271,152	15.5×10^6	1	1	0.013	0.013
	MC6821	18	810	793,608	43.0×10^6	3	3	0.010	0.010
	MC6844	3	135	128,376	7.7×10^6	1	2	0.025	0.039
	MC6845	6	270	203,280	10.2×10^6	0	1	0.009	0.019
	MC6846	5	294	356,688	10.8×10^6	5	6	0.056	0.066
	MC6847	3	133	110,040	5.4×10^6	2	6	0.052	0.120
	MC6850	3	135	136,000	8.8×10^6	0	0	0.010	0.010
	MC6852	4	169	174,720	10.9×10^6	1	7	0.018	0.075
	MC6854	6	252	250,536	11.7×10^6	1	3	0.017	0.035
	MC6860	2	76	72,984	5.7×10^6	4	4	0.091	0.091
	MC6862	3	170	159,740	12.2×10^6	5	7	0.051	0.068
	MC68488	2	81	78,336	3.2×10^6	1	2	0.061	0.095
	MC68701	3	135	134,112	5.6×10^6	2	2	0.052	0.052
	MC3870	3	135	135,240	9.7×10^6	1	1	0.020	0.020
TOTAL		75	3,425	3,382,212	175.2×10^6	28	46	0.016	0.027
HMOS	MC6801	8	357	346,352	13.9×10^6	1	2	0.014	0.022
	MC6805P2	19	1499	234,672	10.8×10^6	2	2	0.028	0.028
	MC6809	9	364	338,039	13.7×10^6	2	3	0.014	0.030
	MC68000	10	422	421,464	22.8×10^6	5	7	0.027	0.036
TOTAL		46	2,642	1,340,527	61.2×10^6	10	14	0.018	0.025
CMOS	MC146805E2	14	488	425,696	42.2×10^6	5	8	0.014	0.021
	MC146805G2	2	90	82,200	8.7×10^6	1	1	0.022	0.022
	MC141000	10	433	431,880	45.0×10^6	3	3	0.009	0.009
TOTAL		26	1,011	939,776	95.9×10^6	9	12	0.011	0.014
GRAND TOTAL		147	7,078	5,662,515	332.3×10^6	47	72	0.014	0.022

Table 2. – Microprocessor Family Dynamic Life Test Results

	Wafer Lots	Total Devices	125°C Device Hours	70°C Equivalent Device Hours	Failures		Failure Rate %/1000 Hours	
					Catas- trophic	Total	Catastrophic	Total
WAFER PROCESS TECHNOLOGY								
NMOS	75	3,425	3,382,212	175.2×10^6	28	46	0.016	0.027
HMOS	46	2,642	1,340,527	61.2×10^6	10	14	0.018	0.025
CMOS	26	1,011	939,776	95.9×10^6	9	12	0.011	0.014
PACKAGING SYSTEM TECHNOLOGY								
Ceramic	66	2,800	2,717,583	177.0×10^6	28	42	0.016	0.024
Plastic	81	4,278	2,944,932	155.3×10^6	19	30	0.013	0.020
TOTAL	147	7,078	5,662,515	332.3×10^6	47	72	0.014	0.022

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 contaminants. 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 corrosion of the die metallization, resulting in failures when the device is in use. Moisture reaches 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. Careful control of manufacturing processes, 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.). Results of these tests for 1981 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 96 hours of stress with electrical endpoint tests at 48 and 96 hours. Each electrical failure is subsequently analyzed to determine if corrosion is present on the surface of the die. Autoclave test results for 1981 are summarized in Table 3.

Table 3. – Autoclave Test Results
121°C 100% R.H. 15 psig

Hours	48	96
Failures/Sample	1/1626	3/1625
Percent Defective	0.06	0.18
Cumulative Percent Defective	0.06	0.24

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 resistance terminations; enable pins are disabled: and all other pins have alternate V_{DD} and V_{SS} 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 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 1981 are summarized in Table 4.

Table 4. – Temperature Humidity Bias Test Results
85°C 85% R.H. 5 Volts

Hours	168	504	1008
Failures/Sample	4/1717	8/1713	6/1705
Percent Failures	0.23	0.47	0.35
Cumulative Percent Failures	0.23	0.70	1.05

A Weibull plot (Figure 1) shows the continued improvement in T.H.B. performance as measured in 1979, 1980 and 1981. Improvements in processes, materials and manufacturing techniques have all contributed to this positive trend in plastic performance.

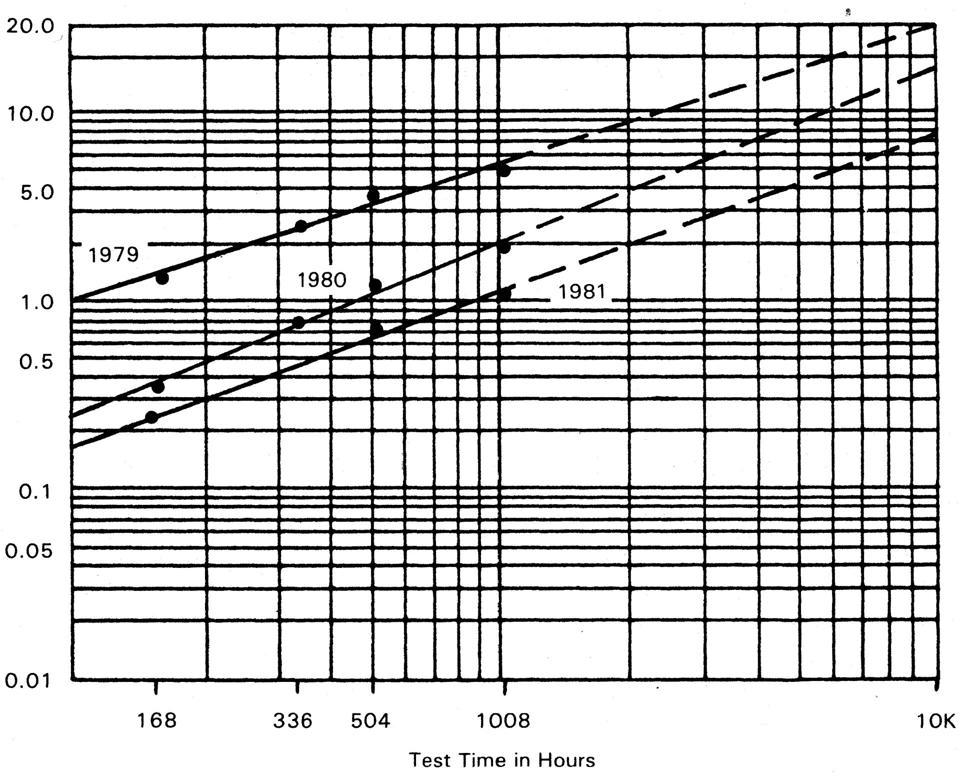


Figure 1. – Weibull Plot of Temperature Humidity Bias Test Results

THERMAL CYCLING PERFORMANCE

Thermal cycling accelerates the effects of thermal expansion mismatch between the various components of the plastic packaging system through rapid successive excursions to high and low temperature extremes. The differences in the thermal coefficients of expansion of the wire, die, encapsulant, and leadframe cause stresses to develop during thermal cycling. 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.

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 cool dwell, devices are heated to 150°C for at least ten minutes. The system employs a circulating air environment to assure rapid stabilization at the specified temperature. The dwell at each extreme, plus the total transition time, constitutes one test cycle.

Electrical measurements and high temperature continuity tests are performed after 100, 250, 500, 750 and 1000 cycles. The predominant failure mechanism 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. 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	250	500	750	1000
Failures/Sample	0/2406	3/2406	7/2403	5/2396	3/2391
Percent Defective	0.00	0.12	0.29	0.21	0.13
Cumulative Percent Defective	0.00	0.12	0.41	0.62	0.75

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 sudden change in temperature due to the high 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 to an adjacent chamber filled with fluorocarbon at 150°C and held for an equivalent time. The dwell at each endpoint, plus the total transition time, constitutes one test cycle. Thermal shock endpoint electrical measurements and high temperature continuity tests are performed at 100, 300, 500, 800, and 1000 cycles. Results of thermal shock tests performed in 1981 are shown in Table 6.

Table 6. – Thermal Shock Test Results
–65°C to +150°C Liquid-to-Liquid

Cycles	100	300	500	800	1000
Failures/Sample	3/832	2/829	0/827	2/824	4/822
Percent Defective	0.36	0.24	0.00	0.24	0.49
Cumulative Percent Defective	0.36	0.60	0.60	0.84	1.33

CONCLUSIONS

Reliability testing performed by Motorola during 1981 has produced excellent results. The specific test results included in this report are representative of Motorola Microprocessor expected field performance. Failure rate estimates have been based on the out-come of test 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 no potential problems with long-term corrosion. Thermal integrity testing shows that there are no problems with the packaging system. 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 verifies a history of continuous improvement which has made Motorola Microprocessor components the optimum choice for reliable performance.

APPENDIX A

Quality and Reliability System

A complete Reliability and Quality Assurance system is in place to monitor and control the performance of Motorola 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% Acceptable Quality Level (AQL) sampling plan. The visual/mechanical inspection is also performed to a 0.15% 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.M.E.) and a Scanning Auger Microprobe (S.A.M.); and a Product Analysis Laboratory for detailed analyses of failure modes and mechanisms for Microprocessor devices.

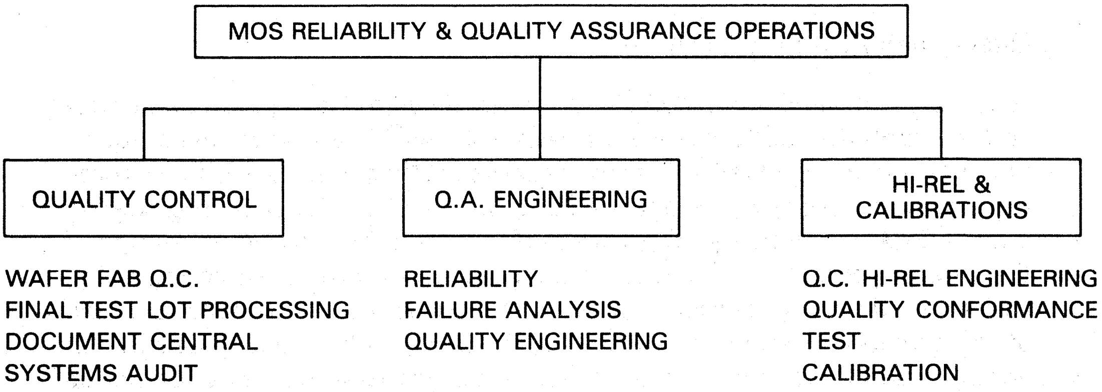


Figure A1. – Reliability and Quality Assurance Organisation

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 wide-spread 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 aluminium 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 dammage 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

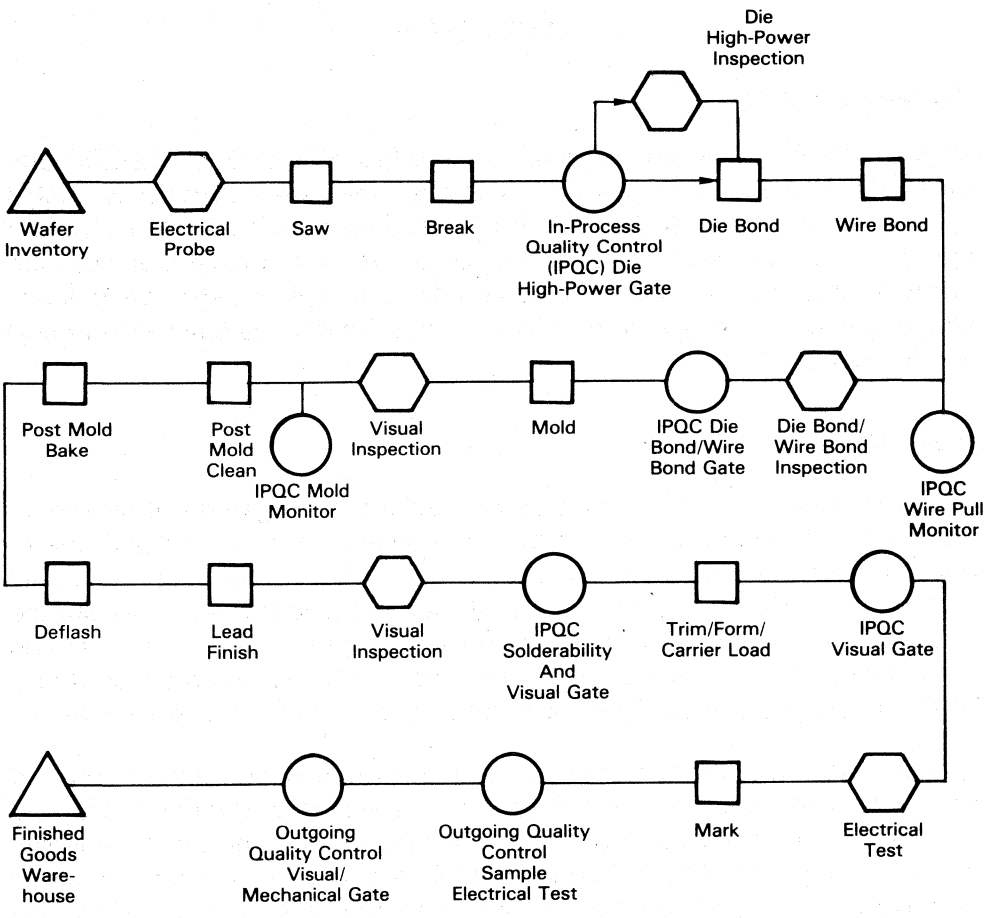


Figure B1. – Plastic Process Flow

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 heat spreaders. 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 improvement in electrical characteristics and reliability performance.

Another approach to lower thermal resistance for devices with high power dissipation is plastic assembly using a heat spreader. The heat spreader is an anodized aluminium piece part that sits below the plane of the leadframe. During the encapsulation process, the heat spreader is surrounded by plastic and becomes part of the package structure. Heat spreaders, 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 heat spreader 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 will also be offered in a heat spreader 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.

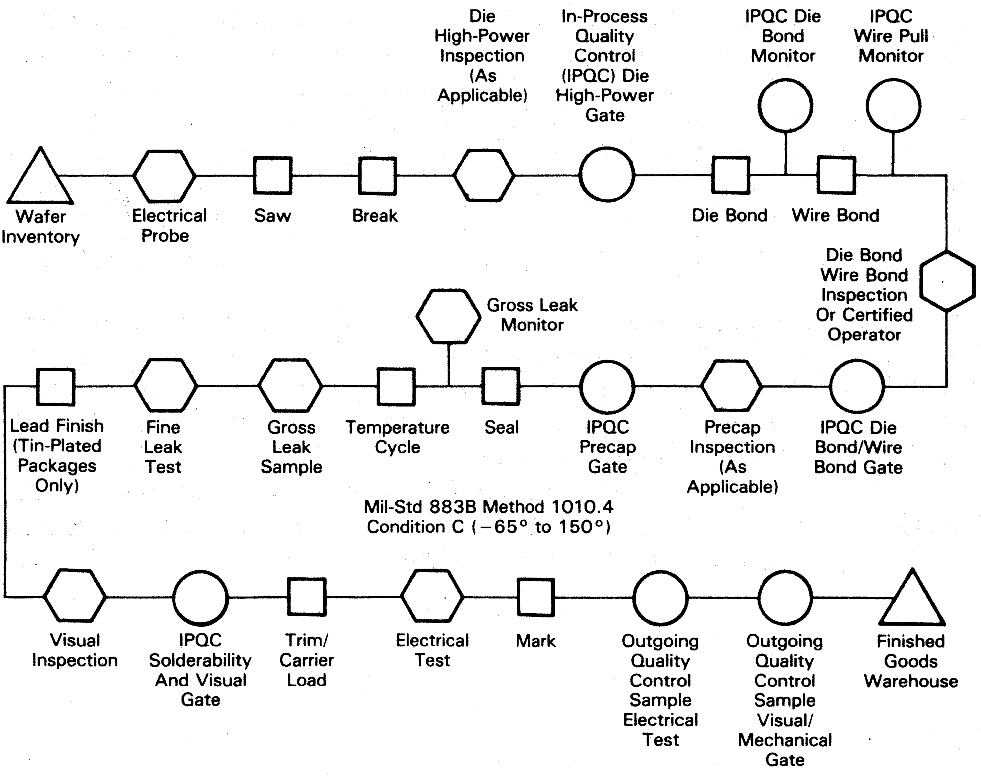


Figure B2. – Hermetic Process Flow

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 kovar or 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 lower sealing temperature, most manufacturers, including Motorola, employ a eutectic bond. Both aluminium 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 are ceramic, hermetic, employ a eutectic die bond, use ultrasonic aluminium 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 effect the 10 mil thick leadfram 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 (Fig. C1).

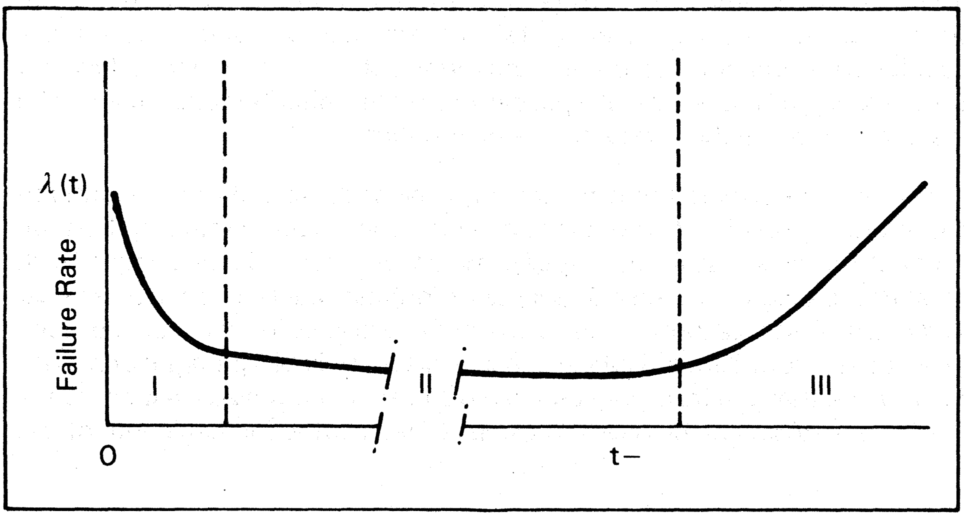


Figure C1. – Device Failures Rate as a Function of Time

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 population 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:

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

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 = Devices 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 (F_a) which equates test temperature with operating temperature is derived from the Arrhenius relationship:

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

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 (T_O) 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 electronvolt. A 1.0 eV was selected as an average value because a variety of different failure mechanisms exist for microprocessors 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. T_r and T_t of the equation are the average junction temperatures present at the rated and test ambients. Motorola uses junction, rather than ambient temperatures, because they produce acceleration factors that are more conservative and representative of actual conditions. These temperatures are calculated as follows:

$$T_J = T_A + P_D \cdot \Theta_{JA} \qquad (3)$$

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 to the failure rate under acceleration conditions. To determine the failure rate at the operating temperature, multiply the number of device hours by the acceleration factor $F_a(t)$.

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 the way Motorola Reliability Engineering expresses failure rates for Microprocessors. One other way of expressing failure rates is Failure In Time (FIT) which refers to failed units per 10^9 device hours ($1 \text{ FIT} = \lambda \times 10^4$).

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:

$$\text{MTTF} = \frac{1}{\lambda} \tag{4}$$

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. THB 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 V_{DD} 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 combinations, 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.

Semiconductor devices failing a rigorous electrical test frequently function adequately under nominal operating conditions such that they would not cause a system failure. These units have parameters which exceed data sheet limits, but which do not impact system reliability to the extent of catastrophic failures. About 35 percent of the failures observed in life testing were non-catastrophic in nature. The ultimate classification criterion must be predicated upon the maintenance of system operation and life test failure modes are, therefore, categorized as being either catastrophic or non-catastrophic. Opens, short, or functional (logic state) failure modes are considered to be catastrophic, and are used in the determination of the "system failure rate". Input leakage, three-state leakage, and parametric failure modes are non-catastrophic and are combined with the catastrophic failures to determine the "total failure rate".

Table D1. – Definition of Failure Modes

I. CATASTROPHIC MODES

A. OPENS – No electrical connection between an external terminal and corresponding die circuitry (possible intermittent). 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 – Failure of one or more output terminals to respond with a correct logical state under nominal supply, clock, and V_{IH}/V_{IL} level: a violation of the internal Boolean relationships defined by the circuit design.

III. NON-CATASTROPHIC MODES

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

B. 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.

C. PARAMETRIC – A broad classification of non-catastrophic failure modes which exclude leakages but include:

1. Failure to respond at one or more output terminals with a correct logical state under worst-case supply, clock, and V_{IH}/V_{IL} conditions; usually the result of excessive propagation delays, improper V_{OH}/V_{OL} levels, noise 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 not caused by leakage currents. Device is 100% functional.
3. Incorrect output analog voltage or current level not resulting in a functional failure.

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

APPENDIX E

Microprocessor Average Junction Temperatures & Gate Counts

MOS Technology	Device Type	Average Junction Temperature @Ta = 70°C Ceramic	Plastic A42	Plastic Cu	Equivalent Number of Gates
NMOS	MC6800	83	92	—	1,367
	MC6802/08	91	116	—	3,633
	MC6810	83	92	—	1,083
	MC6821	79	92	—	450
	MC6844	85	103	—	1,000
	MC6845	89	105	—	750
	MC6846	89	109	—	3,755
	MC6847	83	94	—	833
	MC6850	81	92	—	580
	MC6852	83	91	—	907
	MC6854	89	101	—	1,400
	MC6860	78	82	—	473
	MC6862	78	84	—	483
	MC68488	85	98	—	893
	MC68701	—	—	—	8,533
HMOS	MC6801/03	95	*96	97	8,533
	MC6805P2	88	106	95	4,833
	MC6805R2/U2	—	—	—	6,430
	MC6809	92	117	96	3,000
	MC6829	92	117	96	3,293
	MC68000	97	*95	—	12,667
	MC68120	96	—	—	9,644
	MC68705R3	—	—	—	14,433
CMOS	MC146805E2	71	72	—	2,800
	MC146805F2	—	—	—	5,633
	MC146805G2	71	72	—	5,350

NOTE: The above assumes 4 devices/gate for CMOS and 3 devices/gate for NMOS and HMOS and that ROM based parts use only 50% of the devices in ROM.

* = Plastic packages with molded in heat spreader.
A42 = Plastic packages with Alloy 42 leadframe.
Cu = Plastic packages with copper leadframe.

Data Sheets

DATA SHEETS

Data sheets at Motorola are grouped into one of three categories depending on the amount of information presented. These categories are:

Product Preview — This is the first official information released about a potential device. It contains very basic information about a product and usually precedes sample devices by approximately six months. This type of data sheet is distinguished by the words "Product Preview" appearing in the header of the first page as shown in Figure 4-1.

Advanced Information — This information is released with sample devices. It contains an extensive discussion of device operation and provides complete parametric information such as Maximum Ratings, Thermal Characteristics, Electrical Characteristics, Bus Timing, and I/O Port Timing as applicable. Timing diagrams are included to support the tabular material. All of the parametric information given is the result of early testing of initial product from the manufacturing process. Values given are subject to change without notice. This type of data sheet is distinguished by the words "Advanced Information" appearing in the header of the first page as shown in Figure 4-1.

Final Data Sheet — This data sheet evolves from the Advanced Information data sheet. It is a result of test information collected from a fully-implemented manufacturing process. The parametric information has been analyzed and approved. Motorola considers this is a fully characterized device. This type of data sheet is distinguished by the absence of any designation appearing in the header of the first page as shown in Figure 4-1.

DATA SHEET ORGANIZATION

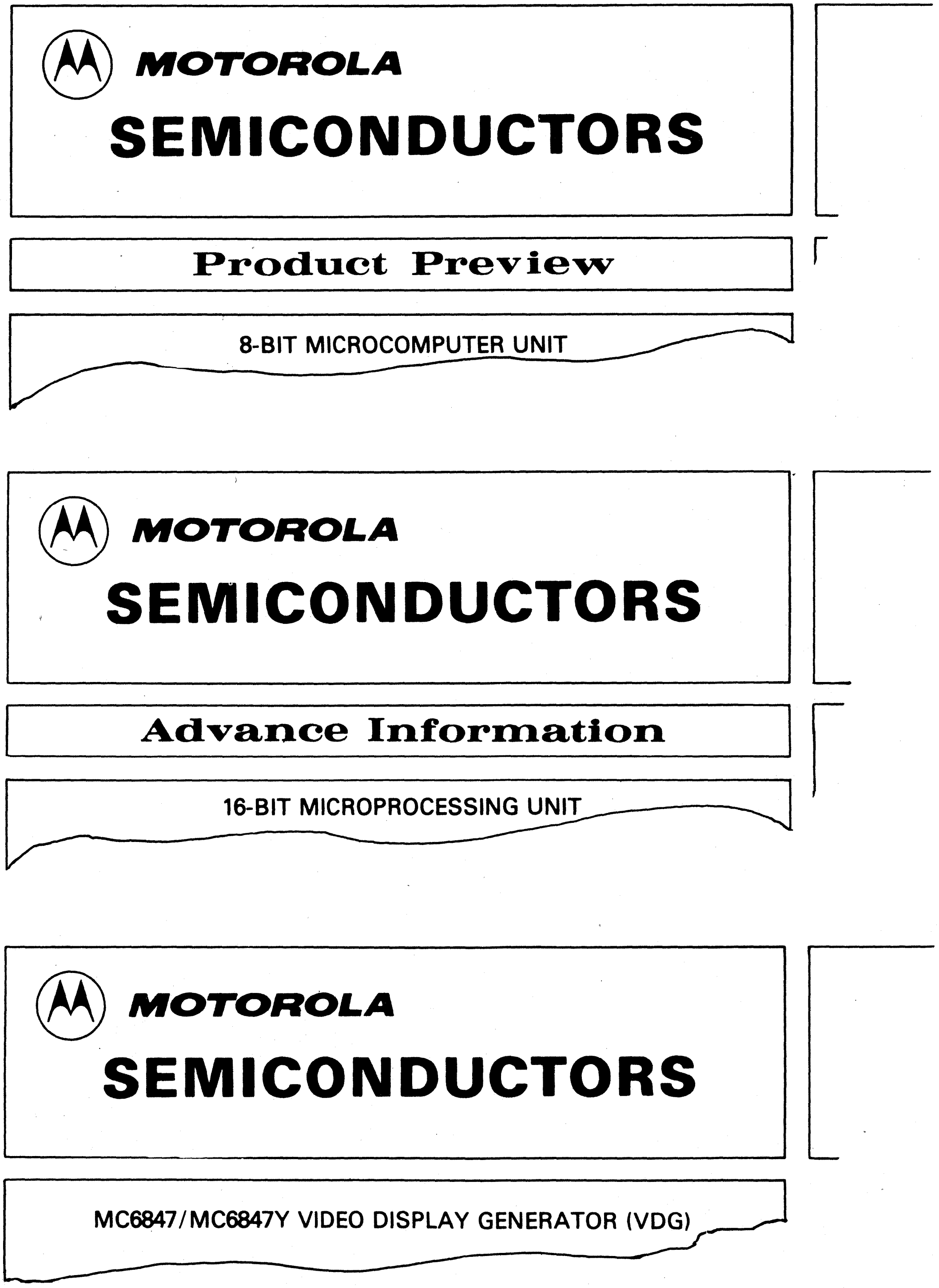
The data sheets that follow are arranged in ascending numerical order disregarding the prefix letters and any speed designation letters.

When device numbers are essentially identical as in the MC6805 and MC6809 families, the alphabetical letters used to distinguish the family members are arranged in ascending alphabetical order. When one data sheet covers closely related devices such as the MC6801, MC6803, and MC6803NR, it is sorted by the lowest part number — in this case, MC6801.

MECHANICAL DATA

The package availability for each device is indicated on the front page of the individual data sheet. Mechanical data for the different package types used is located in a separate chapter of the manual. The data is arranged by package size (pin count) and then package type (plastic, ceramic, etc.).

Figure 4-1. DATA SHEET TYPE DESIGNATIONS



MOTOROLA

SEMICONDUCTORS

Product Preview

8-BIT MICROCOMPUTER UNIT

MOTOROLA

SEMICONDUCTORS

Advance Information

16-BIT MICROPROCESSING UNIT

MOTOROLA

SEMICONDUCTORS

MC6847/MC6847Y VIDEO DISPLAY GENERATOR (VDG)

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.

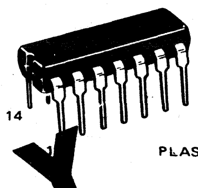
The MC1372 is a companion part to the MC6847 Video Display Generator, providing and accepting the correct dc interconnection levels. This device may also be used as a general-purpose modulator with a variety of video signal generating devices such as video games, test equipment, video tape recorders, 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 Entered Video Signal
- Low Power Dissipation
- Linear Chroma Modulators for High Versatility
- Composite Video Signal Generation Capability
- Ground-Referenced Video Prevents Overmodulation

NTSC ONLY

COLOR TV
VIDEO
MODULATOR CIRCUIT

SILICON MONOLITHIC
INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 646

Pin Connections

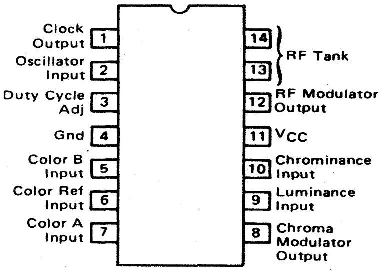
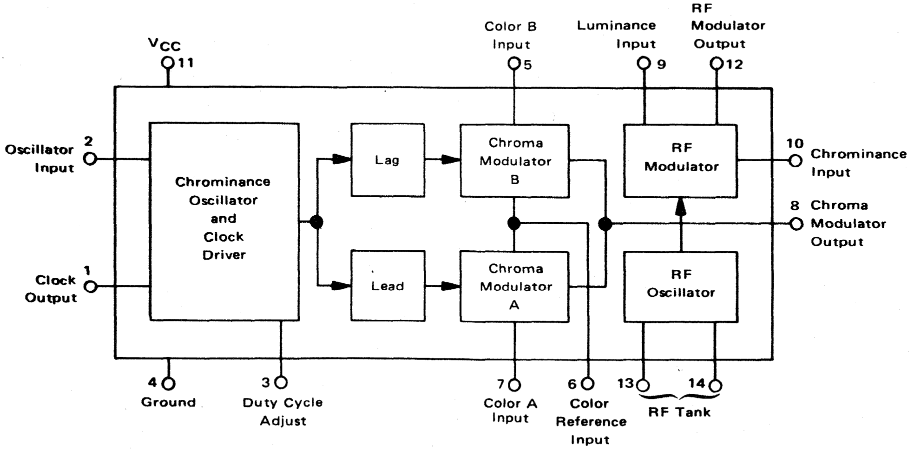


FIGURE 1 – BLOCK DIAGRAM



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

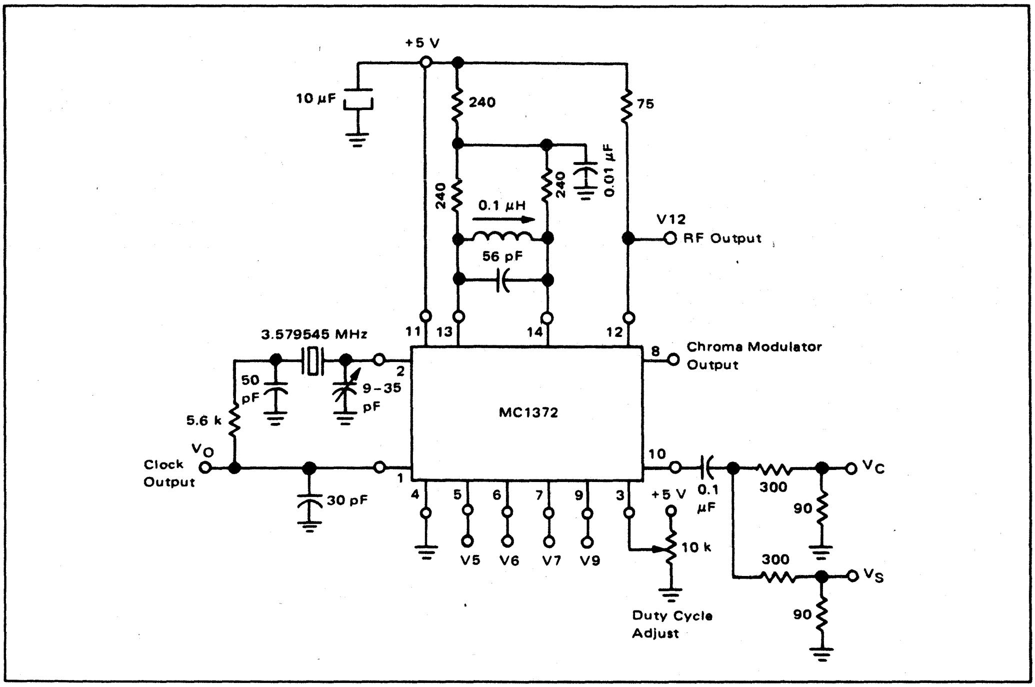


FIGURE 3 – TEST CIRCUIT 2

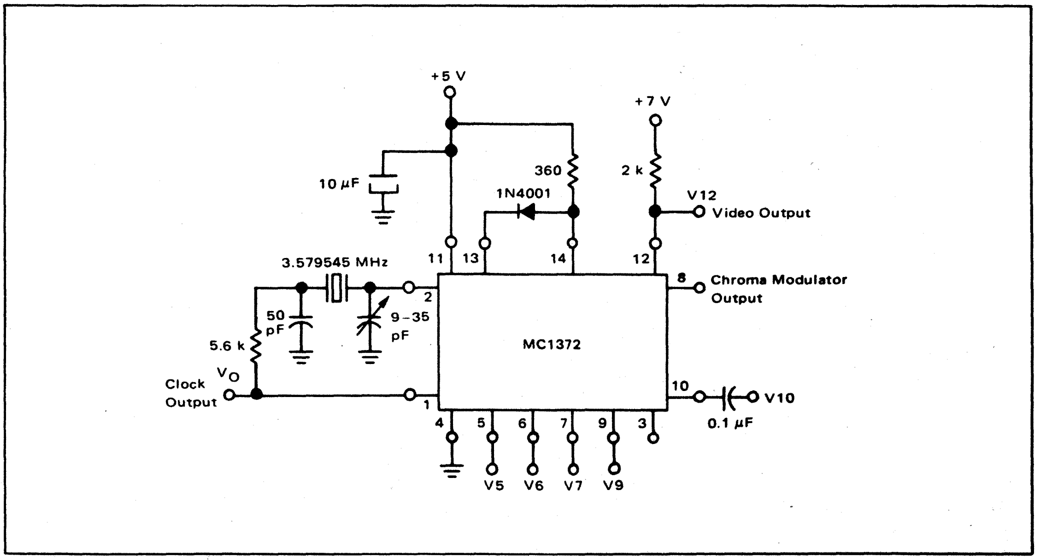
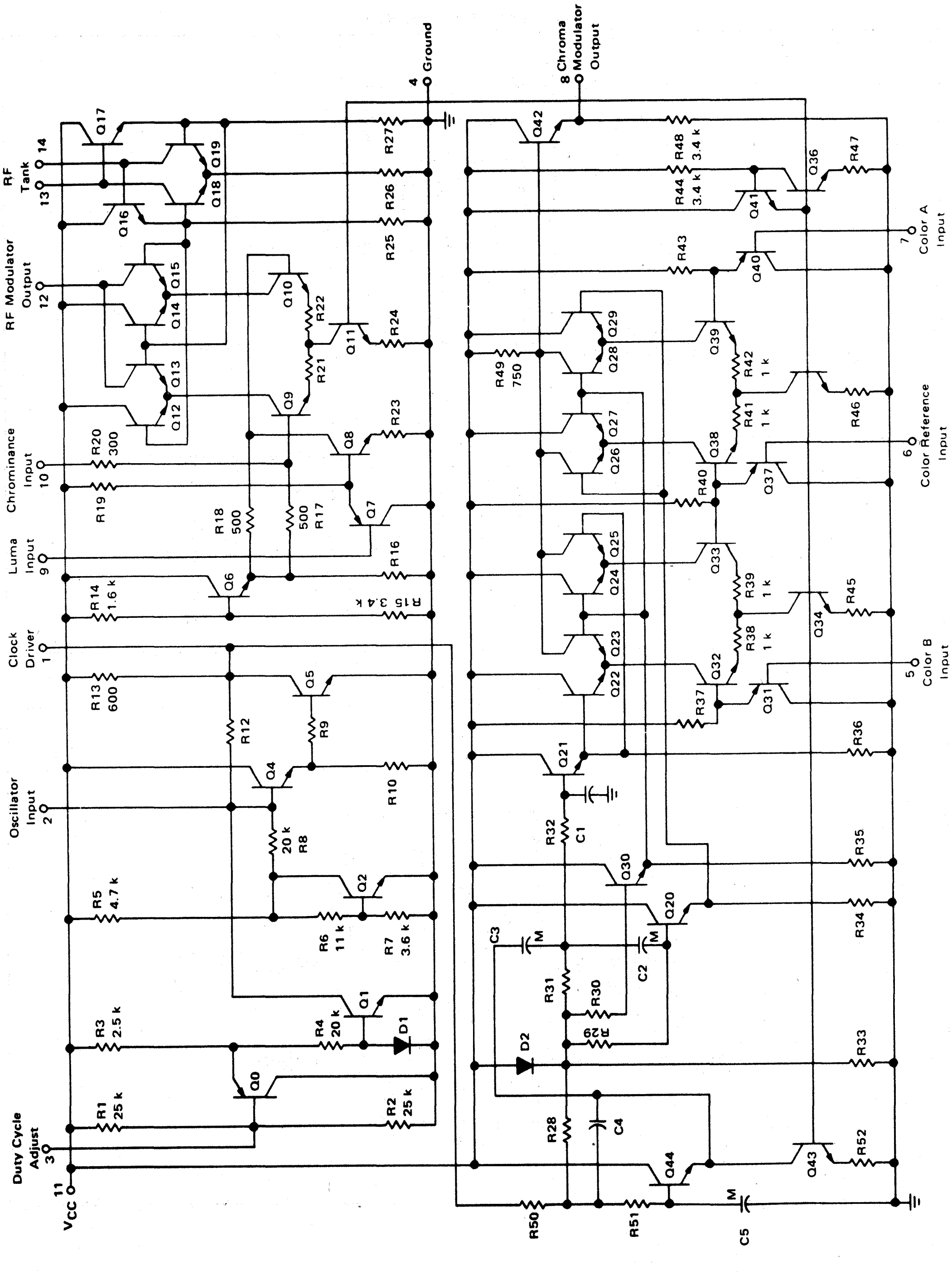


FIGURE 4 – SCHEMATIC DIAGRAM



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°, 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° from the nominal 90°, to provide the 100° phase shift as discussed previously.

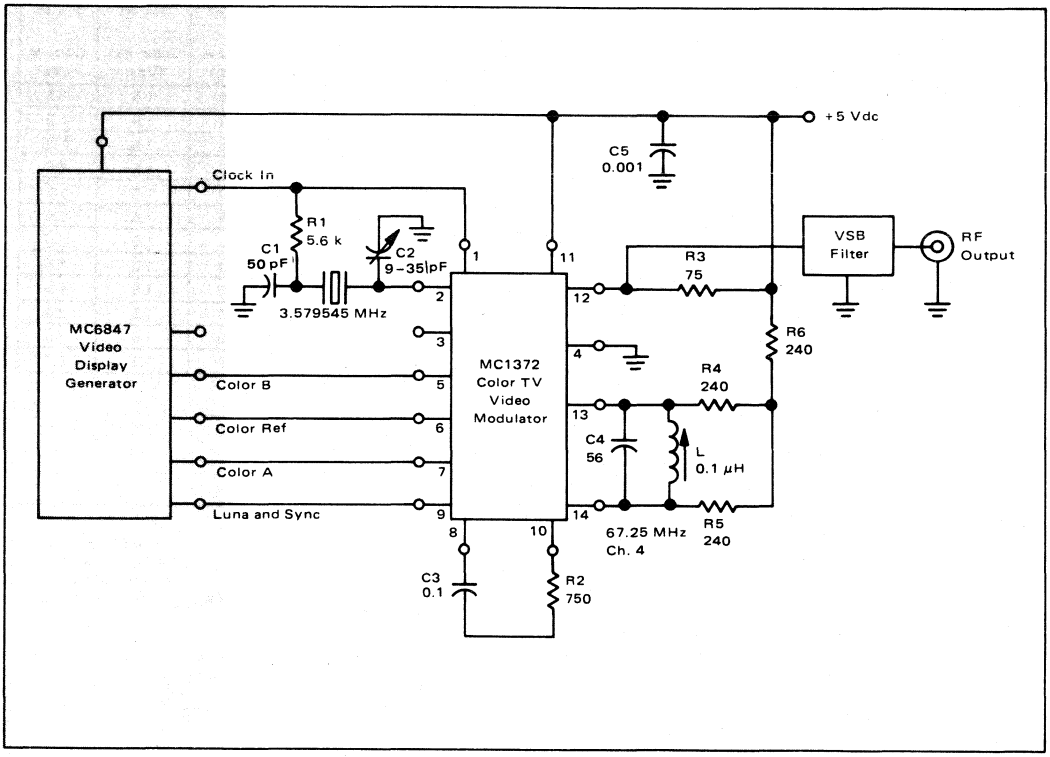
RF Modulator and Oscillator

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

OUTLINE DIMENSIONS

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature can be found from the equation:

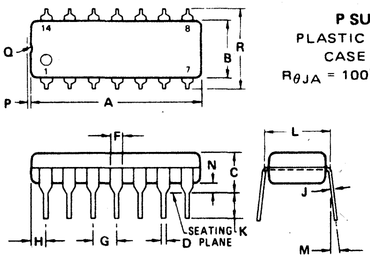
$$P_D(T_A) = \frac{T_J(max) - T_A}{R_{\theta JA}(typ)}$$

where $P_D(T_A)$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst-case operating condition.

$T_J(max)$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA}(typ)$ = Typical Thermal Resistance Junction to Ambient



P SUFFIX
PLASTIC PACKAGE
CASE 646-04
$R_{\theta JA}$ = 100°C/W Typical

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	18.03	19.56	0.710	0.770
B	6.10	6.60	0.240	0.260
C	–	5.08	–	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	–	0.115	–
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	–	0.020	–
R	–	8.26	–	0.325

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. DIMENSION "R" TO BE MEASURED AT THE TOP OF THE LEADS (NOT AT THE TIPS).

MOTOROLA

MC3446A

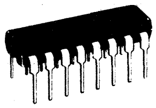
QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVER

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

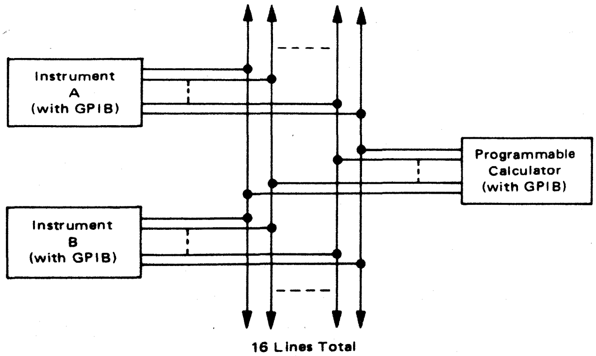
QUAD INTERFACE BUS TRANSCEIVER

SILICON MONOLITHIC INTEGRATED CIRCUIT

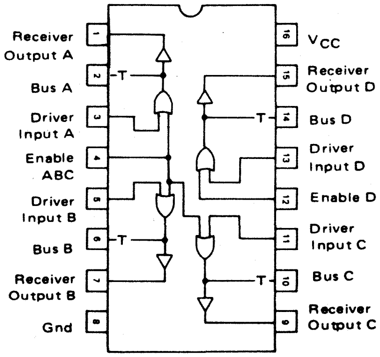


P SUFFIX
PLASTIC PACKAGE
CASE 648

TYPICAL MEASUREMENT SYSTEM APPLICATION



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 $\leqslant V_{CC} \leqslant$ 5.5 V and 0 $\leqslant T_A \leqslant$ 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 ($V_{IH(D)}$= 2.4 V, $V_{BUS} \geqslant$ 5.0 V)	$I_{(BUS)}$	0.7	–	–	mA
($V_{IH(D)}$ = 2.4 V, V_{BUS} = 0.5 V)		-1.3	–	-3.2	
($V_{BUS} \leqslant$ 5.5 V)		–	–	2.5	
(V_{CC} = 0, 0 V $\leqslant V_{BUS} \leqslant$ 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

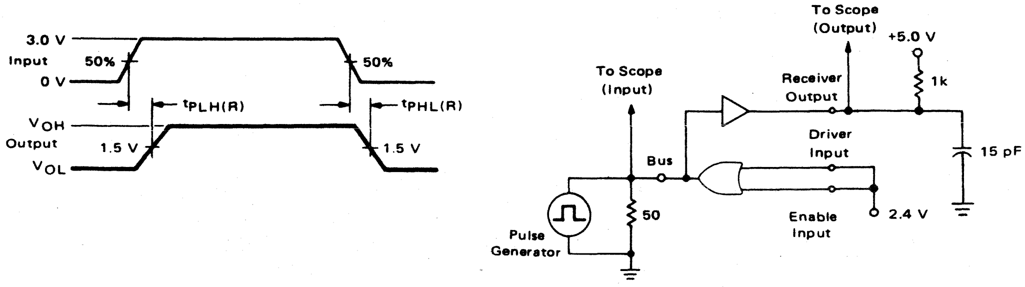
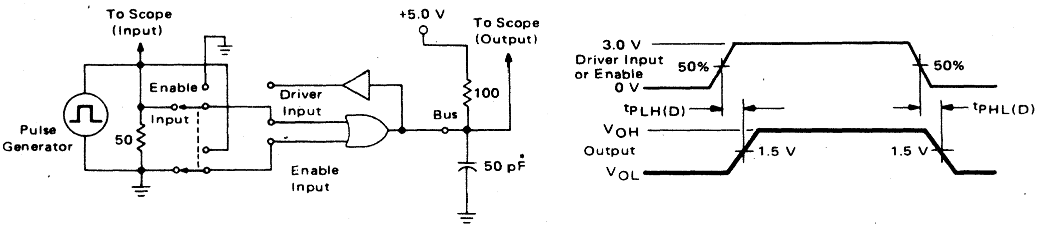


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)



* Includes Probe and Jig Capacitance

FIGURE 3 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

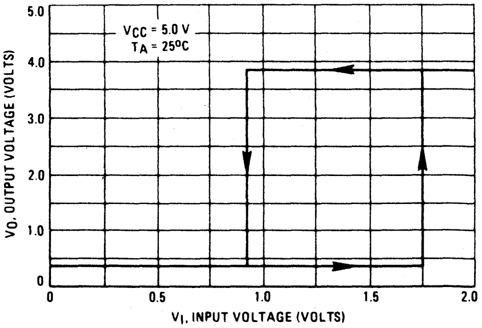
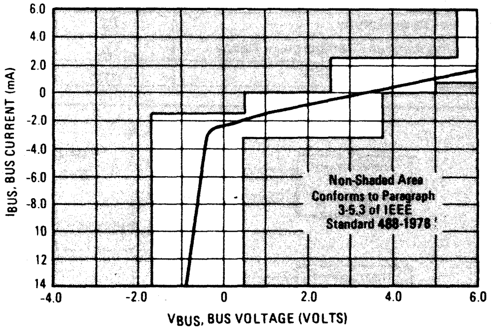


FIGURE 4 – TYPICAL BUS LOAD LINE



MOTOROLA

MC3447

OCTAL BIDIRECTIONAL BUS TRANSCEIVER WITH TERMINATION NETWORKS

SILICON MONOLITHIC INTEGRATED CIRCUIT

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.

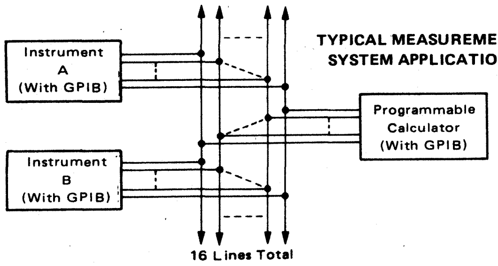
Low power consumption has been achieved by trading a minimum of speed for low current drain on non-critical channels. A fast channel is provided for critical ATN and EOI paths.

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.

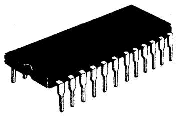
- Low Power – Average Power Supply Current = 30 mA Listening
  75 mA Talking
- Eight Driver/Receiver Pairs
- Three-State Outputs
- High Impedance Inputs
- Receiver Hysteresis – 600 mV (Typ)
- Fast Propagation Times – 15–20 ns (Typ)
- TTL Compatible Receiver Outputs
- Single +5 Volt Supply
- Open Collector 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)

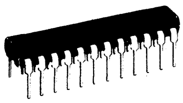
Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C



TYPICAL MEASUREMENT SYSTEM APPLICATION

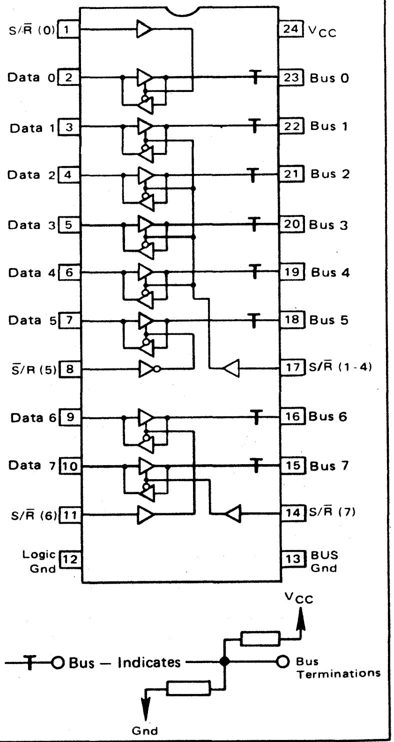


L SUFFIX
CERAMIC PACKAGE
CASE 623



P3 SUFFIX
PLASTIC PACKAGE
CASE 724

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)

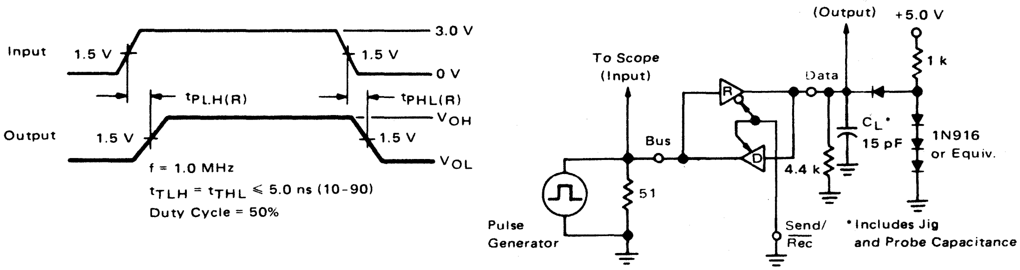


FIGURE 2 – DATA INPUT TO BUS OUTPUT (DRIVER)

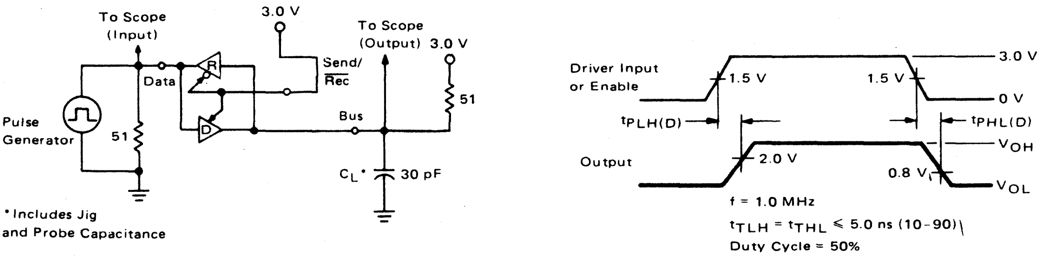


FIGURE 3 – SEND/RECEIVE INPUT TO BUS OUTPUT (DRIVER)

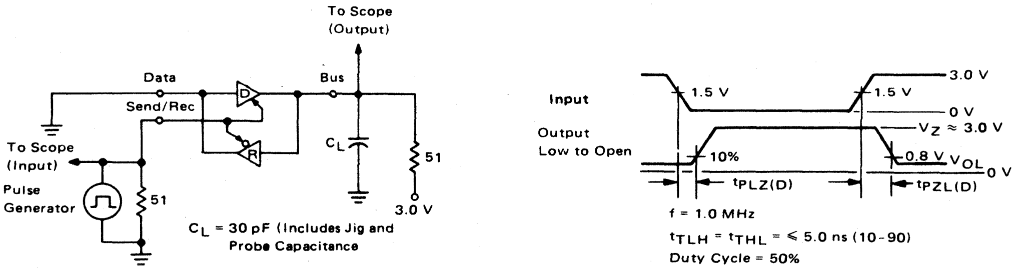
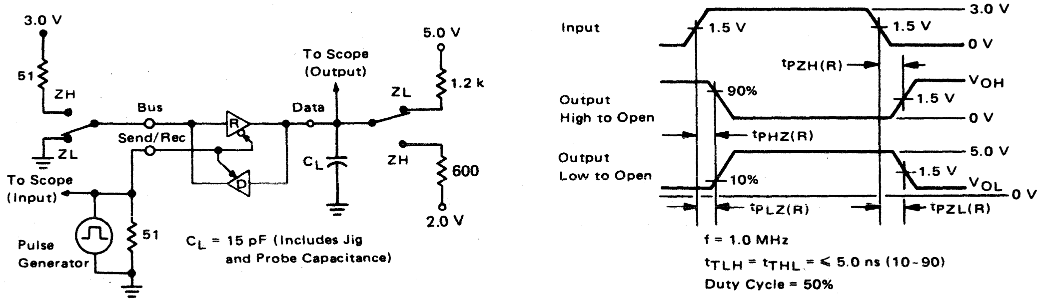


FIGURE 4 – SEND/$\overline{\text{RECEIVE}}$ INPUT TO DATA OUTPUT (RECEIVER)



C_L = 15 pF (Includes Jig and Probe Capacitance)

f = 1.0 MHz

$t_{TLH} = t_{THL} = \leq 5.0$ ns (10–90)

Duty Cycle = 50%

FIGURE 5 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

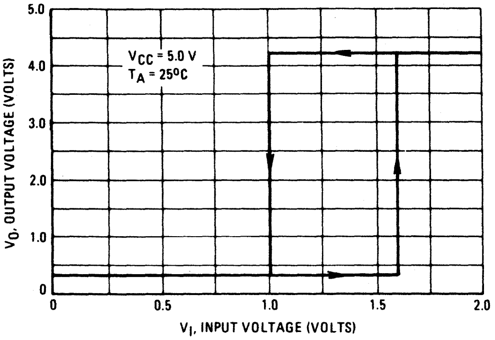


FIGURE 6 – TYPICAL BUS LOAD LINE

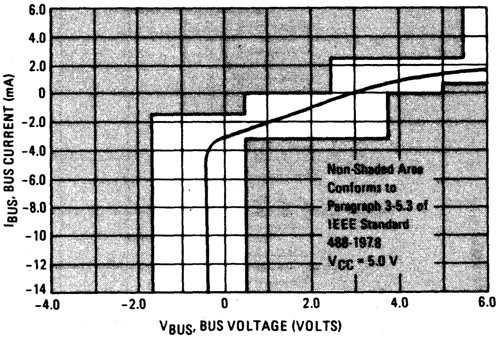


FIGURE 7 – SUGGESTED PRINTED CIRCUIT BOARD LAYOUT USING MC3447s AND MC68488

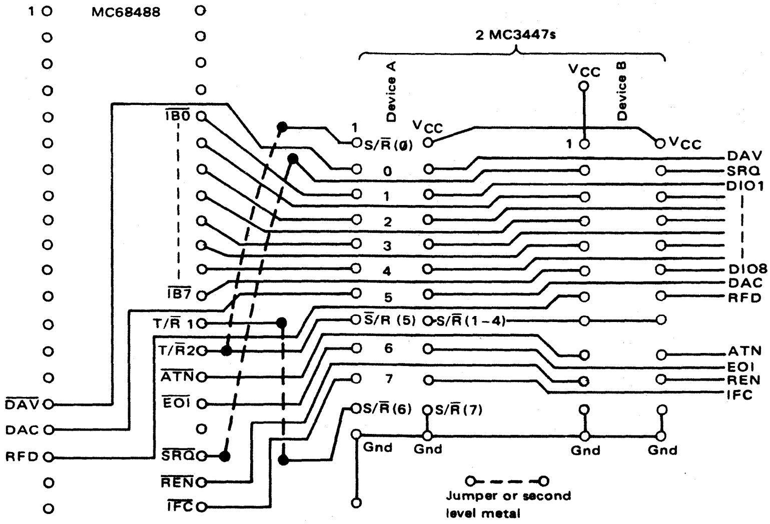
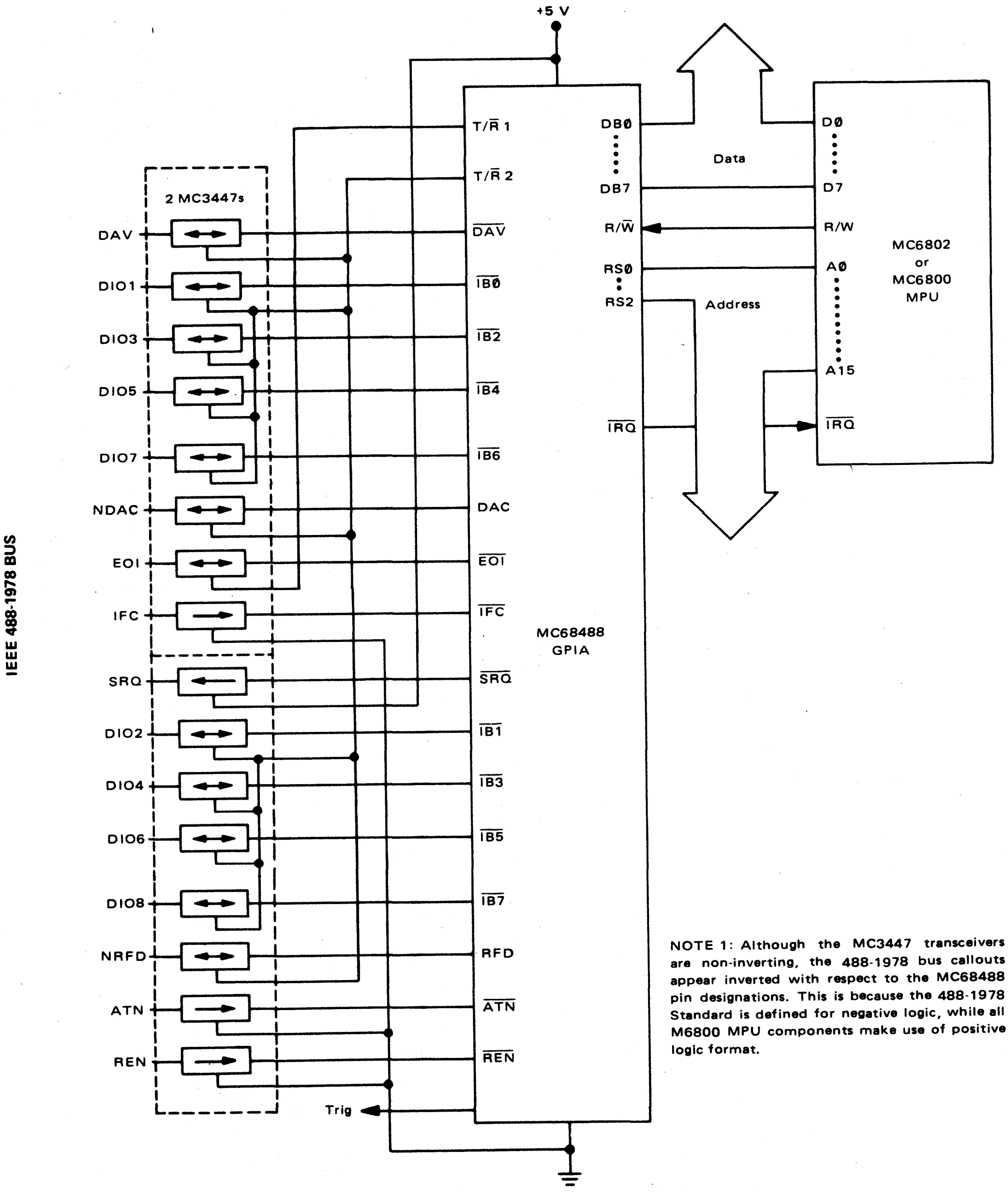


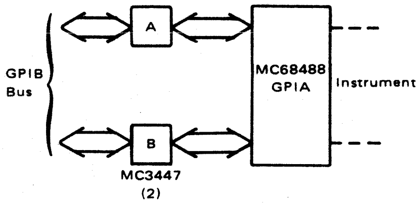
FIGURE 8 – SIMPLE SYSTEM CONFIGURATION



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

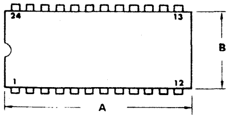
MC68488 Connections		MC3447 Pin Designations					MC68488 Connections	
A	B						A	B
T/$\overline{\text{R}}$ 2	V_{CC}	S/$\overline{\text{R}}$ (0)	1		24	V_{CC}	V_{CC}	V_{CC}
$\overline{\text{DAV}}$	$\overline{\text{SRQ}}$	Data 0 0	2		23	Bus 0	DAV	SRQ
$\overline{\text{IB0}}$	$\overline{\text{IB1}}$	Data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

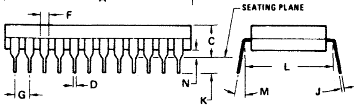


OUTLINE DIMENSIONS

L SUFFIX

CERAMIC PACKAGE
CASE 623-04
$\theta_{JA}(typ) = 53^{\circ}C/W$





DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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	2.29	4.06	0.090	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

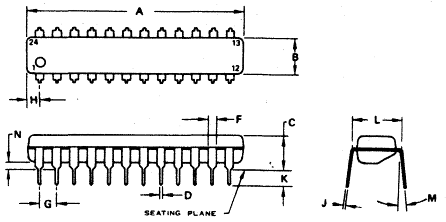
NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL)

P3 SUFFIX

PLASTIC PACKAGE
CASE 724-02
$\theta_{JA}(typ) = 90^{\circ}C/W$

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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°	–	10°
N	0.51	1.02	0.020	0.040

NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION (DIM D).



THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA(Typ)}}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA(Typ)}$ = Typical Thermal Resistance Junction to Ambient

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.

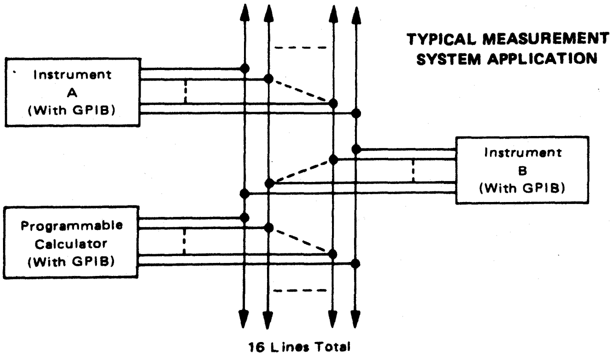
Each driver/receiver pair forms the complete interface between the bus and an instrument. Either the driver or the receiver of each channel is enabled by its corresponding Send/Receive input with the disabled output of the pair forced to a high impedance state. An additional option allows the driver 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)

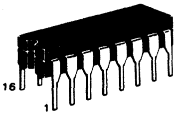
Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C



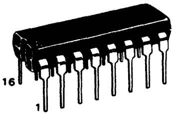
TYPICAL MEASUREMENT SYSTEM APPLICATION

QUAD THREE-STATE BUS TRANSCEIVER WITH TERMINATION NETWORKS

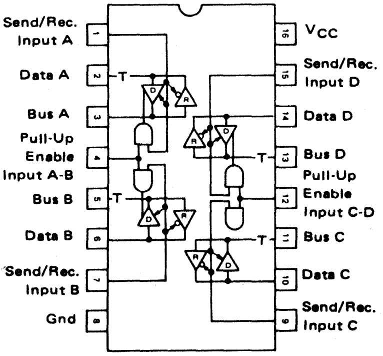
SILICON MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620



P SUFFIX
PLASTIC PACKAGE
CASE 648



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	–	400	600	–	mV
($V_{I(S/R)}$ = 0.8 V)					
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	$V_{OH(R)}$	2.7	–	–	V
($V_{I(S/R)}$ = 0.8 V, $I_{OH(R)}$ = -800 µA, $V_{(BUS)}$ = 2.0 V)					
Receiver Output Voltage – Low Logic State	$V_{OL(R)}$	–	–	0.5	V
($V_{I(S/R)}$ = 0.8 V, $I_{OL(R)}$ = 16 mA, $V_{(BUS)}$ = 0.8 V)					
Receiver Output Short Circuit Current	$I_{OS(R)}$	-15	–	-75	mA
($V_{I(S/R)}$ = 0.8 V, $V_{(BUS)}$ = 2.0 V)					
Driver Input Voltage – High Logic State	$V_{IH(D)}$	2.0	–	–	V
($V_{I(S/R)}$ = 2.0 V)					
Driver Input Voltage – Low Logic State	$V_{IL(D)}$	–	–	0.8	V
($V_{I(S/R)}$ = 2.0 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	$V_{IC(D)}$	–	–	-1.5	V
($V_{I(S/R)}$ = 2.0 V, $I_{IC(D)}$ = -18 mA)					
Driver Output Voltage – High Logic State	$V_{OH(D)}$	2.5	–	–	V
($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V, I_{OH} = -5.2 mA)					
Driver Output Voltage – Low Logic State (Note 1)	$V_{OL(D)}$	–	–	0.5	V
($V_{I(S/R)}$ = 2.0 V, $I_{OL(D)}$ = 48 mA)					
Output Short Circuit Current	$I_{OS(D)}$	-30	–	-120	mA
($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V)					
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)

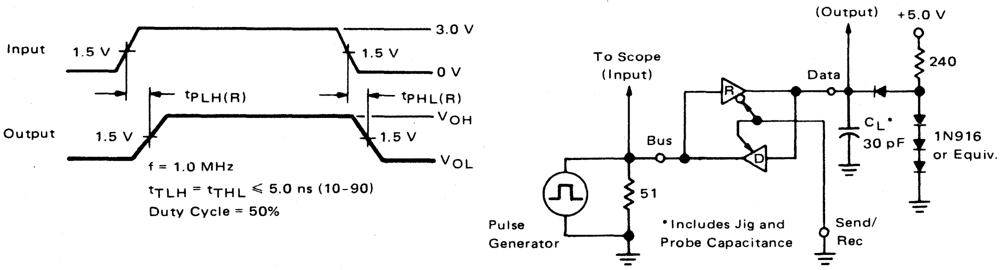


FIGURE 2 – DATA INPUT TO BUS OUTPUT (DRIVER)

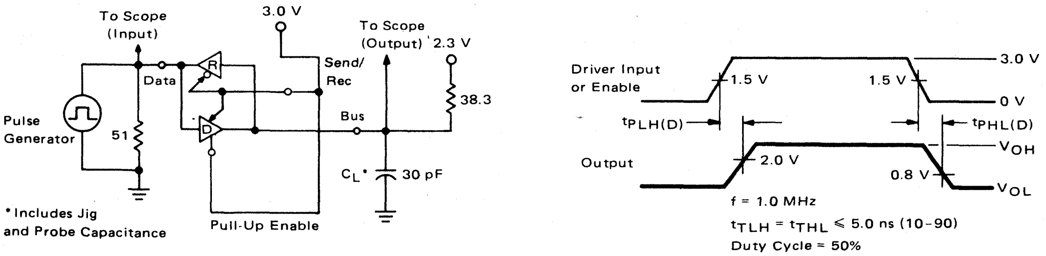


FIGURE 3 – SEND/RECEIVE INPUT TO BUS OUTPUT (DRIVER)

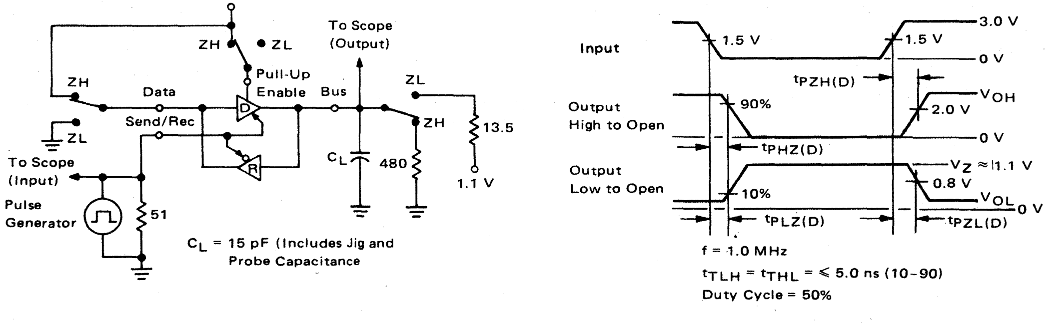
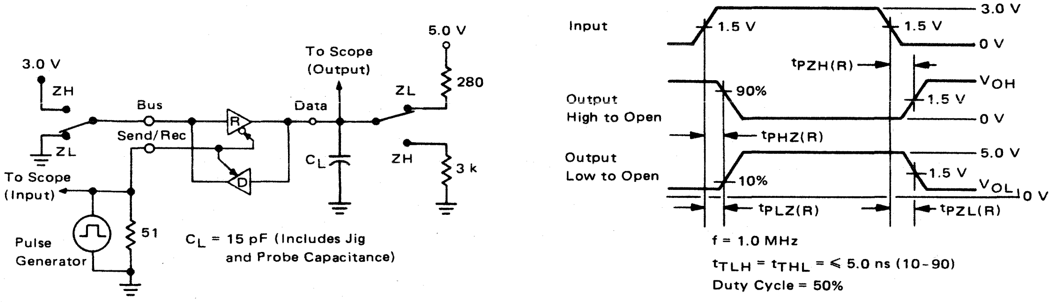


FIGURE 4 — SEND/RECEIVE INPUT TO DATA OUTPUT (RECEIVER)



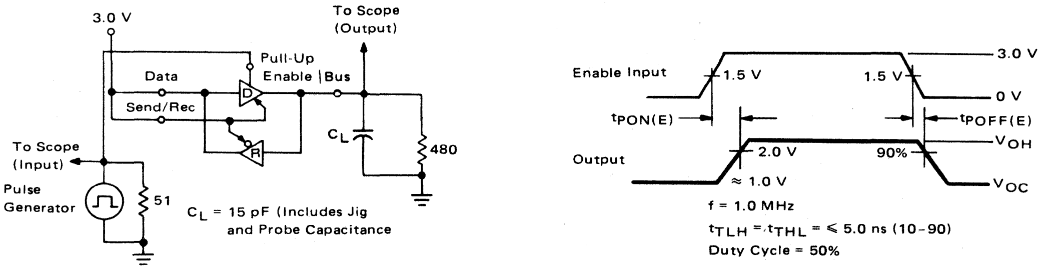
C_L = 15 pF (Includes Jig and Probe Capacitance)

f = 1.0 MHz

$t_{TLH} = t_{THL} = \leqslant 5.0$ ns (10–90)

Duty Cycle = 50%

FIGURE 5 — ENABLE INPUT TO BUS OUTPUT (DRIVER)



C_L = 15 pF (Includes Jig and Probe Capacitance)

f = 1.0 MHz

$t_{TLH} = t_{THL} = \leqslant 5.0$ ns (10–90)

Duty Cycle = 50%

FIGURE 6 — TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

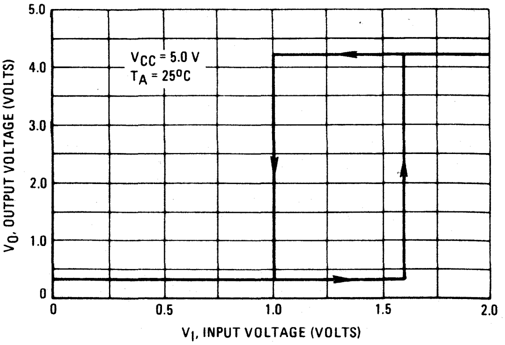


FIGURE 7 — TYPICAL BUS LOAD LINE

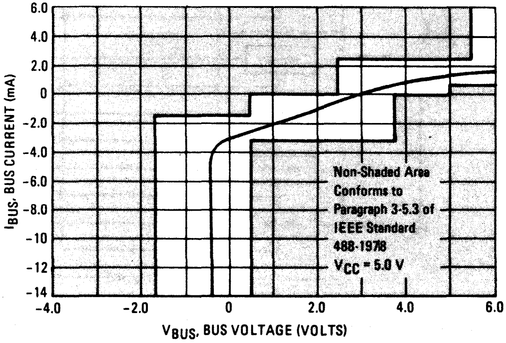
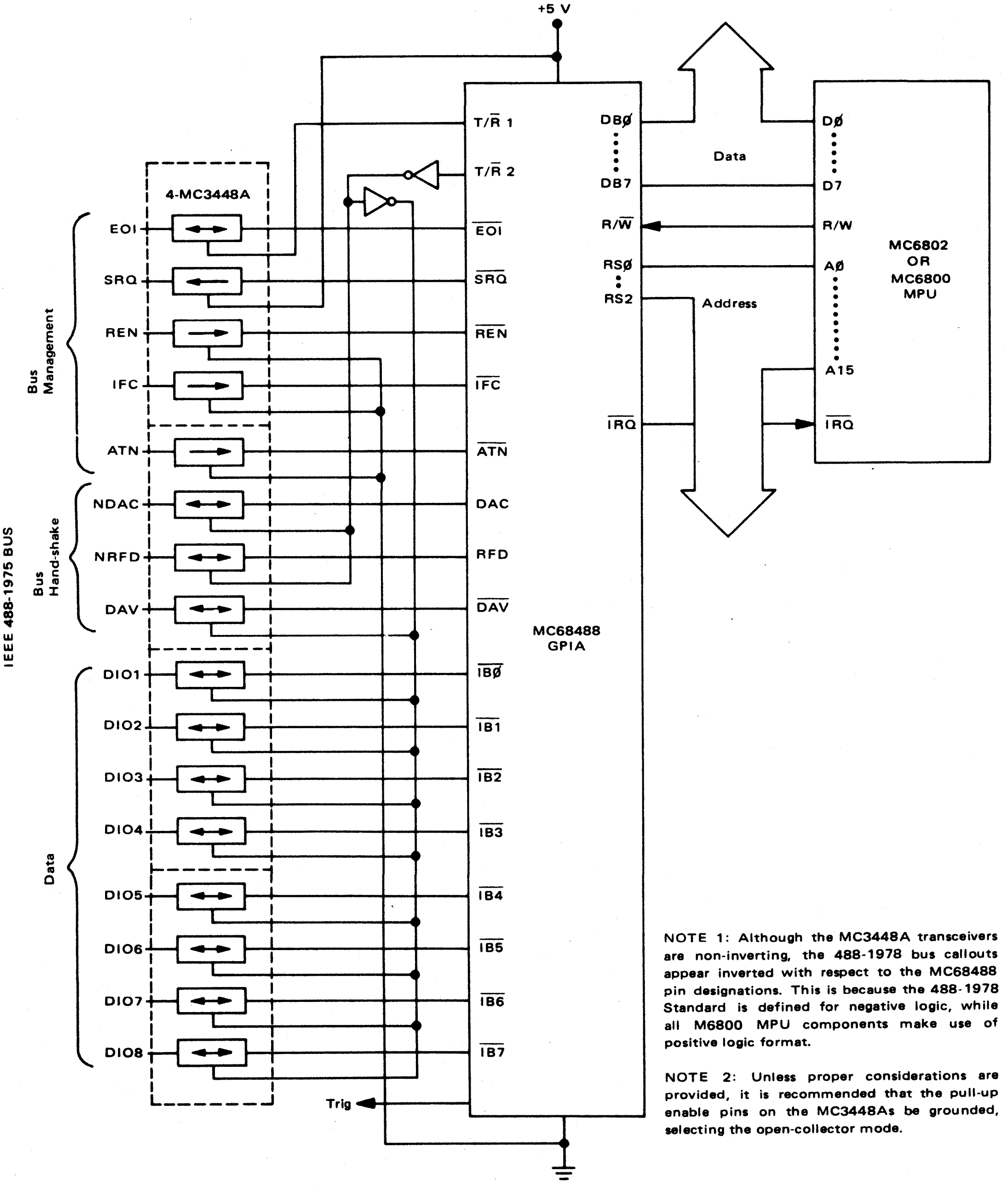


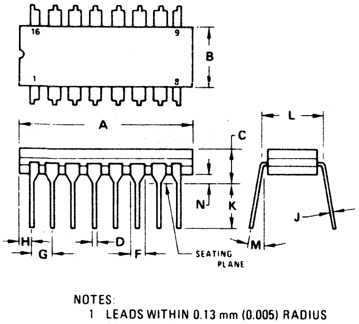
FIGURE 8 — SIMPLE SYSTEM CONFIGURATION



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.

OUTLINE DIMENSIONS

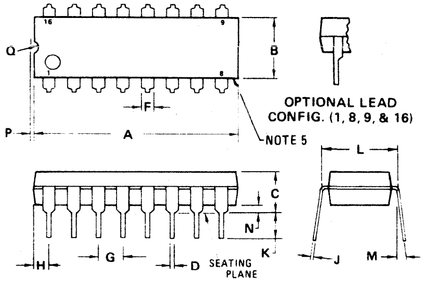


NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

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

CASE 620-02

$R_{\theta JA}$ = 60°C/W (Typical)



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.

CASE 648-05

$R_{\theta JA}$ = 100°C/W (Typical)

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA(Typ)}}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA(Typ)}$ = Typical Thermal Resistance Junction to Ambient

MOTOROLA

MC3482A/MC6882A
MC3482B/MC6882B

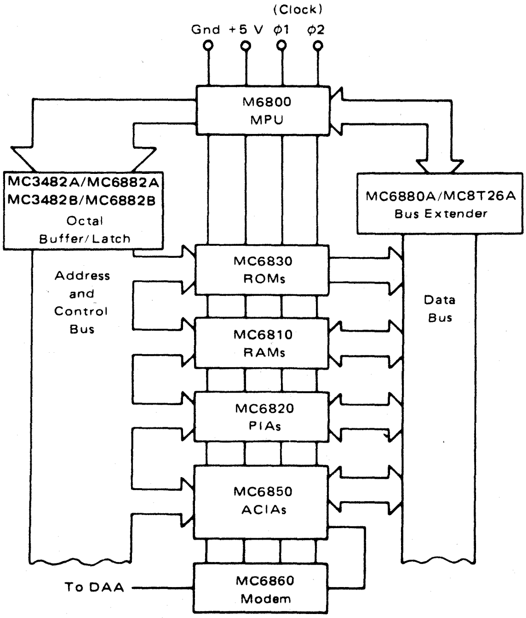
This device may be ordered under either of the above type numbers.

OCTAL THREE-STATE BUFFER/LATCH

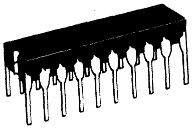
This series of devices combines four features usually found desirable in bus-oriented systems: 1) High impedance logic inputs insure that these devices do not seriously load the bus; 2) Three-state logic configuration allows buffers not being utilized to be effectively removed from the bus; 3) Schottky 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

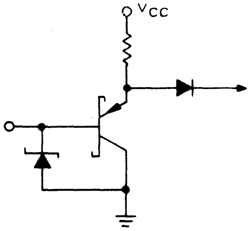


OCTAL THREE-STATE BUFFER/LATCH

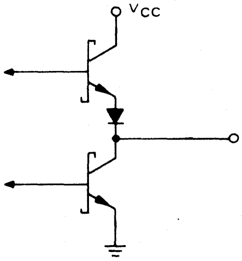


L SUFFIX
CASE 732

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

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	T_J		°C
Ceramic Package		175	

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 MC3482A/MC6882A (V_{CC} = 5.25 V) 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.)

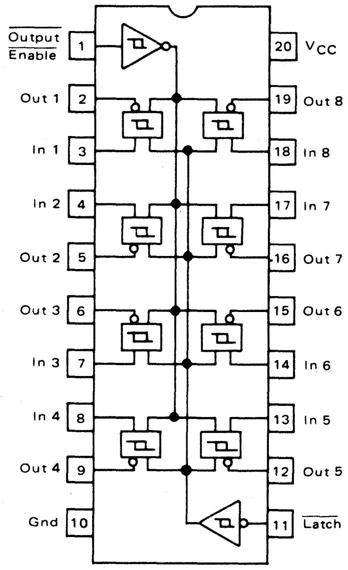
Characteristics	Symbol	MC3482A/ MC6882A			MC3482B/ MC6882B			Unit
		Min	Typ	Max	Min	Typ	Max	
Propagation Delay Times								ns
Data to Output								
Low to High	$t_{PLH(D)}$							
C_L = 50 pF		4.0	9.0	16	4.0	9.0	16	
C_L = 250 pF		–	12	20	–	12	20	
C_L = 375 pF		–	14	22	–	14	22	
C_L = 500 pF		10	16	24	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								ns
(C_L = 20 pF)								
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

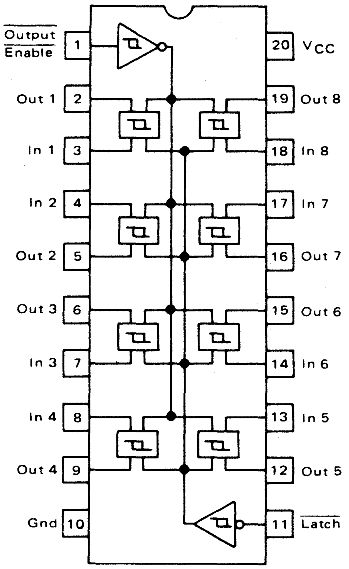
PIN CONNECTIONS AND TRUTH TABLES

MC3482A/MC6882A



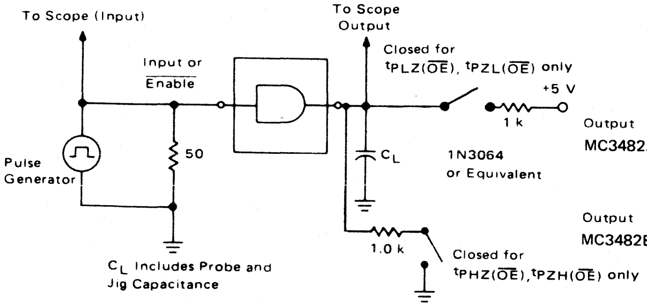
$\overline{\text{Output}}$ Enable	$\overline{\text{Latch}}$	Input	Output
0	1	0	1
0	1	1	0
0	0	X	Q_0
1	X	X	Z

MC3482B/MC6882B



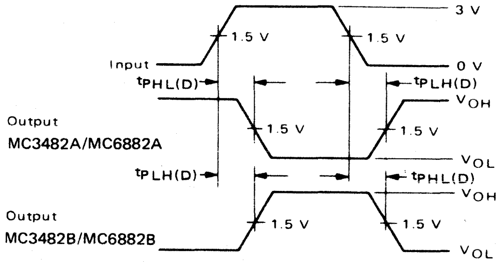
$\overline{\text{Output}}$ Enable	$\overline{\text{Latch}}$	Input	Output
0	1	0	0
0	1	1	1
0	0	X	Q_0
1	X	X	Z

FIGURE 1 – TEST CIRCUIT FOR SWITCHING CHARACTERISTICS



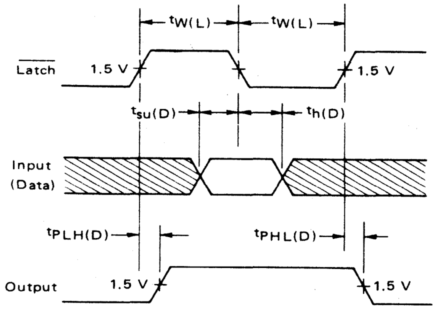
C_L Includes Probe and Jig Capacitance

FIGURE 2 – WAVEFORMS FOR PROPAGATION DELAY TIMES DATA TO OUTPUT



Input Pulse Conditions
$t_{THL} = t_{TLH} < 5$ ns
$f = 1.0$ MHz

FIGURE 3 – WAVE FORMS FOR AC SETUP AND LATCH DISABLE TO OUTPUT DELAY



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		INCHES	
	MIN	MAX	MIN	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

CASE 732-03

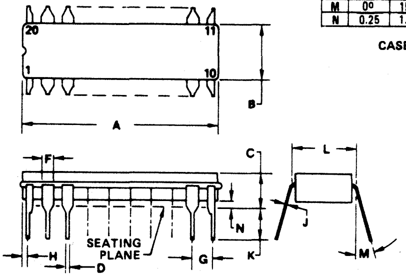
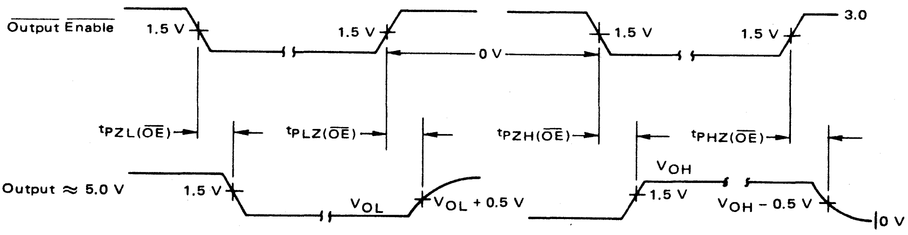


FIGURE 4 – WAVEFORMS FOR PROPAGATION DELAY TIMES – $\overline{\text{OUTPUT ENABLE}}$ TO OUTPUT



MOTOROLA

MC6800
(1.0 MHz)
MC68A00
(1.5 MHz)
MC68B00
(2.0 MHz)

MOS

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

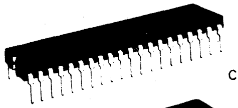
MICROPROCESSOR

8-BIT MICROPROCESSING UNIT (MPU)

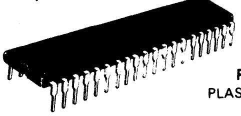
The MC6800 is a monolithic 8-bit microprocessor forming the central control function for Motorola's M6800 family. Compatible with TTL, the MC6800, as with all M6800 system parts, requires only one +5.0-volt power supply, and no external TTL devices for bus interface.

The MC6800 is capable of addressing 64K bytes of memory with its 16-bit address lines. The 8-bit data bus is bidirectional as well as three-state, making direct memory addressing and multiprocessing applications realizable.

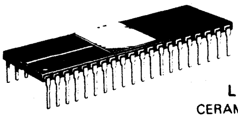
- 8-Bit Parallel Processing
- Bidirectional Data Bus
- 16-Bit Address Bus — 64K Bytes of Addressing
- 72 Instructions — Variable Length
- Seven Addressing Modes — Direct, Relative, Immediate, Indexed, Extended, Implied and Accumulator
- Variable Length Stack
- Vectored Restart
- Maskable Interrupt Vector
- Separate Non-Maskable Interrupt — Internal Registers Saved in Stack
- Six Internal Registers — Two Accumulators, Index Register, Program Counter, Stack Pointer and Condition Code Register
- Direct Memory Addressing (DMA) and Multiple Processor Capability
- Simplified Clocking Characteristics
- Clock Rates as High as 2.0 MHz
- Simple Bus Interface Without TTL
- Halt and Single Instruction Execution Capability



S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

MAXIMUM RATINGS

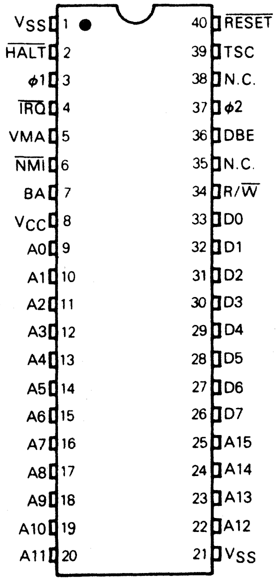
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6800, MC68A00, MC68B00 MC6800C, MC68A00C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL RESISTANCE

Rating	Symbol	Value	Unit
Plastic Package Cerdip Package Ceramic Package	θ_{JA}	100 60 50	°C/W

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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

PIN ASSIGNMENT



POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

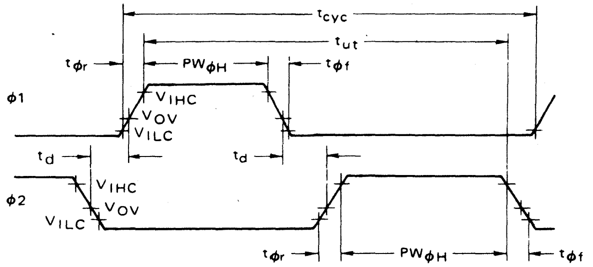
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc, ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
	$\phi1, \phi2$	V_{IHC}	$V_{CC}-0.6$	–	$V_{CC}+0.3$	
Input Low Voltage	Logic	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
	$\phi1, \phi2$	V_{ILC}	$V_{SS}-0.3$	–	$V_{SS}+0.4$	
Input Leakage Current						
(V_{in} = 0 to 5.25 V, V_{CC} = Max)	Logic	I_{in}	–	1.0	2.5	μA
(V_{in} = 0 to 5.25 V, V_{CC} = 0 V to 5.25 V)	$\phi1, \phi2$		–	–	100	
Three-State Input Leakage Current	D0-D7	I_{IZ}	–	2.0	10	μA
(V_{in} = 0.4 to 2.4 V, V_{CC} = Max)	A0-A15, $R/\overline{W}$		–	–	100	
Output High Voltage						
($I_{Load} = -205$ μA, V_{CC} = Min)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
($I_{Load} = -145$ μA, V_{CC} = Min)	A0-A15, $R/\overline{W}$, VMA		$V_{SS}+2.4$	–	–	
($I_{Load} = -100$ μA, V_{CC} = Min)	BA		$V_{SS}+2.4$	–	–	
Output Low Voltage (I_{Load} = 1.6 mA, V_{CC} = Min)		V_{OL}	–	–	$V_{SS}+0.4$	V
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	0.5	1.0	W
Capacitance						
(V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\phi1$	C_{in}	–	25	35	pF
	$\phi2$		–	45	70	
	D0-D7		–	10	12.5	
	Logic Inputs		–	6.5	10	
	A0-A15, $R/\overline{W}$, VMA	C_{out}	–	–	12	pF

CLOCK TIMING (V_{CC} = 5.0 V, ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Frequency of Operation	MC6800	f	0.1	–	1.0	MHz
	MC68A00		0.1	–	1.5	
	MC68B00		0.1	–	2.0	
Cycle Time (Figure 1)	MC6800	t_{cyc}	1.000	–	10	μs
	MC68A00		0.666	–	10	
	MC68B00		0.500	–	10	
Clock Pulse Width	$\phi1, \phi2$ – MC6800	$PW_{\phi H}$	400	–	9500	ns
(Measured at $V_{CC}-0.6$ V)	$\phi1, \phi2$ – MC68A00		230	–	9500	
	$\phi1, \phi2$ – MC68B00		180	–	9500	
Total $\phi1$ and $\phi2$ Up Time	MC6800	t_{ut}	900	–	–	ns
	MC68A00		600	–	–	
	MC68B00		440	–	–	
Rise and Fall Time (Measured between $V_{SS}+0.4$ and $V_{CC}-0.6$)		t_r, t_f	–	–	100	ns
Delay Time or Clock Separation (Figure 1)						
(Measured at $V_{OV} = V_{SS}+0.6$ V@$t_r = t_f \leq 100$ ns)		t_d	0	–	9100	ns
(Measured at $V_{OV} = V_{SS}+1.0$ V@$t_r = t_f \leq 35$ ns)			0	–	9100	

FIGURE 1 — CLOCK TIMING WAVEFORM



READ/WRITE TIMING (Reference Figures 2 through 6, 8, 9, 11, 12 and 13)

Characteristic	Symbol	MC6800			MC68A00			MC68B00			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Address Delay											
C = 90 pF	t_{AD}	–	–	270	–	–	180	–	–	150	ns
C = 30 pF		–	–	250	–	–	165	–	–	135	
Peripheral Read Access Time $t_{acc} = t_{ut} - (t_{AD} + t_{DSR})$	t_{acc}	605	–	–	400	–	–	290	–	–	ns
Data Setup Time (Read)	t_{DSR}	100	–	–	60	–	–	40	–	–	ns
Input Data Hold Time	t_H	10	–	–	10	–	–	10	–	–	ns
Output Data Hold Time	t_H	10	25	–	10	25	–	10	25	–	ns
Address Hold Time (Address, R/$\overline{W}$, VMA)	t_{AH}	30	50	–	30	50	–	30	50	–	ns
Enable High Time for DBE Input	t_{EH}	450	–	–	280	–	–	220	–	–	ns
Data Delay Time (Write)	t_{DDW}	–	–	225	–	–	200	–	–	160	ns
Processor Controls											ns
Processor Control Setup Time	t_{PCS}	200	–	–	140	–	–	110	–	–	
Processor Control Rise and Fall Time	t_{PCr}, t_{PCf}	–	–	100	–	–	100	–	–	100	
Bus Available Delay	t_{BA}	–	–	250	–	–	165	–	–	135	
Three-State Enable	t_{TSE}	–	–	40	–	–	40	–	–	40	
Three-State Delay	t_{TSD}	–	–	270	–	–	270	–	–	220	
Data Bus Enable Down Time During $\phi 1$ Up Time	t_{DBE}	150	–	–	120	–	–	75	–	–	
Data Bus Enable Rise and Fall Times	t_{DBEr}, t_{DBEf}	–	–	25	–	–	25	–	–	25	

FIGURE 2 — READ DATA FROM MEMORY OR PERIPHERALS

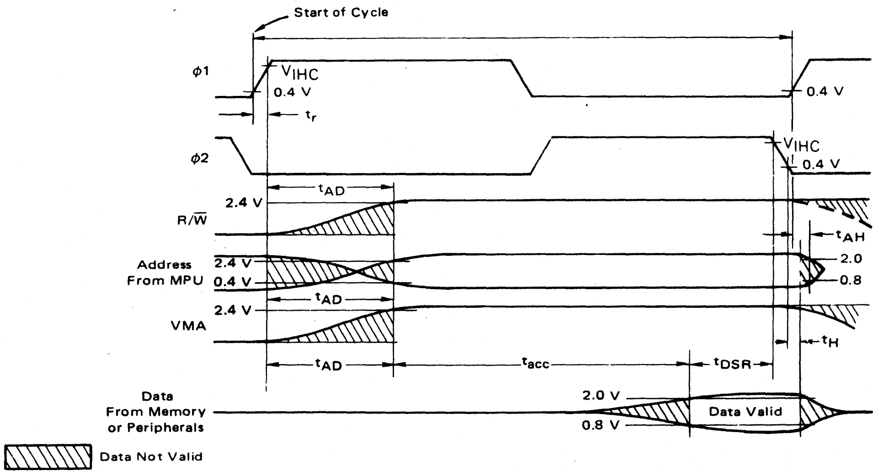


FIGURE 3 — WRITE IN MEMORY OR PERIPHERALS

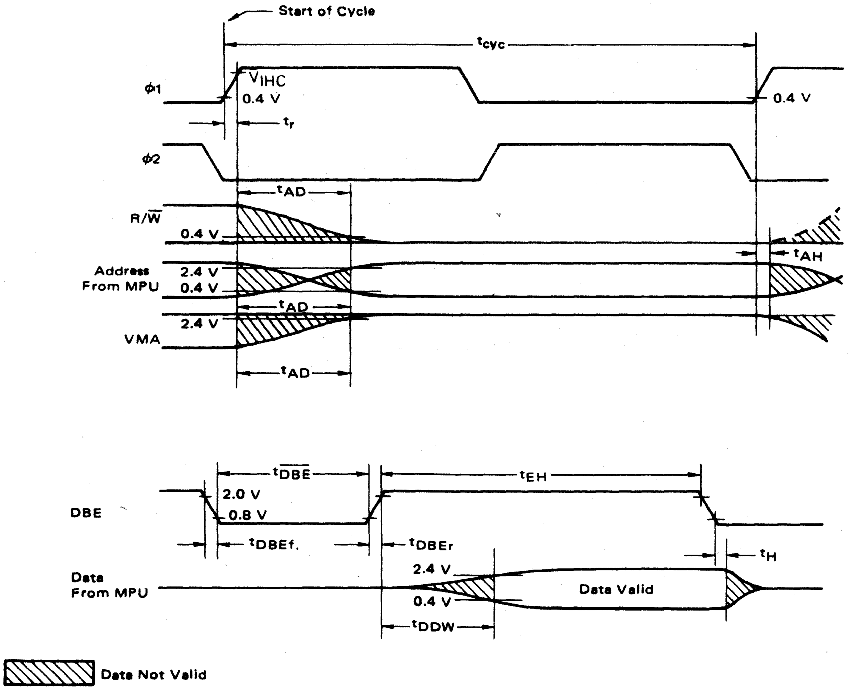


FIGURE 4 — TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING (T_{DDW})

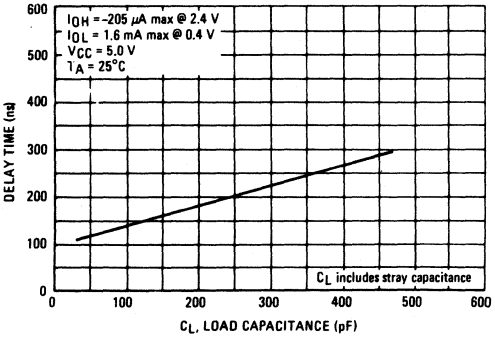
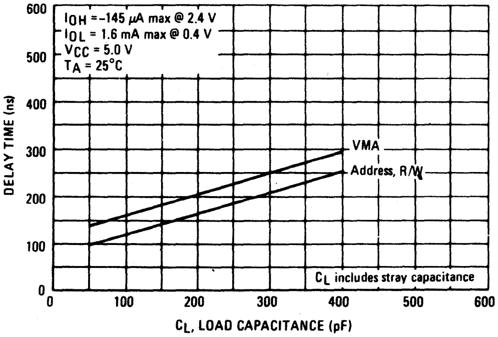


FIGURE 5 — TYPICAL READ/WRITE, VMA, AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING (T_{AD})



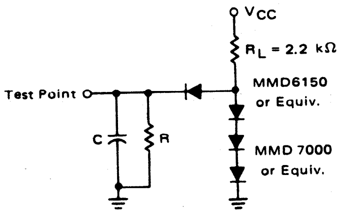


FIGURE 6 — BUS TIMING TEST LOADS

C = 130 pF for D0-D7, E
= 90 pF for A0-A15, R/$\overline{W}$, and VMA
(Except t_{AD2})
= 30 pF for A0-A15, R/$\overline{W}$, and VMA
(t_{AD2} only)
= 30 pF for BA

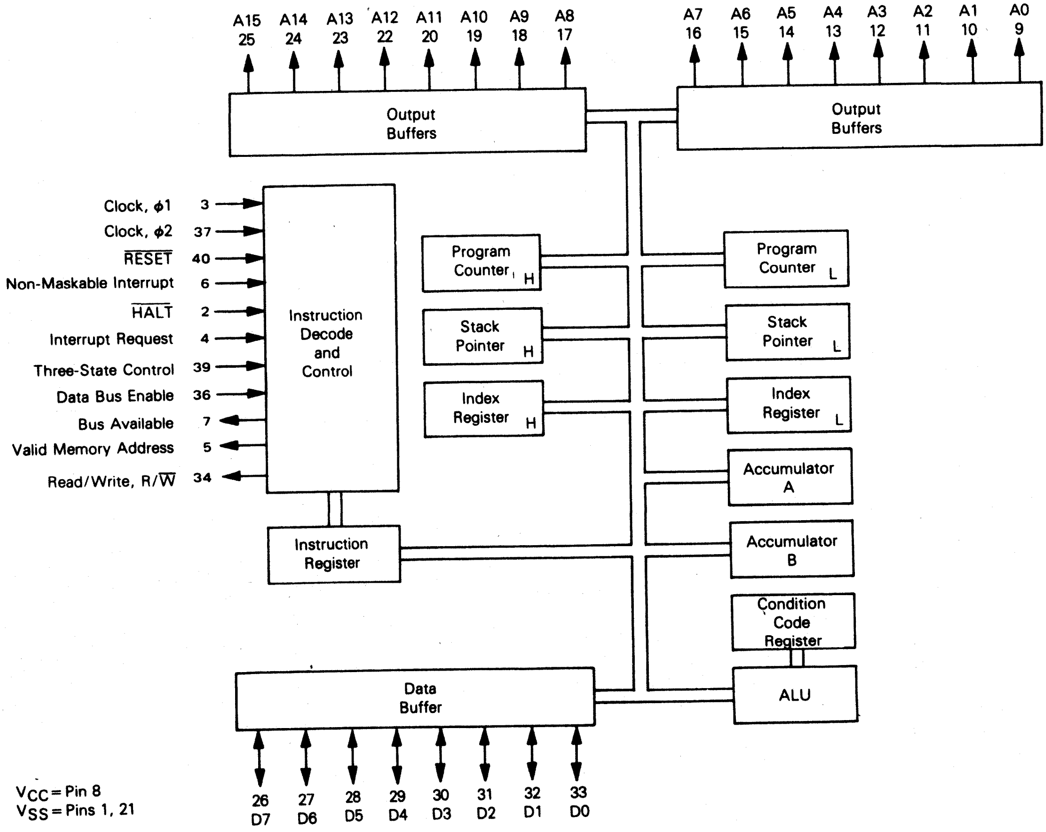
R = 11.7 kΩ for D0-D7
= 16.5 kΩ for A0-A15, R/$\overline{W}$, and VMA
= 24 kΩ for BA

TEST CONDITIONS

The dynamic test load for the Data Bus is 130 pF and one standard TTL load as shown. The Address, R/$\overline{W}$, and VMA outputs are tested under two conditions to allow optimum operation in both buffered and unbuffered systems. The resistor (R) is chosen to insure specified load currents during V_{OH} measurement.

Nctice that the Data Bus lines, the Address lines, the Interrupt Request line, and the DBE line are all specified and tested to guarantee 0.4 V of dynamic noise immunity at both "1" and "0" logic levels.

FIGURE 7 — EXPANDED BLOCK DIAGRAM



MPU SIGNAL DESCRIPTION

Proper operation of the MPU requires that certain control and timing signals be provided to accomplish specific functions and that other signal lines be monitored to determine the state of the processor.

Clocks Phase One and Phase Two (ϕ1, ϕ2) — Two pins are used for a two-phase non-overlapping clock that runs at the V_{CC} voltage level.

Figure 1 shows the microprocessor clocks. The high level is specified at V_{IHC} and the low level is specified at V_{ILC}. The allowable clock frequency is specified by f (frequency). The minimum ϕ1 and ϕ2 high level pulse widths are specified by $PW_{\phi H}$ (pulse width high time). To guarantee the required access time for the peripherals, the clock up time, t_{ut}, is specified. Clock separation, t_d, is measured at a maximum voltage of V_{OV} (overlap voltage). This allows for a multitude of clock variations at the system frequency rate.

Address Bus (A0-A15) — Sixteen pins are used for the address bus. The outputs are three-state bus drivers capable of driving one standard TTL load and 90 pF. When the output is turned off, it is essentially an open circuit. This permits the MPU to be used in DMA applications. Putting TSC in its high state forces the Address bus to go into the three-state mode.

Data Bus (D0-D7) — Eight pins are used for the data bus. It is bidirectional, transferring data to and from the memory and peripheral devices. It also has three-state output buffers capable of driving one standard TTL load and 130 pF. Data Bus is placed in the three-state mode when DBE is low.

Data Bus Enable (DBE) — This level sensitive input is the three-state control signal for the MPU data bus and will enable the bus drivers when in the high state. This input is TTL compatible; however in normal operation, it would be driven by the phase two clock. During an MPU read cycle, the data bus drivers will be disabled internally. When it is desired that another device control the data bus, such as in Direct Memory Access (DMA) applications, DBE should be held low.

If additional data setup or hold time is required on an MPU write, the DBE down time can be decreased, as shown in Figure 3 (DBE$\neq\phi$2). The minimum down time for DBE is t_{DBE} as shown. By skewing DBE with respect to E, data setup or hold time can be increased.

Bus Available (BA) — The Bus Available signal will normally be in the low state; when activated, it will go to the high state indicating that the microprocessor has stopped and that the address bus is available. This will occur if the $\overline{\text{HALT}}$ line is in the low state or the processor is in the WAIT state as a result of the execution of a WAIT instruction. At such time, all three-state output drivers will go to their off state and other outputs to their normally inactive level. The processor is removed from the WAIT state by the occurrence of a maskable (mask bit $I=0$) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF. If TSC is in the high state, Bus Available will be low.

Read/Write (R/$\overline{\text{W}}$) — This TTL compatible output signals the peripherals and memory devices wether the MPU is in a Read (high) or Write (low) state. The normal standby state of this signal is Read (high). Three-State Control going high will turn Read/Write to the off (high impedance) state. Also, when the processor is halted, it will be in the off state. This output is capable of driving one standard TTL load and 90 pF.

$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input is used to reset and start the MPU from a power down condition resulting from a power failure or initial start-up of the processor. This level sensitive input can also be used to reinitialize the machine at any time after start-up.

If a high level is detected in this input, this will signal the MPU to begin the reset sequence. During the reset sequence, the contents of the last two locations (FFFE, FFFF) in memory will be loaded into the Program Counter to point to the beginning of the reset routine. During the reset routine, the interrupt mask bit is set and must be cleared under program control before the MPU can be interrupted by $\overline{\text{IRQ}}$. While $\overline{\text{RESET}}$ is low (assuming a minimum of 8 clock cycles have occurred) the MPU output signals will be in the following states: VMA = low, BA = low, Data Bus = high impedance, R/$\overline{\text{W}}$ = high (read state), and the Address Bus will contain the reset address FFFE. Figure 8 illustrates a power up sequence using the $\overline{\text{RESET}}$ control line. After the power supply reaches 4.75 V, a minimum of eight clock cycles are required for the processor to stabilize in preparation for restarting. During these eight cycles, VMA will be in an indeterminate state so any devices that are enabled by VMA which could accept a false write during this time (such as battery-backed RAM) must be disabled until VMA is forced low after eight cycles. $\overline{\text{RESET}}$ can go high asynchronously with the system clock any time after the eighth cycle.

$\overline{\text{RESET}}$ timing is shown in Figure 8. The maximum rise and fall transition times are specified by t_{PCr} and t_{PCf}. If $\overline{\text{RESET}}$ is high at t_{PCS} (processor control setup time), as shown in Figure 8, in any given cycle then the restart sequence will begin on the next cycle as shown. The $\overline{\text{RESET}}$ control line may also be used to reinitialize the MPU system at any time during its operation. This is accomplished by pulsing RESET low for the duration of a minimum of three complete ϕ2 cycles. The $\overline{\text{RESET}}$ pulse can be completely asynchronous with the MPU system clock and will be recognized during ϕ2 if setup time t_{PCS} is met.

Interrupt Request ($\overline{\text{IRQ}}$) — This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being executed before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored away on the stack. Next, the MPU will respond to the interrupt request by setting the interrupt mask bit high so that no further interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations FFF8 and FFF9. An address loaded at these locations causes the MPU to branch to an interrupt routine in memory. Interrupt timing is shown in Figure 9.

FIGURE 8 — $\overline{\text{RESET}}$ TIMING

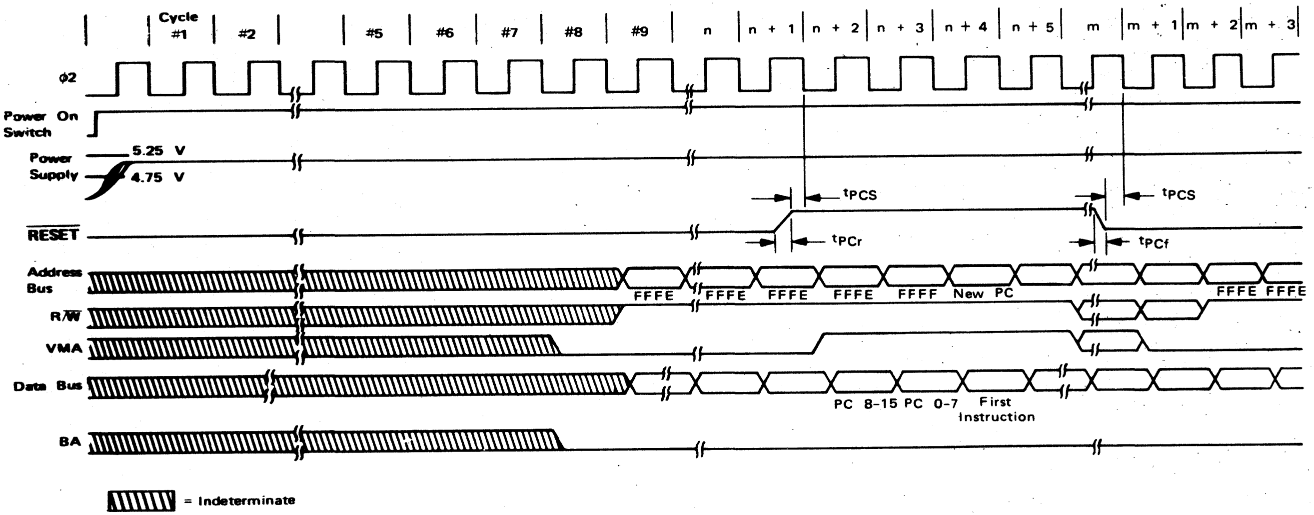
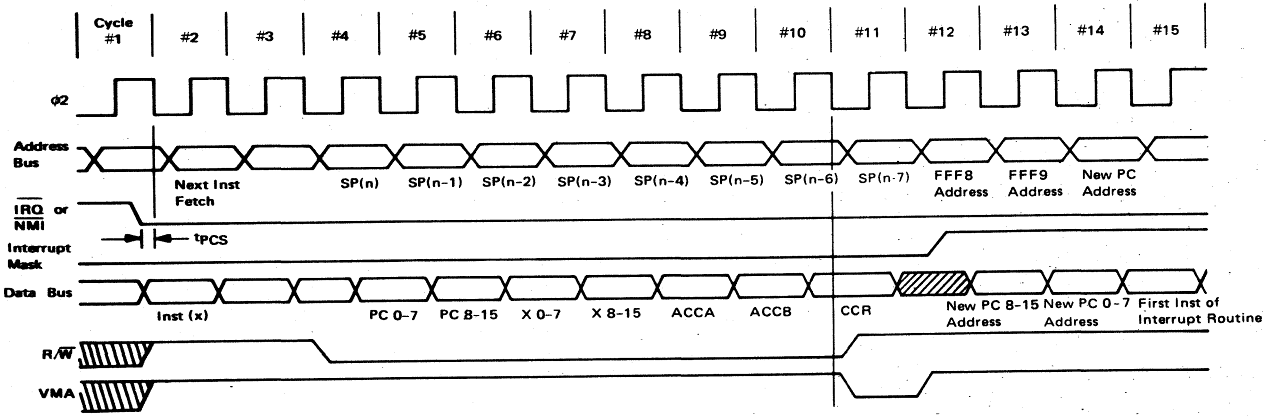


FIGURE 9 — INTERRUPT TIMING



The HALT line must be in the high state for interrupts to be serviced. Interrupts will be latched internally while HALT is low.

The IRQ has a high-impedance pullup device internal to the chip; however, a 3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

Non-Maskable Interrupt (NMI) and Wait for Interrupt (WAI) – The MC6800 is capable of handling two types of interrupts: maskable (IRQ) as described earlier, and non-maskable (NMI) which is an edge sensitive input. IRQ is maskable by the interrupt mask in the condition code register while NMI is not maskable. The handling of these interrupts by the MPU is the same except that each has its own vector address. The behavior of the MPU when interrupted is shown in Figure 9 which details the MPU response to an interrupt while the MPU is executing the control program. The interrupt shown could be either IRQ or NMI and can be asynchronous with respect to ϕ2. The interrupt is shown going low at time t_{PCS} in cycle #1 which precedes the first cycle of an instruction (OP code fetch). This instruction is not executed but instead the Program Counter (PC), Index Register (IX), Accumulators (ACCX), and the Condition Code Register (CCR) are pushed onto the stack.

The Interrupt Mask bit is set to prevent further interrupts. The address of the interrupt service routine is then fetched from FFFC, FFFD for an NMI interrupt and from FFF8, FFF9 for an IRQ interrupt. Upon completion of the interrupt service routine, the execution of RTI will pull the PC, IX, ACCX, and CCR off the stack; the Interrupt Mask bit is restored to its condition prior to Interrupts (see Figure 10).

Figure 11 is a similar interrupt sequence, except in this case, a WAIT instruction has been executed in preparation for the interrupt. This technique speeds up the MPU's response to the interrupt because the stacking of the PC, IX, ACCX, and the CCR is already done. While the MPU is waiting for the interrupt, Bus Available will go high indicating the following states of the control lines: VMA is low, and the Address Bus, R/W and Data Bus are all in the high impedance state. After the interrupt occurs, it is serviced as previously described.

A 3-10 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

MEMORY MAP FOR INTERRUPT VECTORS

Vector		Description
MS	LS	
FFFE	FFFF	Reset
FFFC	FFFD	Non-Maskable Interrupt
FFFA	FFFB	Software Interrupt
FFF8	FFF9	Interrupt Request

Refer to Figure 10 for program flow for Interrupts.

Three-State Control (TSC) – When the level sensitive Three-State Control (TSC) line is a logic "1", the Address Bus and the R/W line are placed in a high-impedance state. VMA and BA are forced low when TSC = "1" to prevent false reads or writes on any device enabled by VMA. It is necessary to delay program execution while TSC is held high. This is done by insuring that no transitions of ϕ1 (or ϕ2) occur during this period. (Logic levels of the clocks are irrelevant so long as they do not change). Since the MPU is a dynamic device, the ϕ1 clock can be stopped for a maximum time $PW_{\phi H}$ without destroying data within the MPU. TSC then can be used in a short Direct Memory Access (DMA) application.

Figure 12 shows the effect of TSC on the MPU. TSC must have its transitions at t_{TSE} (three-state enable) while holding ϕ1 high and ϕ2 low as shown. The Address Bus and R/W line will reach the high-impedance state at t_{TSD} (three-state delay), with VMA being forced low. In this example, the Data Bus is also in the high-impedance state while ϕ2 is being held low since DBE = ϕ2. At this point in time, a DMA transfer could occur on cycles #3 and #4. When TSC is returned low, the MPU Address and R/W lines return to the bus. Because it is too late in cycle #5 to access memory, this cycle is dead and used for synchronization. Program execution resumes in cycle #6.

Valid Memory Address (VMA) – This output indicates to peripheral devices that there is a valid address on the address bus. In normal operation, this signal should be utilized for enabling peripheral interfaces such as the PIA and ACIA. This signal is not three-state. One standard TTL load and 90 pF may be directly driven by this active high signal.

HALT – When this level sensitive input is in the low state, all activity in the machine will be halted. This input is level sensitive.

The HALT line provides an input to the MPU to allow control of program execution by an outside source. If HALT is high, the MPU will execute the instructions; if it is low, the MPU will go to a halted or idle mode. A response signal, Bus Available (BA) provides an indication of the current MPU status. When BA is low, the MPU is in the process of executing the control program; if BA is high, the MPU has halted and all internal activity has stopped.

When BA is high, the Address Bus, Data Bus, and R/W line will be in a high-impedance state, effectively removing the MPU from the system bus. VMA is forced low so that the floating system bus will not activate any device on the bus that is enabled by VMA.

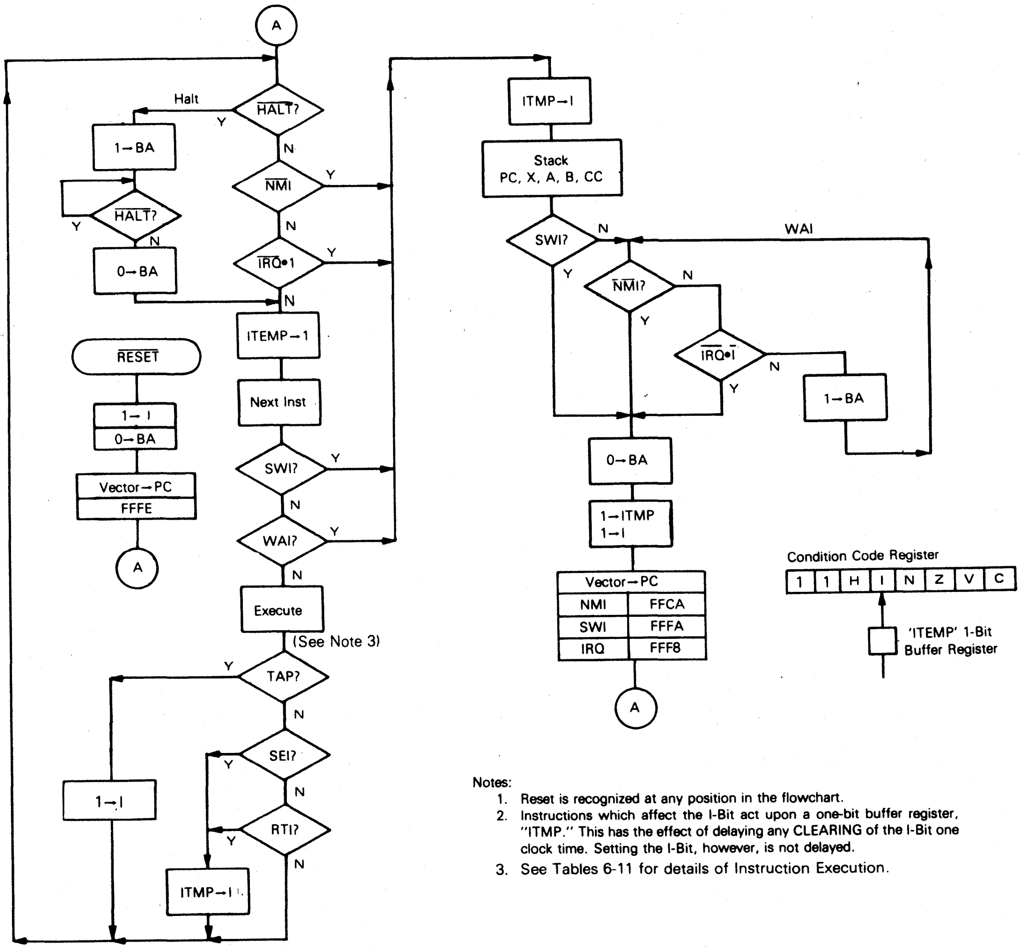
While the MPU is halted, all program activity is stopped, and if either an NMI or IRQ interrupt occurs, it will be latched into the MPU and acted on as soon as the MPU is taken out of the halted mode. If a RESET command occurs while the MPU is halted, the following states occur: VMA = low, BA = low, Data Bus = high impedance, R/W = high (read state), and the Address Bus will contain address FFFE as long as RESET is low. As soon as the RESET line goes high, the MPU will go to locations FFFE and FFFF for the address of the reset routine.

Figure 13 shows the timing relationships involved when halting the MPU. The instruction illustrated is a one byte, 2 cycle instruction such as CLRA. When HALT goes low, the MPU will halt after completing execution of the current instruction. The transition of HALT must occur t_{PCS} before the trailing edge of ϕ1 of the last cycle of an instruction (point A of Figure 13). HALT must not go low any time later than the minmum t_{PCS} specified.

The fetch of the OP code by the MPU is the first cycle of the instruction. If HALT had not been low at Point A but went low during ϕ2 of that cycle, the MPU would have halted after completion of the following instruction. BA will go high by time t_{BA} (bus available delay time) after the last instruction cycle. At this point in time, VMA is low and R/W, Address Bus, and the Data Bus are in the high-impedance state.

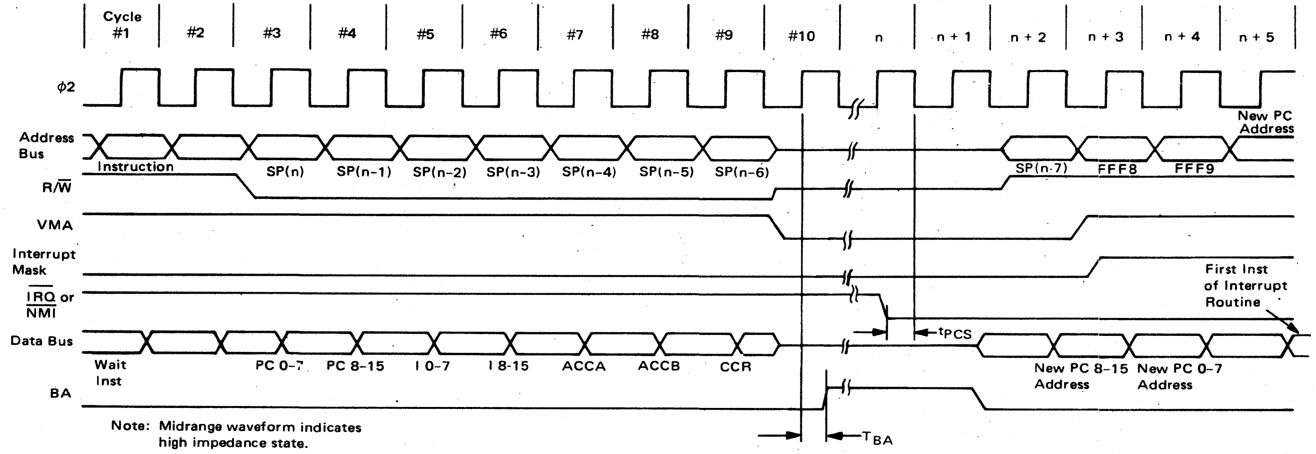
To debug programs it is advantageous to step through programs instruction by instruction. To do this, HALT must be brought high for one MPU cycle and then returned low as shown at point B of Figure 13. Again, the transitions of $\overline{\text{HALT}}$ must occur t_{PCS} before the trailing edge of $\phi 1$. BA will go low at t_{BA} after the leading edge of the next $\phi 1$, indicating that the Address Bus, Data Bus, VMA and R/$\overline{\text{W}}$ lines are back on the bus. A single byte, 2 cycle instruction such as LSR is used for this example also. During the first cycle, the instruction Y is fetched from address M+1. BA returns high at t_{BA} on the last cycle of the instruction indicating the MPU is off the bus. If instruction Y had been three cycles, the width of the BA low time would have been increased by one cycle.

FIGURE 10 — MPU FLOW CHART



Notes:
1. Reset is recognized at any position in the flowchart.
2. Instructions which affect the I-Bit act upon a one-bit buffer register, "ITMP." This has the effect of delaying any CLEARING of the I-Bit one clock time. Setting the I-Bit, however, is not delayed.
3. See Tables 6-11 for details of Instruction Execution.

FIGURE 11 — WAIT INSTRUCTION TIMING



Note: Midrange waveform indicates high impedance state.

FIGURE 12 — THREE-STATE CONTROL TIMING

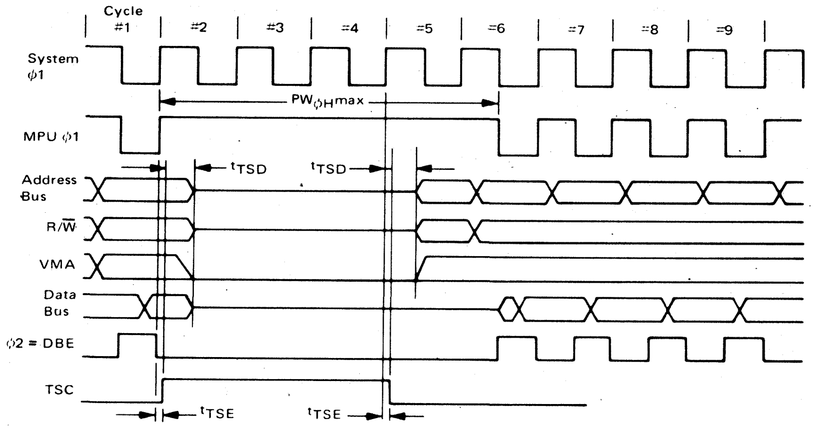
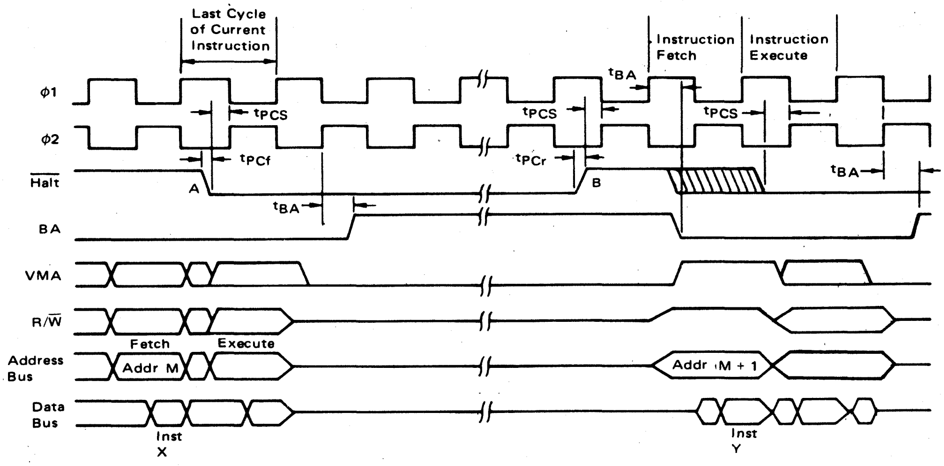


FIGURE 13 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



Note: Midrange waveform indicates high impedance state.

MPU REGISTERS

The MPU has three 16-bit registers and three 8-bit registers available for use by the programmer (Figure 14).

Program Counter — The program counter is a two byte (16 bits) register that points to the current program address.

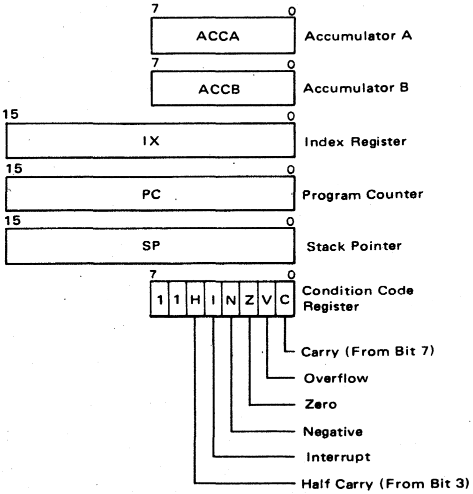
Stack Pointer — The stack pointer is a two byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access Read/Write memory that may have any location (address) that is convenient. In those applications that require storage of information in the stack when power is lost, the stack must be nonvolatile.

Index Register — The index register is a two byte register that is used to store data or a sixteen bit memory address for the Indexed mode of memory addressing.

Accumulators — The MPU contains two 8-bit accumulators that are used to hold operands and results from an arithmetic logic unit (ALU).

Condition Code Register — The condition code register indicates the results of an Arithmetic Logic Unit operation: Negative (N), Zero (Z), Overflow (V), Carry from bit 7 (C), and half carry from bit 3 (H). These bits of the Condition Code Register are used as testable conditions for the conditional branch instructions. Bit 4 is the interrupt mask bit (I). The unused bits of the Condition Code Register (b6 and b7) are ones.

FIGURE 14 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT



MPU INSTRUCTION SET

The MC6800 instructions are described in detail in the M6800 Programming Manual. This Section will provide a brief introduction and discuss their use in developing MC6800 control programs. The MC6800 has a set of 72 different executable source instructions. Included are binary and decimal arithmetic, logical, shift, rotate, load, store, conditional or unconditional branch, interrupt and stack manipulation instructions.

Each of the 72 executable instructions of the source language assembles into 1 to 3 bytes of machine code. The number of bytes depends on the particular instruction and on the addressing mode. (The addressing modes which are available for use with the various executive instructions are discussed later.)

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 72 instructions in all valid modes of addressing, are shown in Table 1. There are 197 valid machine codes, 59 of the 256 possible codes being unassigned.

When an instruction translates into two or three bytes of code, the second byte, or the second and third bytes contain(s) an operand, an address, or information from which an address is obtained during execution.

Microprocessor instructions are often divided into three general classifications: (1) memory reference, so called because they operate on specific memory locations; (2) operating instructions that function without needing a memory reference; (3) I/O instructions for transferring data between the microprocessor and peripheral devices.

In many instances, the MC6800 performs the same operation on both its internal accumulators and the external memory locations. In addition, the MC6800 interface adapters (PIA and ACIA) allow the MPU to treat peripheral devices exactly like other memory locations, hence, no I/O instructions as such are required. Because of these features, other classifications are more suitable for introducing the MC6800's instruction set: (1) Accumulator and memory operations; (2) Program control operations; (3) Condition Code Register operations.

TABLE 1 — HEXADECIMAL VALUES OF MACHINE CODES

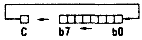
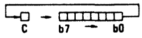
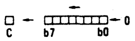
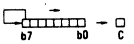
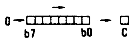
00	•			40	NEG	A		80	SUB	A	IMM	C0	SUB	B	IMM
01	NOP			41	•			81	CMP	A	IMM	C1	CMP	B	IMM
02	•			42	•			82	SBC	A	IMM	C2	SBC	B	IMM
03	•			43	COM	A		83	•			C3	•		
04	•			44	LSR	A		84	AND	A	IMM	C4	AND	B	IMM
05	•			45	•			85	BIT	A	IMM	C5	BIT	B	IMM
06	TAP			46	ROR	A		86	LDA	A	IMM	C6	LDA	B	IMM
07	TPA			47	ASR	A		87	•			C7	•		
08	INX			48	ASL	A		88	EOR	A	IMM	C8	EOR	B	IMM
09	DEX			49	ROL	A		89	ADC	A	IMM	C9	ADC	B	IMM
0A	CLV			4A	DEC	A		8A	ORA	A	IMM	CA	ORA	B	IMM
0B	SEV			4B	•			8B	ADD	A	IMM	CB	ADD	B	IMM
0C	CLC			4C	INC	A		8C	CPX	A	IMM	CC	•		
0D	SEC			4D	TST	A		8D	BSR		REL	CD	•		
0E	CLI			4E	•			8E	LDS		IMM	CE	LDX		IMM
0F	SEI			4F	CLR	A		8F	•			CF	•		
10	SBA			50	NEG	B		90	SUB	A	DIR	D0	SUB	B	DIR
11	CBA			51	•			91	CMP	A	DIR	D1	CMP	B	DIR
12	•			52	•			92	SBC	A	DIR	D2	SBC	B	DIR
13	•			53	COM	B		93	•			D3	•		
14	•			54	LSR	B		94	AND	A	DIR	D4	AND	B	DIR
15	•			55	•			95	BIT	A	DIR	D5	BIT	B	DIR
16	TAB			56	ROR	B		96	LDA	A	DIR	D6	LDA	B	DIR
17	TBA			57	ASR	B		97	STA	A	DIR	D7	STA	B	DIR
18	•			58	ASL	B		98	EOR	A	DIR	D8	EOR	B	DIR
19	DAA			59	ROL	B		99	ADC	A	DIR	D9	ADC	B	DIR
1A	•			5A	DEC	B		9A	ORA	A	DIR	DA	ORA	B	DIR
1B	ABA			5B	•			9B	ADD	A	DIR	DB	ADD	B	DIR
1C	•			5C	INC	B		9C	CPX		DIR	DC	•		
1D	•			5D	TST	B		9D	•			DD	•		
1E	•			5E	•			9E	LDS		DIR	DE	LDX		DIR
1F	•			5F	CLR	B		9F	STS		DIR	DF	STX		DIR
20	BRA		REL	60	NEG		IND	A0	SUB	A	IND	E0	SUB	B	IND
21	•			61	•			A1	CMP	A	IND	E1	CMP	B	IND
22	BHI		REL	62	•			A2	SBC	A	IND	E2	SBC	B	IND
23	BLS		REL	63	COM		IND	A3	•			E3	•		
24	BCC		REL	64	LSR		IND	A4	AND	A	IND	E4	AND	B	IND
25	BCS		REL	65	•			A5	BIT	A	IND	E5	BIT	B	IND
26	BNE		REL	66	ROR		IND	A6	LDA	A	IND	E6	LDA	B	IND
27	BEQ		REL	67	ASR		IND	A7	STA	A	IND	E7	STA	B	IND
28	BVC		REL	68	ASL		IND	A8	EOR	A	IND	E8	EOR	B	IND
29	BVS		REL	69	ROL		IND	A9	ADC	A	IND	E9	ADC	B	IND
2A	BPL		REL	6A	DEC		IND	AA	ORA	A	IND	EA	ORA	B	IND
2B	BMI		REL	6B	•			AB	ADD	A	IND	EB	ADD	B	IND
2C	BGE		REL	6C	INC		IND	AC	CPX		IND	EC	•		
2D	BLT		REL	6D	TST		IND	AD	JSR		IND	ED	•		
2E	BGT		REL	6E	JMP		IND	AE	LDS		IND	EE	LDX		IND
2F	BLE		REL	6F	CLR		IND	AF	STS		IND	EF	STX		IND
30	TSX			70	NEG		EXT	B0	SUB	A	EXT	F0	SUB	B	EXT
31	INS			71	•			B1	CMP	A	EXT	F1	CMP	B	EXT
32	PUL	A		72	•			B2	SBC	A	EXT	F2	SBC	B	EXT
33	PUL	B		73	COM		EXT	B3	•			F3	•		
34	DES			74	LSR		EXT	B4	AND	A	EXT	F4	AND	B	EXT
35	TXS			75	•			B5	BIT	A	EXT	F5	BIT	B	EXT
36	PSH	A		76	ROR		EXT	B6	LDA	A	EXT	F6	LDA	B	EXT
37	PSH	B		77	ASR		EXT	B7	STA	A	EXT	F7	STA	B	EXT
38	•			78	ASL		EXT	B8	EOR	A	EXT	F8	EOR	B	EXT
39	RTS			79	ROL		EXT	B9	ADC	A	EXT	F9	ADC	B	EXT
3A	•			7A	DEC		EXT	BA	ORA	A	EXT	FA	ORA	B	EXT
3B	RTI			7B	•			BB	ADD	A	EXT	FB	ADD	B	EXT
3C	•			7C	INC		EXT	BC	CPX		EXT	FC	•		
3D	•			7D	TST		EXT	BD	JSR		EXT	FD	•		
3E	WAI			7E	JMP		EXT	BE	LDS		EXT	FE	LDX		EXT
3F	SWI			7F	CLR		EXT	BF	STS		EXT	FF	STX		EXT

Notes: 1. Addressing Modes:

A = Accumulator A
B = Accumulator B
REL = Relative
IND = Indexed
IMM = Immediate
DIR = Direct

2. Unassigned code indicated by "*".

TABLE 2 — ACCUMULATOR AND MEMORY OPERATIONS

		ADDRESSING MODES															BOOLEAN/ARITHMETIC OPERATION	COND. CODE REG.					
		IMMED			DIRECT			INDEX			EXTND			IMPLIED			(All register labels refer to contents)	5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	●	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	●	●	↕	↕	R	●
Clear	CLR							6F	7	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	5	2	B1	4	3				A − M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	●	●	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	●	●	↕	↕	↕	↕
Complement, 1's	COM							63	7	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
Complement; 2's	NEG							60	7	2	70	6	3				00 − M → M	●	●	↕	↕	①	②
(Negate)	NEGA													40	2	1	00 − A → A	●	●	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	●	●	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	●	●	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	●	●	↕	↕	4	●
	DECA													4A	2	1	A − 1 → A	●	●	↕	↕	4	●
	DECB													5A	2	1	B − 1 → B	●	●	↕	↕	4	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	●	●	↕	↕	R	●
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	●	●	↕	↕	⑤	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	⑤	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	⑤	●
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	●	●	↕	↕	R	●
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	●	●	●	●	●	●
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	●	●	●	●	●	●
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	●	●	●	●	●	●
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	●	●	●	●	●	●
Rotate Left	ROL							69	7	2	79	6	3				M 	●	●	↕	↕	⑥	↕
	ROLA													49	2	1	A	●	●	↕	↕	⑥	↕
	ROLB													59	2	1	B	●	●	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M 	●	●	↕	↕	⑥	↕
	RORA													46	2	1	A	●	●	↕	↕	⑥	↕
	RORB													56	2	1	B	●	●	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M 	●	●	↕	↕	⑥	↕
	ASLA													48	2	1	A	●	●	↕	↕	⑥	↕
	ASLB													58	2	1	B	●	●	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M 	●	●	↕	↕	⑥	↕
	ASRA													47	2	1	A	●	●	↕	↕	⑥	↕
	ASRB													57	2	1	B	●	●	↕	↕	⑥	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M 	●	●	R	↕	⑥	↕
	LSRA													44	2	1	A	●	●	R	↕	⑥	↕
	LSRB													54	2	1	B	●	●	R	↕	⑥	↕
Store Acmltr.	STAA				97	4	2	A7	6	2	B7	5	3				A → M	●	●	↕	↕	R	●
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	●	●	↕	↕	↕	↕
Subtract Acmltrs.	SBA													10	2	1	A − B → A	●	●	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	●	●	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A − 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B − 00	●	●	↕	↕	R	R
																		H	I	N	Z	V	C

LEGEND:

OP	Operation Code (Hexadecimal);
~	Number of MPU Cycles;
#	Number of Program Bytes;
+	Arithmetic Plus;
−	Arithmetic Minus;
•	Boolean AND;
M_{SP}	Contents of memory location pointed to be Stack Pointer;
+	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 from bit 7
R	Reset Always
S	Set Always
↕	Test and set if true, cleared otherwise
●	Not Affected

CONDITION CODE REGISTER NOTES:

(Bit set if test is true and cleared otherwise)

1	(Bit V)	Test: Result = 10000000?
2	(Bit C)	Test: Result = 00000000?
3	(Bit C)	Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)
4	(Bit V)	Test: Operand = 10000000 prior to execution?
5	(Bit V)	Test: Operand = 01111111 prior to execution?
6	(Bit V)	Test: Set equal to result of N⊕C after shift has occurred.

Note – Accumulator addressing mode instructions are included in the column for IMPLIED addressing

PROGRAM CONTROL OPERATIONS

Program Control operation can be subdivided into two categories: (1) Index Register/Stack Pointer instructions; (2) Jump and Branch operations.

Index Register/Stack Pointer Operations

The instructions for direct operation on the MPU's Index Register and Stack Pointer are summarized in Table 3. Decrement (DEX, DES), increment (INX, INS), load (LDX, LDS), and store (STX, STS) instructions are provided for both. The Compare instruction, CPX, can be used to compare the Index Register to a 16-bit value and update the Condition Code Register accordingly.

The TSX instruction causes the Index Register to be loaded with the address of the last data byte put onto the "stack." The TXS instruction loads the Stack Pointer with a value equal to one less than the current contents of the Index Register. This causes the next byte to be pulled from the "stack" to come from the location indicated by the Index Register. The utility of these two instructions can be clarified by describing the "stack" concept relative to the M6800 system.

The "stack" can be thought of as a sequential list of data stored in the MPU's read/write memory. The Stack Pointer contains a 16-bit memory address that is used to access the list from one end on a last-in-first-out (LIFO) basis in contrast to the random access mode used by the MPU's other addressing modes.

The MC6800 instruction set and interrupt structure allow extensive use of the stack concept for efficient handling of data movement, subroutines and interrupts. The instructions can be used to establish one or more "stacks" anywhere in read/write memory. Stack length is limited only by the amount of memory that is made available.

Operation of the Stack Pointer with the Push and Pull instructions is illustrated in Figures 15 and 16. The Push instruction (PSHA) causes the contents of the indicated accumulator (A in this example) to be stored in memory at the location indicated by the Stack Pointer. The Stack Pointer is automatically decremented by one following the storage operation and is "pointing" to the next empty stack location. The Pull instruction (PULA or PULB) causes the last byte stacked to be loaded into the appropriate accumulator. The Stack Pointer is automatically incremented by one just prior to the data transfer so that it will point to the last byte stacked rather than the next empty location. Note that the PULL instruction does not "remove" the data from memory; in the example, 1A is still in location (m + 1) following execution of PULA. A subsequent PUSH instruction would overwrite that location with the new "pushed" data.

Execution of the Branch to Subroutine (BSR) and Jump to Subroutine (JSR) instructions cause a return address to be saved on the stack as shown in Figures 18 through 20. The stack is decremented after each byte of the return address is pushed onto the stack. For both of these instructions, the return address is the memory location following the bytes of code that correspond to the BSR and JSR instruction. The code required for JSR may be either two or three bytes, depending on whether the JSR is in the indexed (two bytes) or the extended (three bytes) addressing mode. Before it is stacked, the Program Counter is automatically incremented the correct number of times to be pointing at the location of the next instruction. The Return from Subroutine Instruction, RTS, causes the return address to be retrieved and loaded into the Program Counter as shown in Figure 21.

There are several operations that cause the status of the MPU to be saved on the stack. The Software Interrupt (SWI) and Wait for Interrupt (WAI) instructions as well as the maskable ($\overline{IRQ}$) and non-maskable ($\overline{NMI}$) hardware interrupts all cause the MPU's internal registers (except for the Stack Pointer itself) to be stacked as shown in Figure 23. MPU status is restored by the Return from Interrupt, RTI, as shown in Figure 22.

Jump and Branch Operation

The Jump and Branch instructions are summarized in Table 4. These instructions are used to control the transfer or operation from one point to another in the control program.

The No Operation instruction, NOP, while included here, is a jump operation in a very limited sense. Its only effect is to increment the Program Counter by one. It is useful during program development as a "stand-in" for some other instruction that is to be determined during debug. It is also used for equalizing the execution time through alternate paths in a control program.

TABLE 3 — INDEX REGISTER AND STACK POINTER INSTRUCTIONS

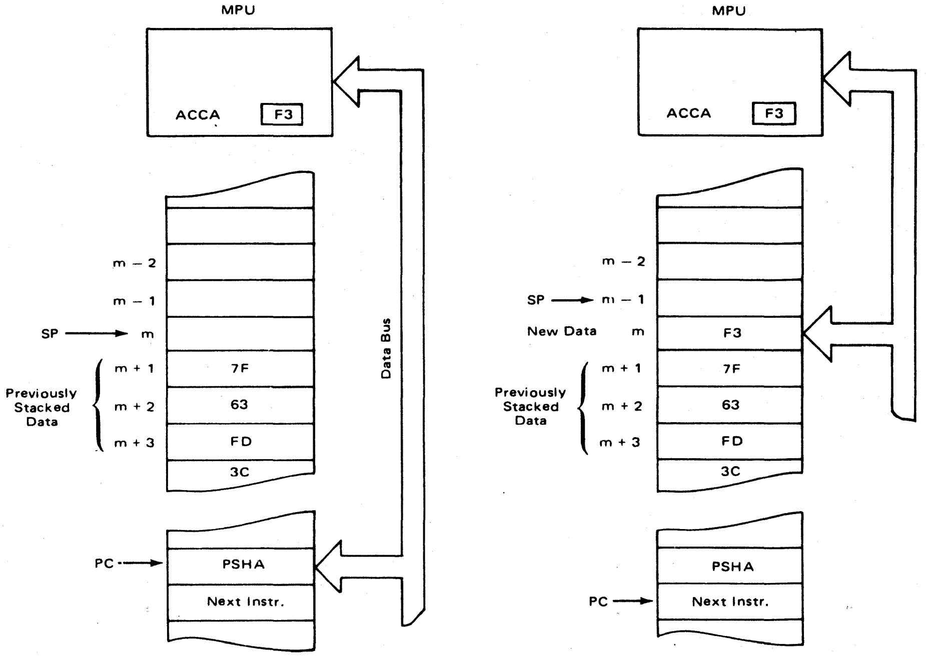
																			COND. CODE REG.					
		IMMED			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$X_H - M, X_L - (M + 1)$	●	●	①	↕	②	●
Decrement Index Reg	DEX													09	4	1	$X - 1 \rightarrow X$	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \rightarrow SP$	●	●	●	●	●	●
Increment Index Reg	INX													08	4	1	$X + 1 \rightarrow X$	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	4	1	$SP + 1 \rightarrow SP$	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \rightarrow X_H, (M + 1) \rightarrow X_L$	●	●	③	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H, (M + 1) \rightarrow SP_L$	●	●	③	↕	R	●
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow (M + 1)$	●	●	③	↕	R	●
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow (M + 1)$	●	●	③	↕	R	●
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	●	●	●	●	●	●
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	●	●	●	●	●	●

① (Bit N) Test: Sign bit of most significant (MS) byte of result = 1?

② (Bit V) Test: 2's complement overflow from subtraction of ms bytes?

③ (Bit N) Test: Result less than zero? (Bit 15 = 1)

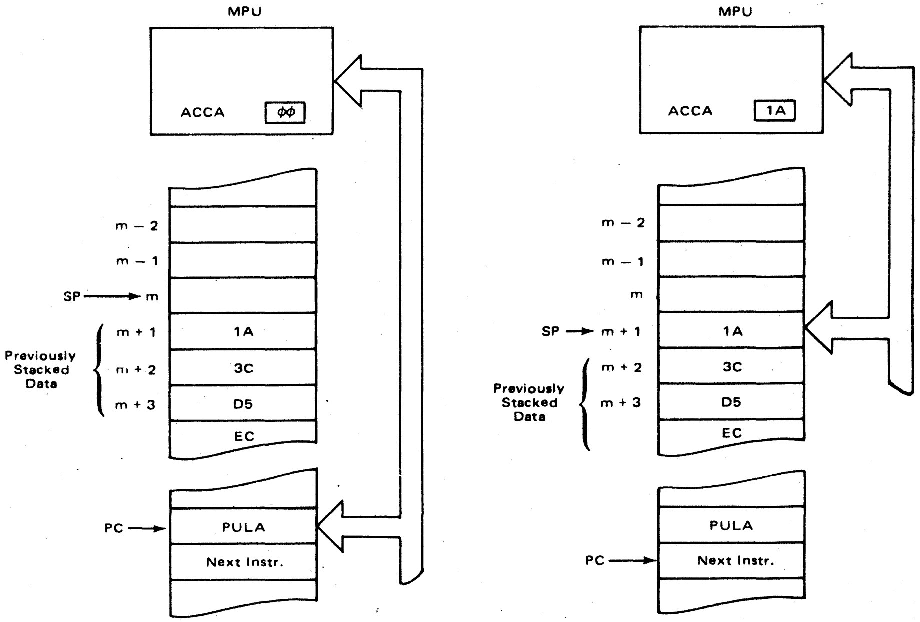
FIGURE 15 — STACK OPERATION, PUSH INSTRUCTION



(a) Before PSHA

(b) After PSHA

FIGURE 16 — STACK OPERATION, PULL INSTRUCTION



(a) Before PULA

(b) After PULA

TABLE 4 – JUMP AND BRANCH INSTRUCTIONS

		RELATIVE			INDEX			EXTND			IMPLIED				COND. CODE REG.					
															5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	BRANCH TEST	H	I	N	Z	V	C
Branch Always	BRA	20	4	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC	24	4	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS	25	4	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ	27	4	2										Z = 1	●	●	●	●	●	●
Branch If ⩾ Zero	BGE	2C	4	2										N ⊕ V = 0	●	●	●	●	●	●
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	●	●	●	●	●	●
Branch If Higher	BHI	22	4	2										C + Z = 0	●	●	●	●	●	●
Branch If ⩽ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	●	●	●	●	●	●
Branch If Minus	BMI	2B	4	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC	28	4	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS	29	4	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL	2A	4	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR	8D	8	2											●	●	●	●	●	●
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	●	●	●	●	●	●
Jump To Subroutine	JSR				AD	8	2	BD	9	3					●	●	●	●	●	●
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	●	●	●	●	●	●
Return From Interrupt	RTI					J					3B	10	1		①					
Return From Subroutine	RTS										39	5	1		●	●	●	●	●	●
Software Interrupt	SWI										3F	12	1	See Special Operations	●	●	●	●	●	●
Wait for Interrupt*	WAI										3E	9	1		●	②	●	●	●	●

*WAI puts Address Bus, R/W, and Data Bus in the three-state mode while VMA is held low.

① (All) Load Condition Code Register from Stack. (See Special Operations)

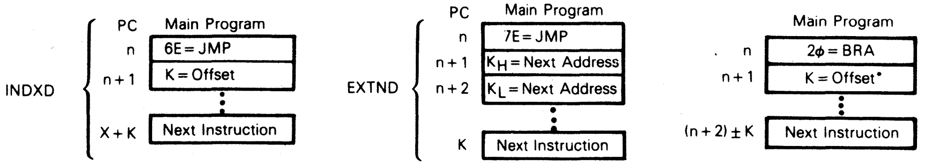
② (Bit 1) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.

Execution of the Jump Instruction, JMP, and Branch Always, BRA, affects program flow as shown in Figure 17. When the MPU encounters the Jump (Indexed) instruction, it adds the offset to the value in the Index Register and uses the result as the address of the next instruction to be executed. In the extended addressing mode, the address of the next instruction to be executed is fetched from the two locations immediately following the JMP instruction. The Branch Always (BRA) instruction is similar to the JMP (extended) instruction except that the relative addressing mode applies and the branch is limited to the range within – 125 or + 127+129 bytes of the branch instruction itself. The opcode for the BRA instruction requires one less byte than JMP (extended) but takes one more cycle to execute.

The effect on program flow for the Jump to Subroutine (JSR) and Branch to Subroutine (BSR) is shown in Figures 18 through 20. Note that the Program Counter is properly incremented to be pointing at the correct return address before it is stacked. Operation of the Branch to Subroutine and Jump to Subroutine (extended) instruction is similar except for the range. The BSR instruction requires less opcode than JSR (2 bytes versus 3 bytes) and also executes one cycle faster than JSR. The Return from Subroutine, RTS, is used as the end of a subroutine to return to the main program as indicated in Figure 21.

The effect of executing the Software Interrupt, SWI, and the Wait for Interrupt, WAI, and their relationship to the hardware interrupts is shown in Figure 22. SWI causes the MPU contents to be stacked and then fetches the starting address of the interrupt routine from the memory locations that respond to the addresses FFFA and FFFB. Note that as in the case of the subroutine instructions, the Program Counter is incremented to point at the correct return address before being stacked. The Return from Interrupt instruction, RTI, (Figure 22) is used at the end of an interrupt routine to restore control to the main program. The SWI instruction is useful for inserting break points in the control program, that is, it can be used to stop operation and put the MPU registers in memory where they can be examined. The WAI instruction is used to decrease the time required to service a hardware interrupt; it stacks the MPU contents and then waits for the interrupt to occur, effectively removing the stacking time from a hardware interrupt sequence.

FIGURE 17 – PROGRAM FLOW FOR JUMP AND BRANCH INSTRUCTIONS

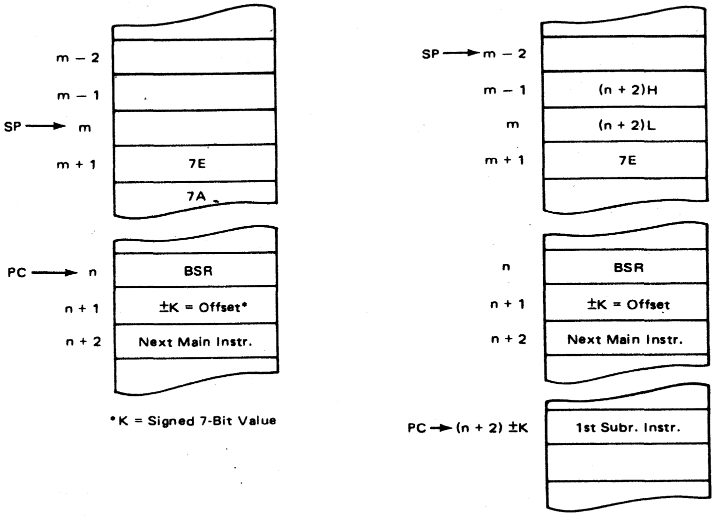


*K = Signed 7-bit value

(a) Jump

(b) Branch

FIGURE 18 — PROGRAM FLOW FOR BSR

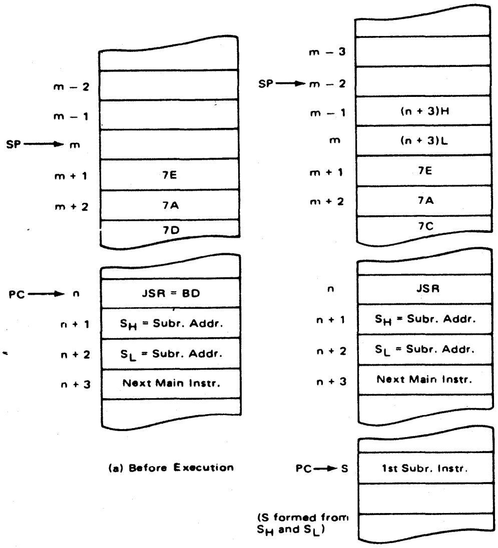


*K = Signed 7-Bit Value

(a) Before Execution

(b) After Execution

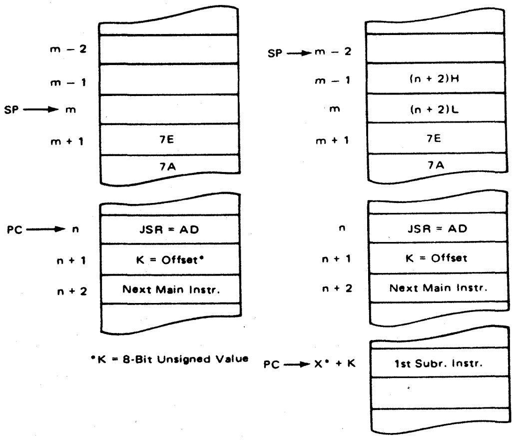
FIGURE 19 — PROGRAM FLOW FOR JSR (EXTENDED)



(a) Before Execution

(b) After Execution

FIGURE 20 — PROGRAM FLOW FOR JSR (INDEXED)



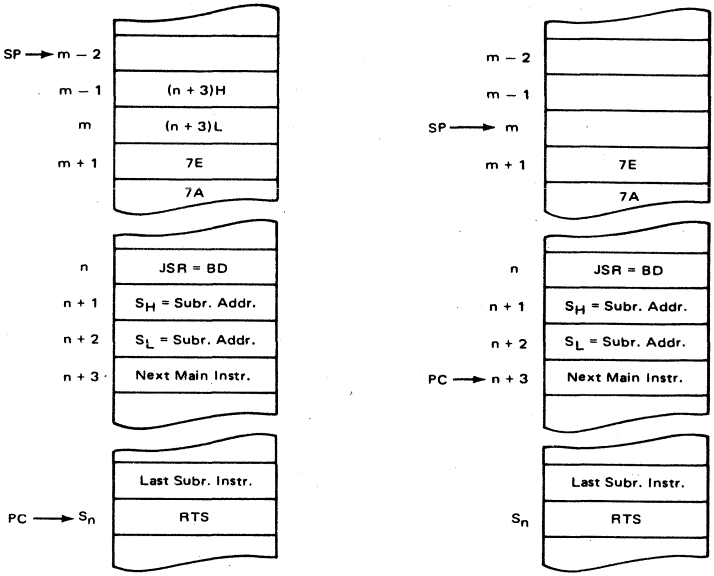
*K = 8-Bit Unsigned Value

*Contents of Index Register

(a) Before Execution

(b) After Execution

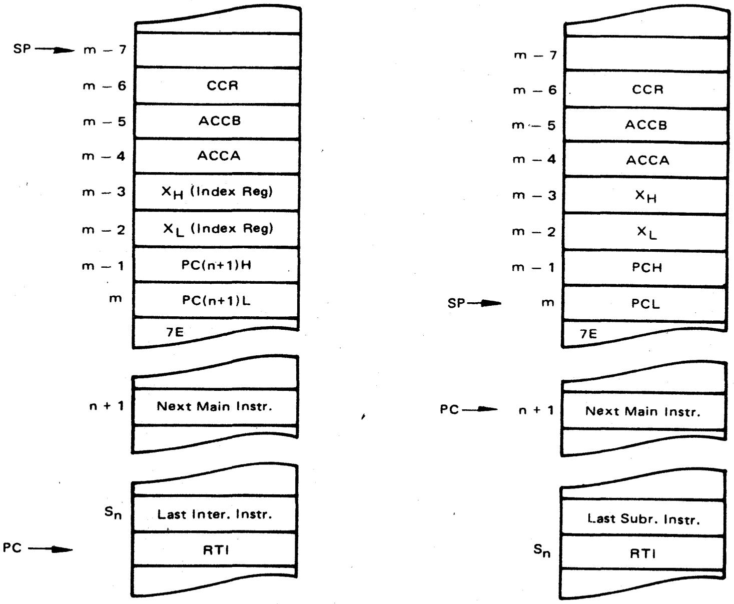
FIGURE 21 — PROGRAM FLOW FOR RTS



(a) Before Execution

(b) After Execution

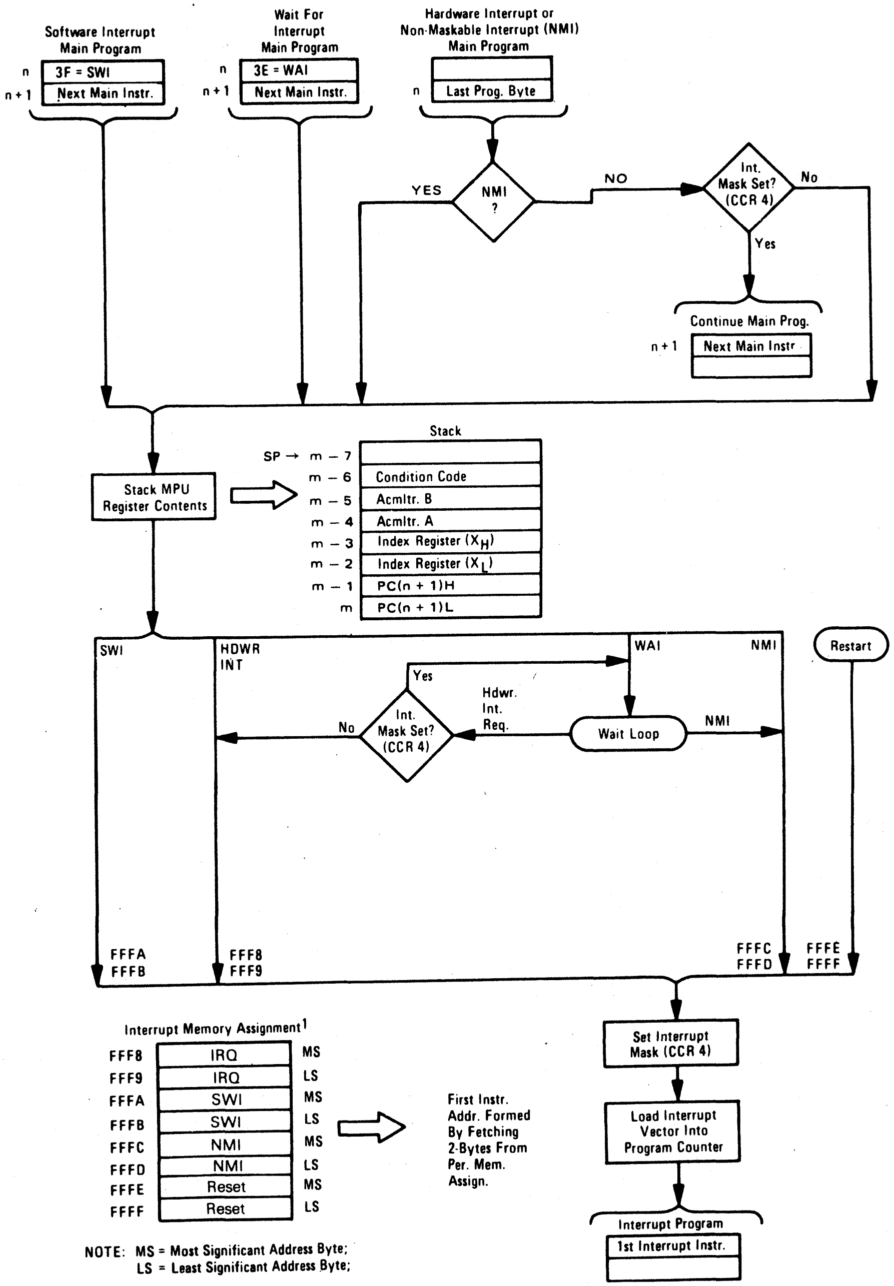
FIGURE 22 — PROGRAM FLOW FOR RTI



(a) Before Execution

(b) After Execution

FIGURE 23 — PROGRAM FLOW FOR INTERRUPTS



NOTE: MS = Most Significant Address Byte;
LS = Least Significant Address Byte;

FIGURE 24 — CONDITIONAL BRANCH INSTRUCTIONS

BMI : $N = 1$;
BPL : $N = \phi$;

BVC : $V = \phi$;
BVS : $V = 1$;

BHI : $C + Z = \phi$;
BLS : $C + Z = 1$;

BEQ : $Z = 1$;
BNE : $Z = \phi$;

BCC : $C = \phi$;
BCS : $C = 1$;

BLT : $N \oplus V = 1$;
BGE : $N \oplus V = \phi$;

BLE : $Z + (N \oplus V) = 1$;
BGT : $Z + (N \oplus V) = \phi$;

The conditional branch instructions, Figure 24, consists of seven pairs of complementary instructions. They are used to test the results of the preceding operation and either continue with the next instruction in sequence (test fails) or cause a branch to another point in the program (test succeeds).

Four of the pairs are used for simple tests of status bits N, Z, V, and C:

1. Branch on Minus (BMI) and Branch On Plus (BPL) tests the sign bit, N, to determine if the previous result was negative or positive, respectively.
2. Branch On Equal (BEQ) and Branch On Not Equal (BNE) are used to test the zero status bit, Z, to determine whether or not the result of the previous operation was equal to zero. These two instructions are useful following a Compare (CMP) instruction to test for equality between an accumulator and the operand. They are also used following the Bit Test (BIT) to determine whether or not the same bit positions are set in an accumulator and the operand.
3. Branch On Overflow Clear (BVC) and Branch On Overflow Set (BVS) tests the state of the V bit to determine if the previous operation caused an arithmetic overflow.
4. Branch On Carry Clear (BCC) and Branch On Carry Set (BCS) tests the state of the C bit to determine if the previous operation caused a carry to occur. BCC and BCS are useful for testing relative magnitude when the values being tested are regarded as unsigned binary numbers, that is, the values are in the range 00 (lowest) to FF (highest). BCC following a comparison (CMP) will cause a branch if the (unsigned) value in the accumulator is higher than or the same as the value of the operand. Conversely, BCS will cause a branch if the accumulator value is lower than the operand.

The fifth complementary pair, Branch On Higher (BHI) and Branch On Lower or Same (BLS) are, in a sense, complements to BCC and BCS. BHI tests for both C and $Z = 0$; if used following a CMP, it will cause a branch if the value in the accumulator is higher than the operand. Conversely, BLS will cause a branch if the unsigned binary value in the accumulator is lower than or the same as the operand.

The remaining two pairs are useful in testing results of operations in which the values are regarded as signed two's complement numbers. This differs from the unsigned binary case in the following sense: in unsigned, the orientation is higher or lower; in signed two's complement, the comparison is between larger or smaller where the range of values is between -128 and $+127$.

Branch On Less Than Zero (BLT) and Branch On Greater Than Or Equal Zero (BGE) test the status bits for $N \oplus V = 1$ and $N \oplus V = 0$, respectively. BLT will always cause a branch following an operation in which two negative numbers were added. In addition, it will cause a branch following a CMP in which the value in the accumulator was negative and the operand was positive. BLT will never cause a branch following a CMP in which the accumulator value was positive and the operand negative. BGE, the complement to BLT, will cause a branch following operations in which two positive values were added or in which the result was zero.

The last pair, Branch On Less Than Or Equal Zero (BLE) and Branch On Greater Than Zero (BGT) test the status bits for $Z \oplus (N + V) = 1$ and $Z \oplus (N + V) = 0$, respectively. The action of BLE is identical to that for BLT except that a branch will also occur if the result of the previous result was zero. Conversely, BGT is similar to BGE except that no branch will occur following a zero result.

CONDITION CODE REGISTER OPERATIONS

The Condition Code Register (CCR) is a 6-bit register within the MPU that is useful in controlling program flow during system operation. The bits are defined in Figure 25.

The instructions shown in Table 5 are available to the user for direct manipulation of the CCR.

A CLI-WAI instruction sequence operated properly, with early MC6800 processors, only if the preceding instruction was odd (Least Significant Bit = 1). Similarly it was advisable to precede any SEI instruction with an odd opcode — such as NOP. These precautions are not necessary for MC6800 processors indicating manufacture in November 1977 or later.

Svstems which require an interrupt window to be opened under program control should use a CLI-NOP-SEI sequence rather than CLI-SEI.

FIGURE 25 — CONDITION CODE REGISTER BIT DEFINITION

b_5	b_4	b_3	b_2	b_1	b_0
H	I	N	Z	V	C

H = Half-carry; set whenever a carry from b_3 to b_4 of the result is generated by ADD, ABA, ADC; cleared if no b_3 to b_4 carry; not affected by other instructions.

I = Interrupt Mask; set by hardware or software interrupt or SEI instruction; cleared by CLI instruction. (Normally not used in arithmetic operations.) Restored to a zero as a result of an RT1 instruction if I_m stored on the stacked is low.

N = Negative; set if high order bit (b_7) of result is set; cleared otherwise.

Z = Zero; set if result = 0; cleared otherwise.

V = Overlow; set if there was arithmetic overflow as a result of the operation; cleared otherwise.

C = Carry; set if there was a carry from the most significant bit (b_7) of the result; cleared otherwise.

TABLE 5 — CONDITION CODE REGISTER INSTRUCTIONS

						COND. CODE REG.					
		IMPLIED				5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	=	BOOLEAN OPERATION	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	●
Acmltr A → CCR	TAP	06	2	1	A → CCR	①					
CCR → Acmltr A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

R = Reset
S = Set
● = Not affected

① (ALL) Set according to the contents of Accumulator A.

ADDRESSING MODES

The MPU operates on 8-bit binary numbers presented to it via the Data Bus. A given number (byte) may represent either data or an instruction to be executed, depending on where it is encountered in the control program. The M6800 has 72 unique instructions, however, it recognizes and takes action on 197 of the 256 possibilitis that can occur using an 8-bit word length. This larger number of instructions results from the fact that many of the executive instructions have more than one addressing mode.

These addressing modes refer to the manner in which the program causes the MPU to obtain its instructions and data. The programmer must have a method for addressing the MPU's internal registers and all of the external memory locations.

Selection of the desired addressing mode is made by the user as the source statements are written. Translation into appropriate opcode then depends on the method used. If manual translation is used, the addressing mode is inherent in the opcode. For example, the Immediate, Direct, Indexed, and Extended modes may all be used with the ADD instruction. The proper mode is determined by selecting (hexadecimal notation) 8B, 9B, AB, or BB, respectively.

The source statement format includes adequate information for the selection if an assembler program is used to generate the opcode. For instance, the Immediate mode is selected by the Assembler whenever it encounters the "#" symbol in the operand field. Similarly, an "X" in the operand field causes the Indexed mode to be selected. Only the Relative mode applies to the branch instructions, therefore, the mnemonic instruction itself is enough for the Assembler to determine addressing mode.

For the instructions that use both Direct and Extended modes, the Assembler selects the Direct mode if the operand value is in the range 0-255 and Extended otherwise. There are a number of instructions for which the Extended mode is valid but the Direct is not. For these instructions, the Assembler automatically selects the Extended mode even if the operand is in the 0-255 range. The addressing modes are summarized in Figure 26.

Inherent (Includes "Accumulator Addressing" Mode)

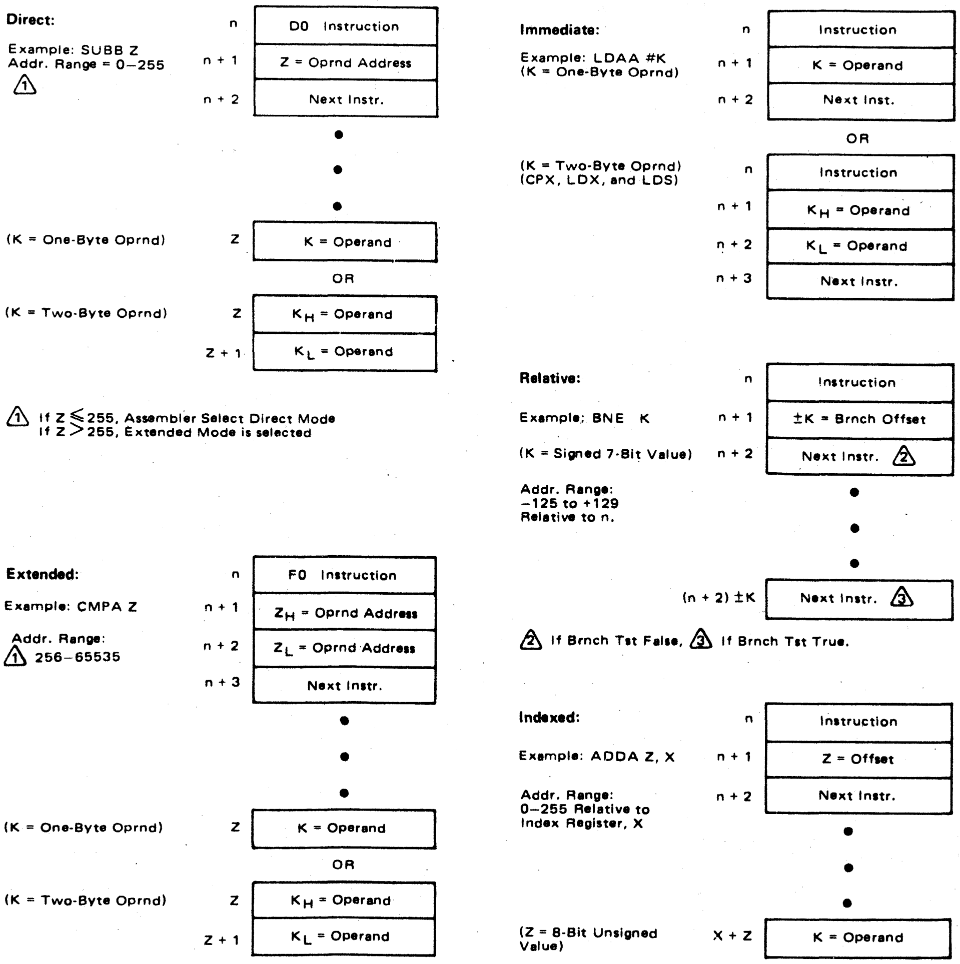
The successive fields in a statement are normally separated by one or more spaces. An exception to this rule occurs for instructions that use dual addressing in the operand field and for instructions that must distinguish between the two accumulators. In these cases, A and B are "operands" but the space between them and the operator may be omitted. This is commonly done, resulting in apparent four character mnemonics for those instructions.

The addition instruction, ADD, provides an example of dual addressing in the operand field:

	Operator	Operand	Comment
	ADDA	MEM12	ADD CONTENTS OF MEM12 TO ACCA
or			
	ADDB	MEM12	ADD CONTENTS OF MEM12 TO ACCB

The example used earlier for the test instruction, TST, also applies to the accumulators and uses the "accumulator addressing mode" to designate which of the two accumulators is being tested:

FIGURE 26 — ADDRESSING MODE SUMMARY



	Operator	**Comment**
	TSTB	TEST CONTENTS OF ACCB
or		
	TSTA	TEST CONTENTS OF ACCA

A number of the instructions either alone or together with an accumulator operand contain all of the address information that is required, that is, "inherent" in the instruction itself. For instance, the instruction ABA causes the MPU to add the contents of accmulators A and B together and place the result in accumulator A. The instruction INCB, another example of "accumulator addressing," causes the contents of accumulator B to be increased by one. Similarly, INX, increment the Index Register, causes the contents of the Index Register to be increased by one.

Program flow for instructions of this type is illustrated in Figures 27 and 28. In these figures, the general case is shown on the left and a specific example is shown on the right. Numerical examples are in decimal notation. Instructions of this type require only one byte of opcode. Cycle-by-cycle operation of the inherent mode is shown in Table 6.

Immediate Addressing Mode — In the Immediate addressing mode, the operand is the value that is to be operated on. For instance, the instruction

Operator	**Operand**	**Comment**
LDAA	#25	LOAD 25 INTO ACCA

causes the MPU to "immediately load accumulator A with the value 25"; no further address reference is required. The Immediate mode is selected by preceding the operand value with the "#" symbol. Program flow for this addressing mode is illustrated in Figure 29.

The operand format allows either properly defined symbols or numerical values. Except for the instructions CPX, LDX, and LDS, the operand may be any value in the range 0 to 255. Since Compare Index Register (CPX), Load Index Register (LDX), and Load Stack Pointer (LDS), require 16-bit values, the immediate mode for these three instructions require two-byte operands. In the Immediate addressing mode, the "address" of the operand is effectively the memory location immediately following the instruction itself. Table 7 shows the cycle-by-cycle operation for the immediate addressing mode.

Direct and Extended Addressing Modes — In the Direct and Extended modes of addressing, the operand field of the source statement is the *address* of the value that is to be operated on. The Direct and Extended modes differ only in the range of memory locations to which they can direct the MPU. Direct addressing generates a single 8-bit operand and, hence, can address only memory locations 0 through 255; a two byte operand is generated for Extended addressing, enabling the MPU to reach the remaining memory locations, 256 through 65535. An example of Direct addressing and its effect on program flow is illustrated in Figure 30.

The MPU, after encountering the opcode for the instruction LDAA (Direct) at memory location 5004 (Program Counter = 5004), looks in the next location, 5005, for the address of the operand. It then sets the program counter equal to the value found there (100 in the example) and fetches the operand, in this case a value to be loaded into accumulator A, from that location. For instructions requiring a two-byte operand such as LDX (Load the Index Register), the operand bytes would be retrieved from locations 100 and 101. Table 8 shows the cycle-by-cycle operation for the direct mode of addressing.

Extended addressing, Figure 31, is similar except that a two-byte address is obtained from locations 5007 and 5008 after the LDAB (Extended) opcode shows up in location 5006. Extended addressing can be thought of as the "standard" addressing mode, that is, it is a method of reaching any place in memory. Direct addressing, since only one address byte is required, provides a faster method of processing data and generates fewer bytes of control code. In most applications, the direct addressing range, memory locations 0-255, are reserved for RAM. They are used for data buffering and temporary storage of system variables, the area in which faster addressing is of most value. Cycle-by-cycle operation is shown in Table 9 for Extended Addressing.

FIGURE 27 — INHERENT ADDRESSING

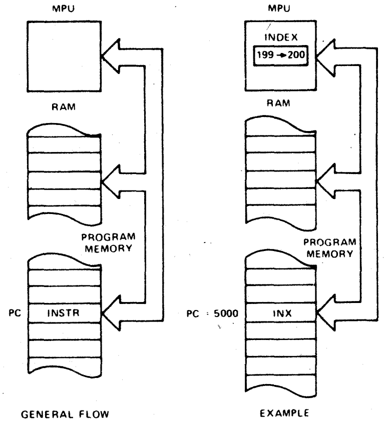
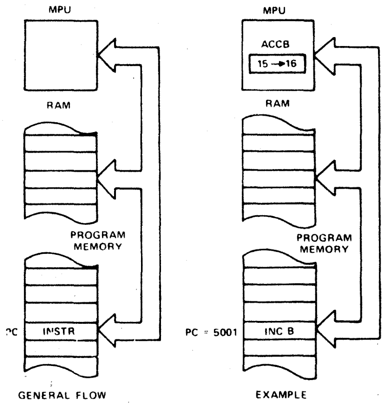


FIGURE 28 — ACCUMULATOR ADDRESSING



Relative Address Mode – In both the Direct and Extended modes, the address obtained by the MPU is an absolute numerical address. The Relative addressing mode, implemented for the MPU's branch instructions, specifies a memory location relative to the Program Counter's current location. Branch instructions generate two bytes of machine code, one for the instruction opcode and one for the "relative" address (see Figure 32). Since it is desirable to be able to branch in either direction, the 8-bit address byte is interpreted as a signed 7-bit value; the 8th bit of the operand is treated as a sign bit, "0" = plus and "1" = minus. The remaining seven bits represent the numerical value. This results in a relative addressing range of ± 127 with respect to the location of the branch instruction itself. However, the branch range is computed with respect to the next instruction that would be executed if the branch conditions are not satisfied. Since two bytes are generated, the next instruction is located at PC + 2. If D is defined as the address of the branch destination, the range is then:

$$(PC+2)-127 \leq D \leq (PC+2)+127$$

or

$$PC-125 \leq D \leq PC+129$$

that is, the destination of the branch instruction must be within −125 to +129 memory locations of the branch instruction itself. For transferring control beyond this range, the unconditional jump (JMP), jump to subroutine (JSR), and return from subroutine (RTS) are used.

In Figure 32, when the MPU encounters the opcode for BEQ (Branch if result of last instruction was zero), it tests the Zero bit in the Condition Code Register. If that bit is "0," indicating a non-zero result, the MPU continues execution with the next instruction (in location 5010 in Figure 32). If the previous result was zero, the branch condition is satisfied and the MPU adds the offset, 15 in this case, to PC + 2 and branches to location 5025 for the next instruction.

The branch instructions allow the programmer to efficiently direct the MPU to one point or another in the control program depending on the outcome of test results. Since the control program is normally in read-only memory and cannot be changed, the relative address used in execution of branch instructions is a constant numerical value. Cycle-by-cycle operation is shown in Table 10 for relative addressing.

Indexed Addressing Mode – With Indexed addressing, the numerical address is variable and depends on the current contents of the Index Register. A source statement such as

Operator	Operand	Comment
STAA	X	PUT A IN INDEXED LOCATION

causes the MPU to store the contents of accumulator A in

TABLE 6 – INHERENT MODE CYCLE-BY-CYCLE OPERATION

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
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	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Previous Register Contents	1	Irrelevant Data (Note 1)
		4	0	New Register Contents	1	Irrelevant Data (Note 1)
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Accumulator Data
		4	0	Stack Pointer – 1	1	Accumulator Data
PUL	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Operand Data from Stack
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	0	New Index Register	1	Irrelevant Data (Note 1)
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Index Register	1	Irrelevant Data
		4	0	New Stack Pointer	1	Irrelevant Data
RTS	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)

TABLE 6 — INHERENT MODE CYCLE-BY-CYCLE OPERATION (CONTINUED)

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

Note 1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.

Note 2. Data is ignored by the MPU.

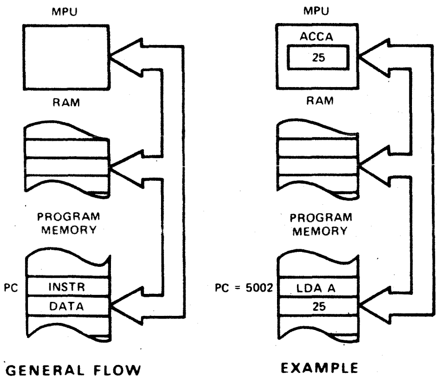
Note 3. While the MPU is waiting for the interrupt, Bus Available will go high indicating the following states of the control lines: VMA is low; Address Bus, R/$\overline{W}$, and Data Bus are all in the high impedance state.

the memory location specified by the contents of the Index Register (recall that the label "X" is reserved to designate the Index Register). Since there are instructions for manipulating X during program execution (LDX, INX, DEC, etc.), the Indexed addressing mode provides a dynamic "on the fly" way to modify program activity.

The operand field can also contain a numerical value that will be automatically added to X during execution. This format is illustrated in Figure 33.

When the MPU encounters the LDAB (Indexed) opcode in location 5006, it looks in the next memory location for the value to be added to X (5 in the example) and calculates the required address by adding 5 to the present Index Register value of 400. In the operand format, the offset may be represented by a label or a numerical value in the range 0-255 as in the example. In the earlier example, STAA X, the operand is equivalent to 0, X, that is, the 0 may be omitted when the desired address is equal to X. Table 11 shows the cycle-by-cycle operation for the Indexed Mode of Addressing.

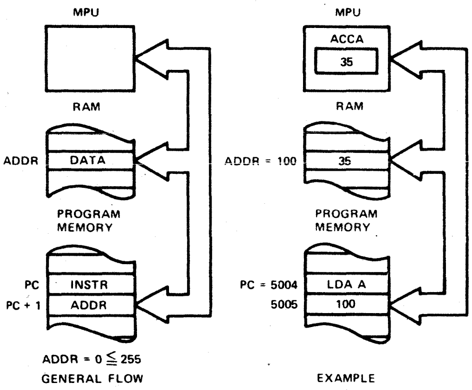
FIGURE 29 — IMMEDIATE ADDRESSING MODE



GENERAL FLOW

EXAMPLE

FIGURE 30 — DIRECT ADDRESSING MODE



GENERAL FLOW

EXAMPLE

TABLE 7 — IMMEDIATE MODE CYCLE-BY-CYCLE OPERATION

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data
CPX LDS LDX	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data (High Order Byte)
		3	1	Op Code Address + 2	1	Operand Data (Low Order Byte)

TABLE 8 — DIRECT MODE CYCLE-BY-CYCLE OPERATION

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data
CPX LDS LDX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data (High Order Byte)
		4	1	Operand Address + 1	1	Operand Data (Low Order Byte)
STA	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address
		3	0	Destination Address	1	Irrelevant Data (Note 1)
		4	1	Destination Address	0	Data from Accumulator
STS STX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	0	Address of Operand	1	Irrelevant Data (Note 1)
		4	1	Address of Operand	0	Register Data (High Order Byte)
		5	1	Address of Operand + 1	0	Register Data (Low Order Byte)

Note 1. If device which is address during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.

FIGURE 31 — EXTENDED ADDRESSING MODE

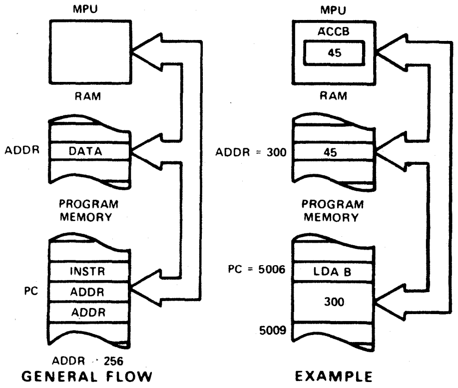


TABLE 9 — EXTENDED MODE CYCLE-BY-CYCLE

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	1	Irrelevant Data (Note 1)
		5	1	Address of Operand	0	Operand Data (High Order Byte)
		6	1	Address of Operand + 1	0	Operand Data (Low Order Byte)
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	1	Subroutine Starting Address	1	Op Code of Next Instruction
		5	1	Stack Pointer	0	Return Address (Low Order Byte)
		6	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		7	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		8	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		9	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	1	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data (High Order Byte)
		5	1	Address of Operand + 1	1	Operand Data (Low Order Byte)
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	1	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	0	Operand Destination Address	1	Irrelevant Data (Note 1)
		5	1	Operand Destination Address	0	Data from Accumulator
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Current Operand Data
		5	0	Address of Operand	1	Irrelevant Data (Note 1)
		6	1/0 (Note 2)	Address of Operand	0	New Operand Data (Note 2)

Note 1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.

Note 2. For TST, VMA = 0 and Operand data does not change.

FIGURE 32 — RELATIVE ADDRESSING MODE

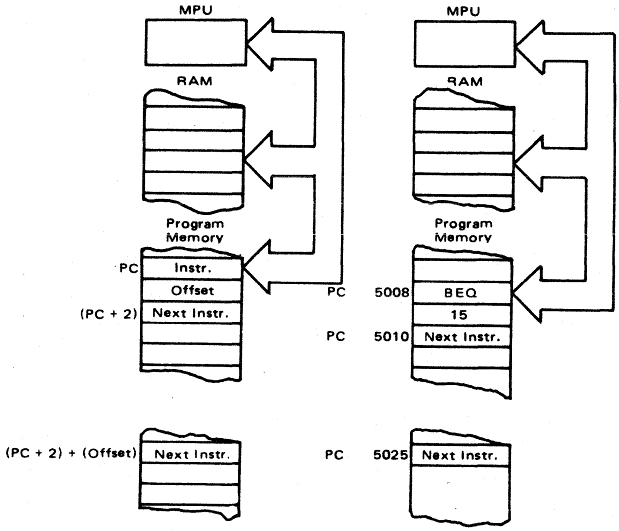


FIGURE 33 — INDEXED ADDRESSING MODE

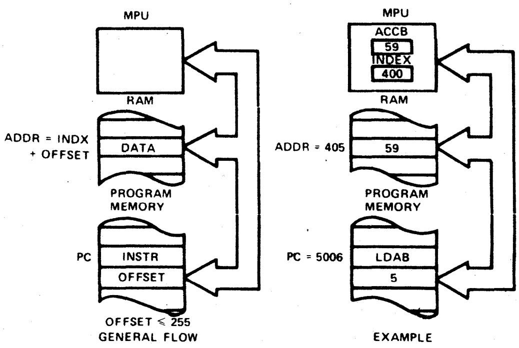


TABLE 10 — RELATIVE MODE CYCLE-BY-CYCLE OPERATION

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
BCC BHI BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMI BVS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		4	0	Branch Address	1	Irrelevant Data (Note 1)
BSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		8	0	Subroutine Address	1	Irrelevant Data (Note 1)

Note 1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.

TABLE 11 — INDEXED MODE CYCLE-BY-CYCLE

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED						
JMP	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data
CPX LDS LDX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data (High Order Byte)
		6	1	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STA	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Current Operand Data
		6	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		7	1/0 (Note 2)	Index Register Plus Offset	0	New Operand Data (Note 2)
STS STX	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data (High Order Byte)
		7	1	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer − 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer − 2	1	Irrelevant Data (Note 1)
		7	0	Index Register	1	Irrelevant Data (Note 1)
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)

Note 1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.

Note 2. For TST, VMA = 0 and Operand data does not change.

MOTOROLA

MC6801
MC6803
MC6803NR

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

MICROCOMPUTER
MICROPROCESSOR

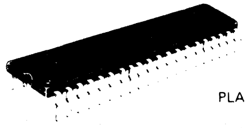
MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

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. The MC6803NR is comparable to MC6801 operating in Mode 3. An EPROM version of the MC6801, the MC68701 microcomputer, is available for systems development. The MC68701 is pin and code compatible with the MC6801/03/03NR and can be used to emulate the MC6801/03/03NR. The MC68701 is described in a separate Advance Information publication. MC6801 MCU Family features include:

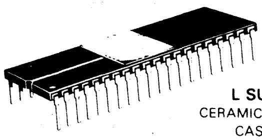
- Enhanced MC6800 Instruction Set
- 8×8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatibility with the M6800
- 16-Bit Three-Function Programmable Timer
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Bus Compatibility with the M6800 Family
- 2048 Bytes of ROM (MC6801)
- 128 Bytes of RAM (MC6801 and MC6803)
- 64 Bytes of RAM Retainable During Powerdown (MC6801 and MC6803)
- 29 Parallel I/O and Two Handshake Control Lines
- Internal Clock Generator with Divide-by-Four Output

PART NUMBER DESIGNATED BY SPEED

MC6801 **MC6803** **MC6803NR** **(1.0 MHz)**	**MC68A01** **MC68A03** **MC68A03NR** **(1.5 MHz)**
MC6801-1 **MC6803-1** **MC6803NR-1** **(1.25 MHz)**	**MC68B01** **MC68B03** **MC68B03NR** **(2.0 MHz)**

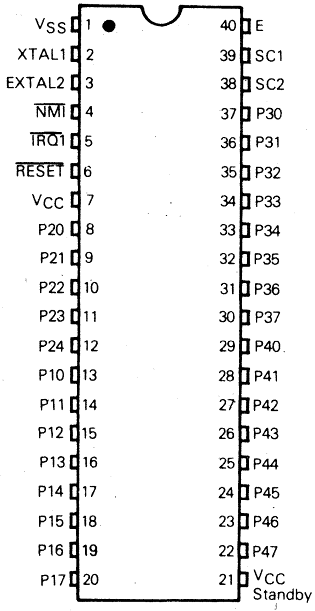


G SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

FIGURE 1 — PIN ASSIGNMENT



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

*An extended temperature device, the MC6801C is available with $T_A = -40°C$ 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 $V_{SS} \leq (V_{in}$ 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	θ_{JA}	50	°C/W
Ceramic		50	

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	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}	3.579	4.0	3.579	5.0	3.579	6.0	3.579	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

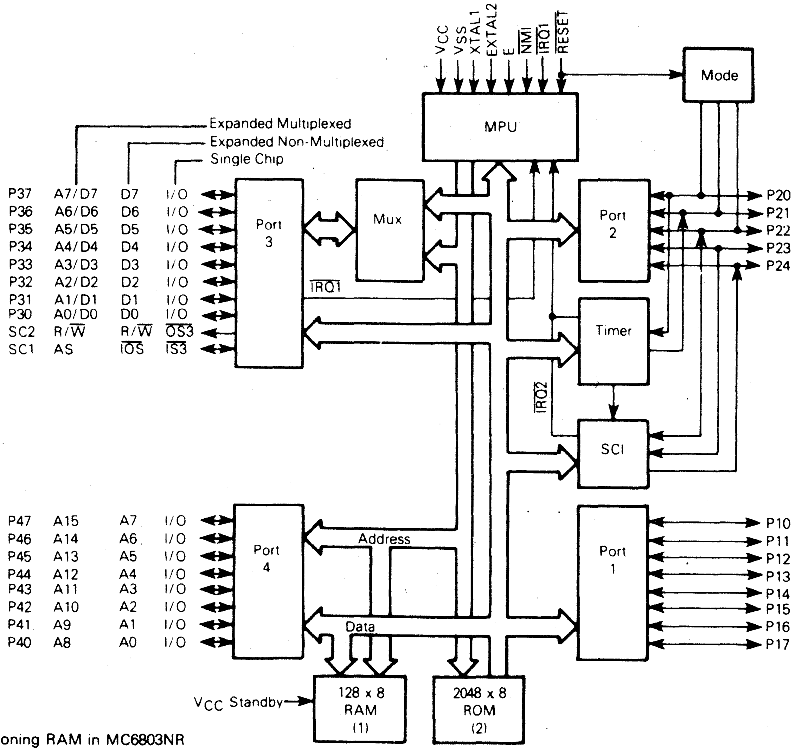
DC ELECTRICAL CHARACTERISTICS (V_{CC}=5.0 Vdc ±5% V_{SS}=0, T_A=0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{\text{RESET}}$	V_{IH}	$V_{SS}+4.0$	–	V_{CC}	V
	Other Inputs*		$V_{SS}+2.0$	–	V_{CC}	
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Load Current (V_{in}=0 to 2.4 V)	Port 4	I_{in}	–	–	0.5	mA
	SC1		–	–	0.8	
Input Leakage Current (V_{in}=0 to 5.25 V)	$\overline{\text{NMI}}$, $\overline{\text{IRQ1}}$, $\overline{\text{RESET}}$	I_{in}	–	1.5	2.5	µA
Three-State (Off State) Input Current (V_{in}=0.5 to 2.4 V)	P10-P17, P30-P37	I_{TSI}	–	2.0	10	µA
	P20-P24		–	10.0	100	
Output High Voltage (I_{load}= −100 µA, V_{CC}= min)	P30-P37	V_{OH}	$V_{SS}+2.4$	–	–	V
(I_{load}= −65 µA, V_{CC}= min)**	P40-P47, E, SC1, SC2		$V_{SS}+2.4$	–	–	
(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	2.5	10.0	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	Powerdown	V_{SBB}	4.0	–	5.25	V
	Powerup	V_{SB}	4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	–	–	6.0	mA

*Except Mode Programming Levels; See Figure 16.

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

FIGURE 2 — M6801 MICROCOMPUTER FAMILY BLOCK DIAGRAM

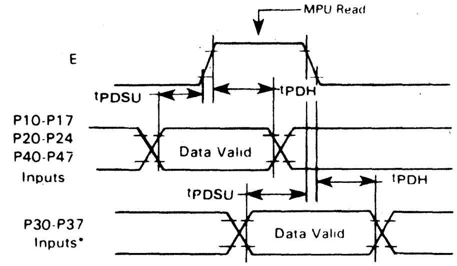


(1) No functioning RAM in MC6803NR
(2) No functioning ROM in MC6803 and MC6803NR

PERIPHERAL PORT TIMING (Refer to Figures 3-6)

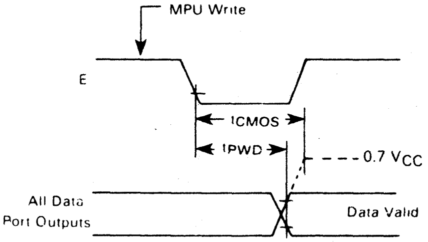
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid Port 1 Port 2, 3, 4	t_{PWD}	 – –	 – –	 350 350	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 3 — DATA SETUP AND HOLD TIMES (MPU READ)



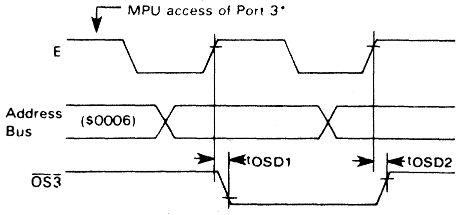
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

FIGURE 4 — DATA SETUP AND HOLD TIMES (MPU WRITE)



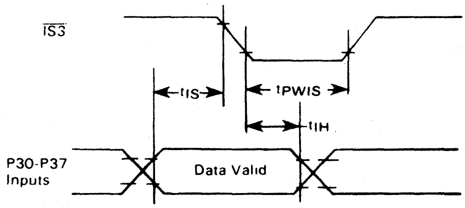
NOTES
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

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



*Access matches Output Strobe Select (OSS = 0, a read, OSS = 1, a write)

FIGURE 6 — 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.

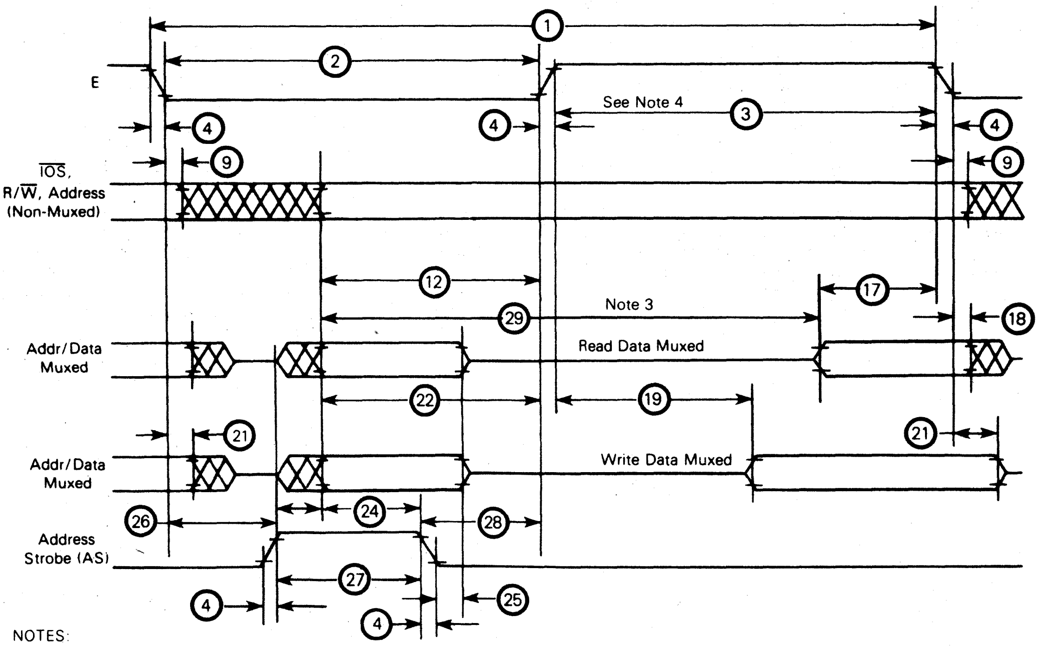
BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6801 MC6803 MC6803NR		MC6801-1 MC6803-1 MC6803NR-1		MC68A01 MC68A03 MC68A03NR		MC68B01 MC68B03 MC68B03NR		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* (See Note 9)	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 ms min. (1.0 MHz devices), 80 ms min. (1.25 MHz device), 65 ms min. (1.5 MHz devices), 50 ms min. (2.0 MHz devices).

FIGURE 7 — BUS TIMING



NOTES:

1. Voltage levels shown are $V_L \le 0.5$ V, $V_H \ge 2.4$ 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 8 — CMOS LOAD

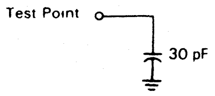
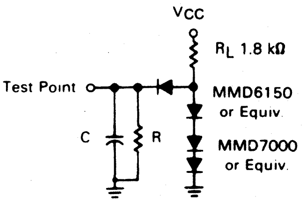


FIGURE 9 — TIMING TEST LOAD PORTS 1, 2, 3, 4



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

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.

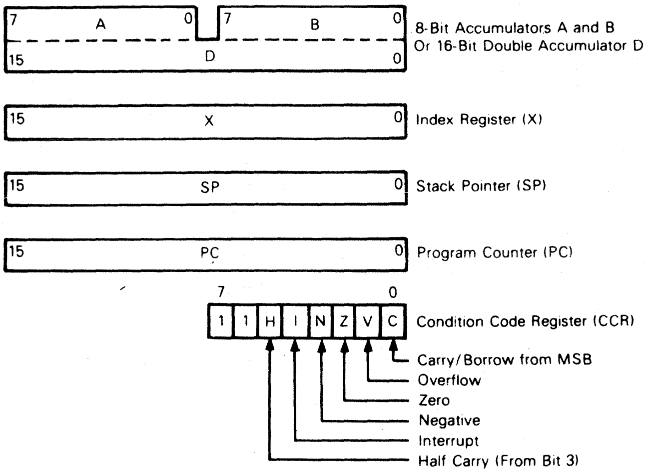
Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

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 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 10, 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. The MC6803NR is comparable to an MC6801 that operates in Mode 3 only.

FIGURE 10 — PROGRAMMING MODEL



OPERATING MODES

The MC6801 provides eight different operating modes (Modes 0 through 7), the MC6803 provides two operating modes (Modes 2 and 3), and the MC6803NR provides one operating mode (Mode 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 is Mode 5 and the remaining five 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 11. 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 12.

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 13 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 MC6801/03/03NR 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; the MC6803NR operates only in Mode 3.

FIGURE 11 — SINGLE-CHIP MODE

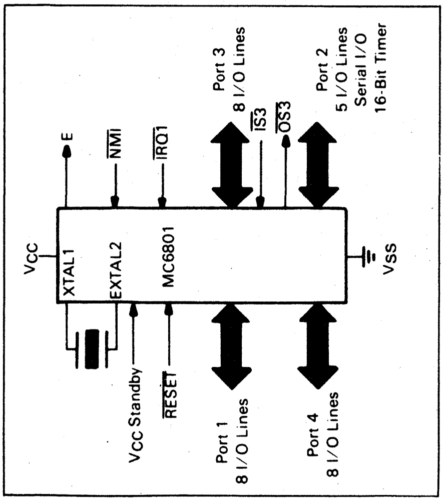


FIGURE 12 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION

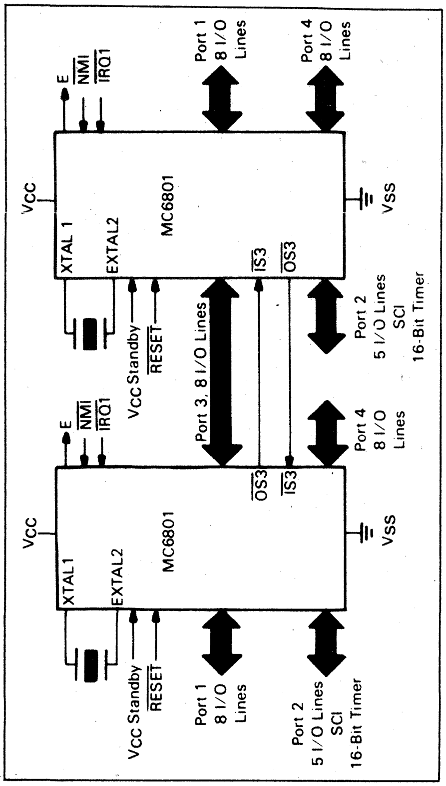
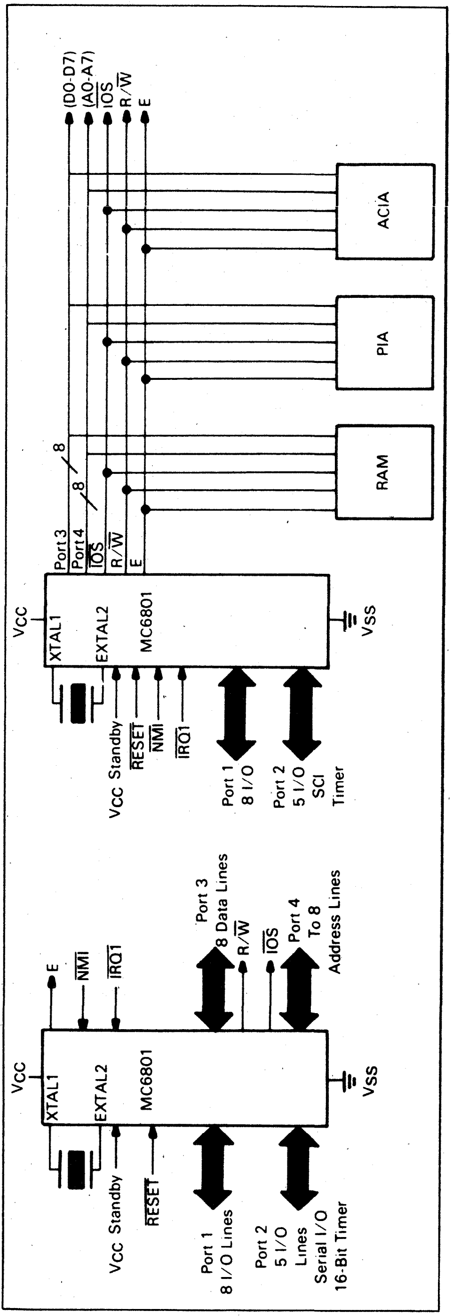


FIGURE 13 — EXPANDED NON-MULTIPLEXED CONFIGURATION



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 multplexed modes. The MC6803 operates only in Modes 2 and 3, while the MC6803NR operates only in Mode 3.

Figure 14 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 15. 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 16. 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 17 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	NMUX(5, 6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed/No RAM or ROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed Test

Legend:

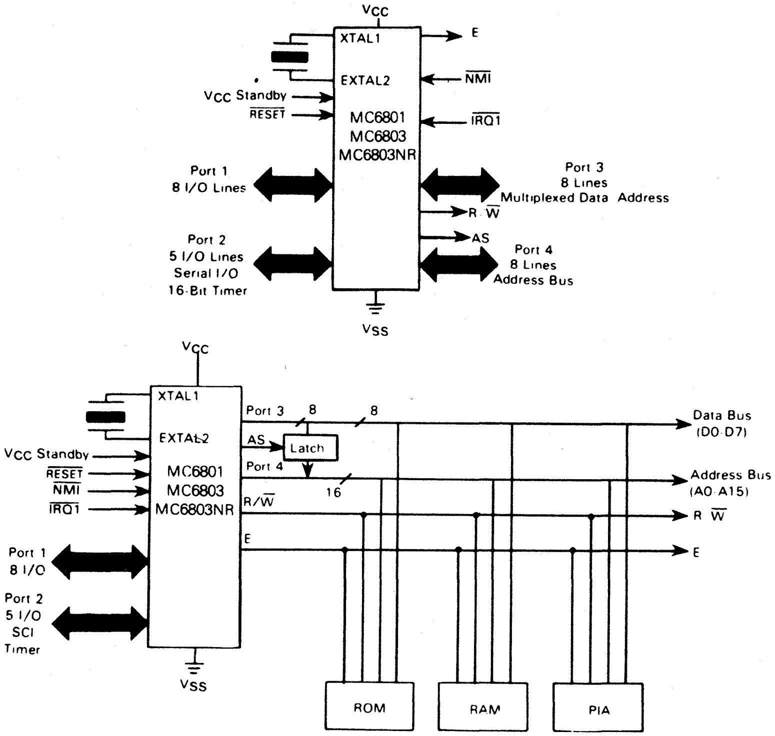
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic "0"
H — Logic "1"

Notes:

(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) $\overline{\text{RESET}}$ vector is external for 2 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; the MC6803NR operates only in Mode 3.

FIGURE 14 — 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 15 — TYPICAL LATCH ARRANGEMENT

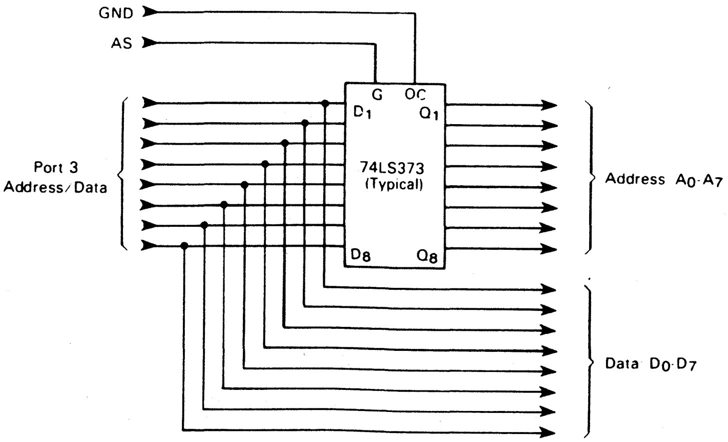
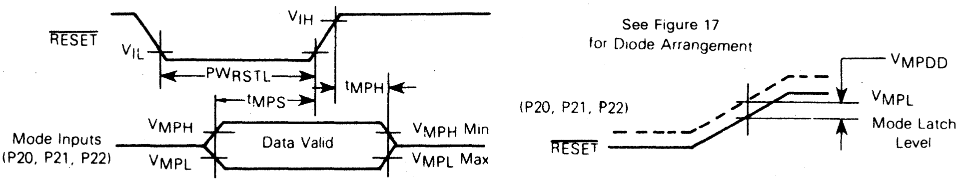


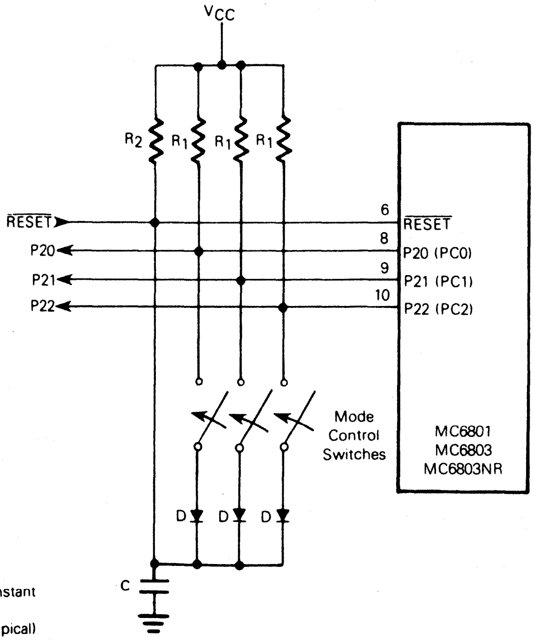
FIGURE 16 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 16)

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

FIGURE 17 — TYPICAL MODE PROGRAMMING CIRCUIT



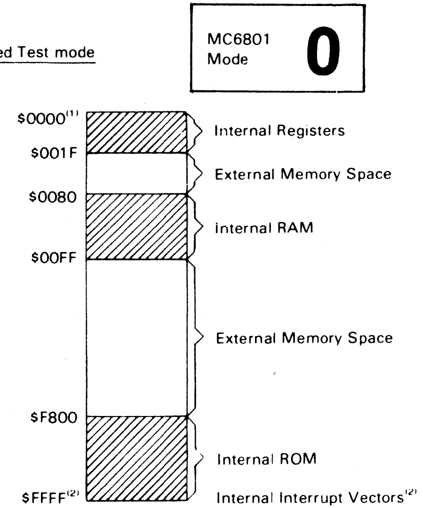
Notes:
1. Mode 7 as shown
2. $R_2 \bullet C$ = Reset time constant
3. R_1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

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 18. In Modes 1R and 6R, the MC6801 ROM has been relocated by a mask option. The first 32 locations of each map are reserved for the internal register area, as shown in Table 4, with exceptions as indicated.

FIGURE 18 — MC6801/03/03NR MEMORY MAPS

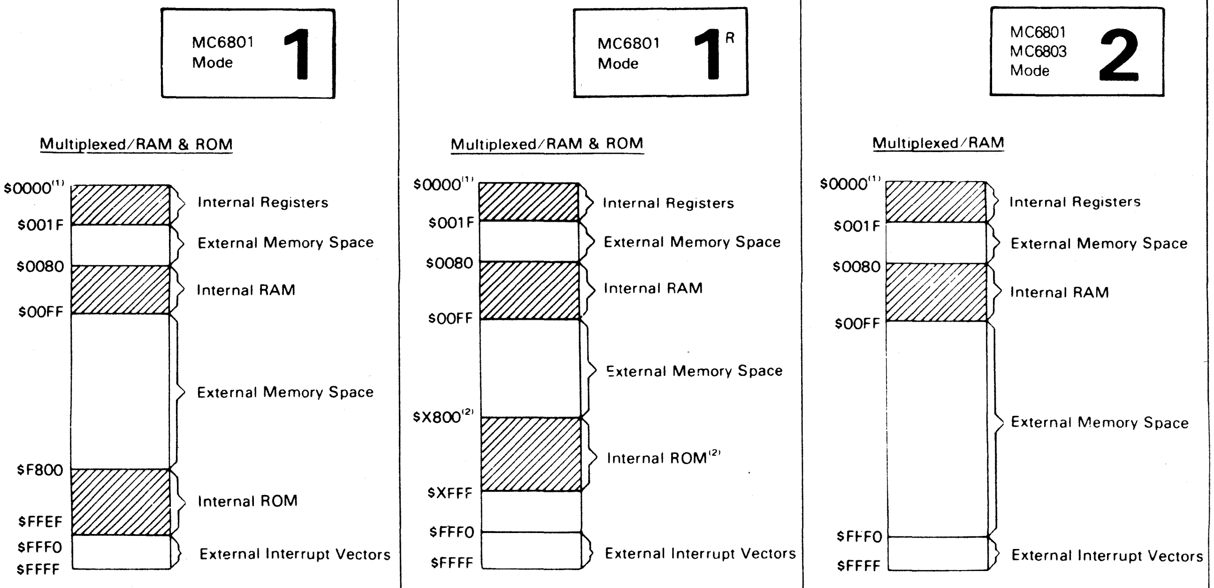
Multiplexed Test mode



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 2 cycles after a positive edge of $\overline{\text{RESET}}$ and internal at all other times.
3) After 2 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.

FIGURE 18 — MC6801/03/03NR MEMORY MAPS (CONTINUED)

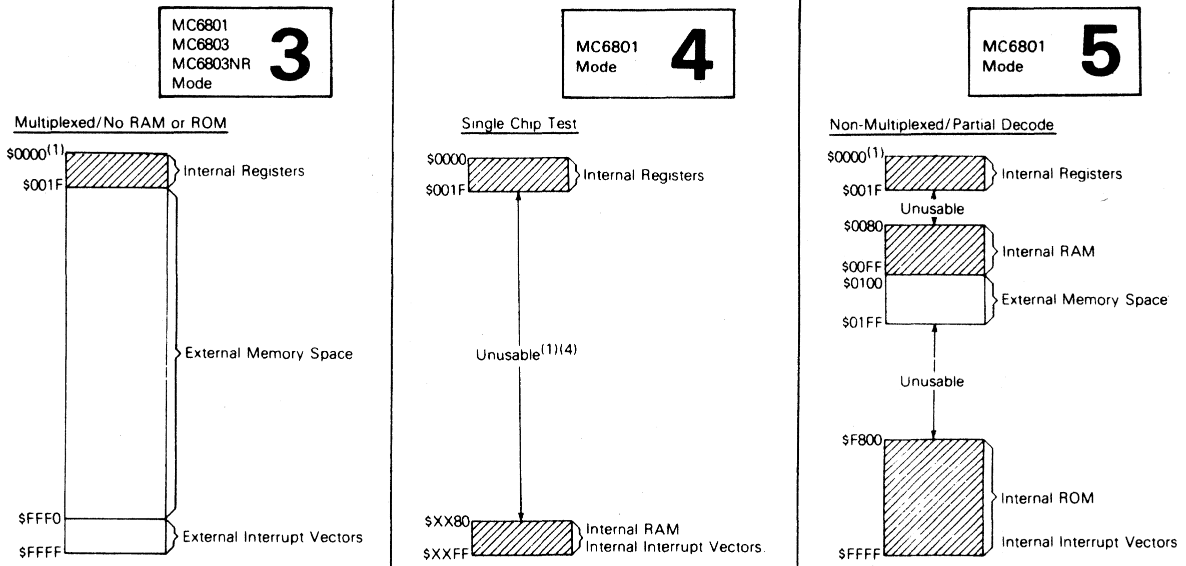


Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Internal ROM addresses $FFF0 to $FFFF are not usable.

Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.
2) Starting addresses for the internal ROM may be $C800, $D800 or $E800 as a mask option.

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

FIGURE 18 — MC6801/03/03NR MEMORY MAPS (CONTINUED)



Notes:

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

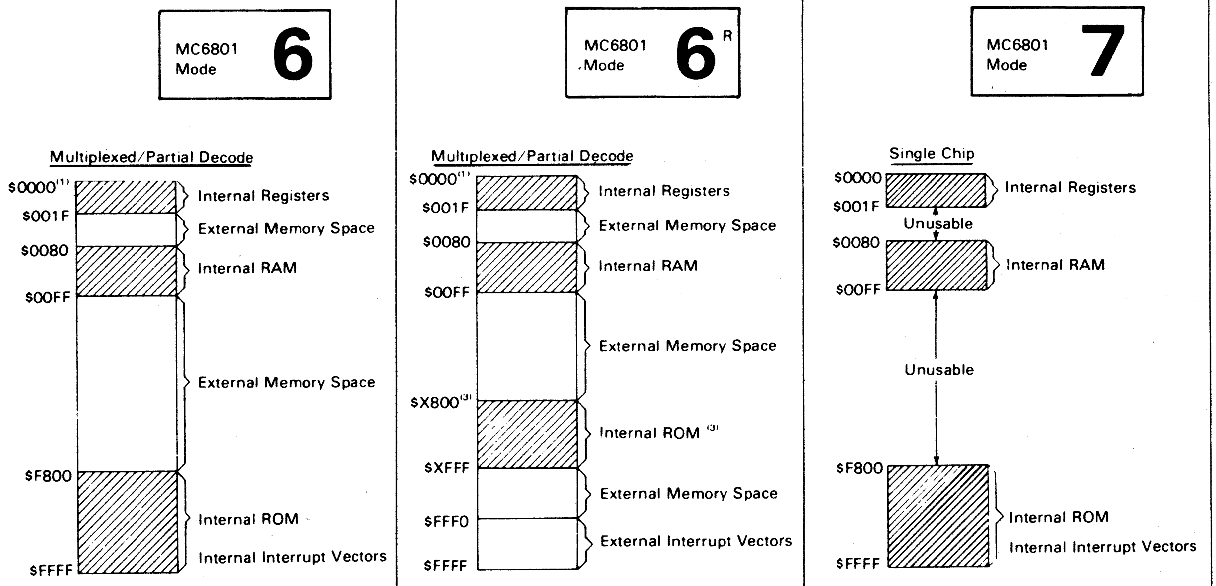
Notes:

1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert 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.

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 $\overline{\text{RESET}}$ 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 18 — MC6801/03/03NR 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) 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.
3) Starting addresses for the internal ROM may be $C800, $D800 or $E800.

MC6801/03/03NR INTERRUPTS

The M6801 Family supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt ($\overline{\text{NMI}}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$. The Programmable Timer and Serial Communications Interface use an internal $\overline{\text{IRQ2}}$ interrupt line, as shown in Figure 2. External devices (and IS3) use $\overline{\text{IRQ1}}$. An $\overline{\text{IRQ1}}$ interrupt is serviced before $\overline{\text{IRQ2}}$ if both are pending.

All 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 19 and is common to every interrupt excluding reset. During interrupt servicing the Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and $\overline{\text{RESET}}$ timing are illustrated in Figures 20 and 21.

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 $\overline{\text{RESET}}$ reaches 4.0 volts. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

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

TABLE 4 — INTERNAL REGISTER AREA

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

*External addresses in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No $\overline{\text{IOS}}$)

**External addresses in Modes 0, 1, 2, 3

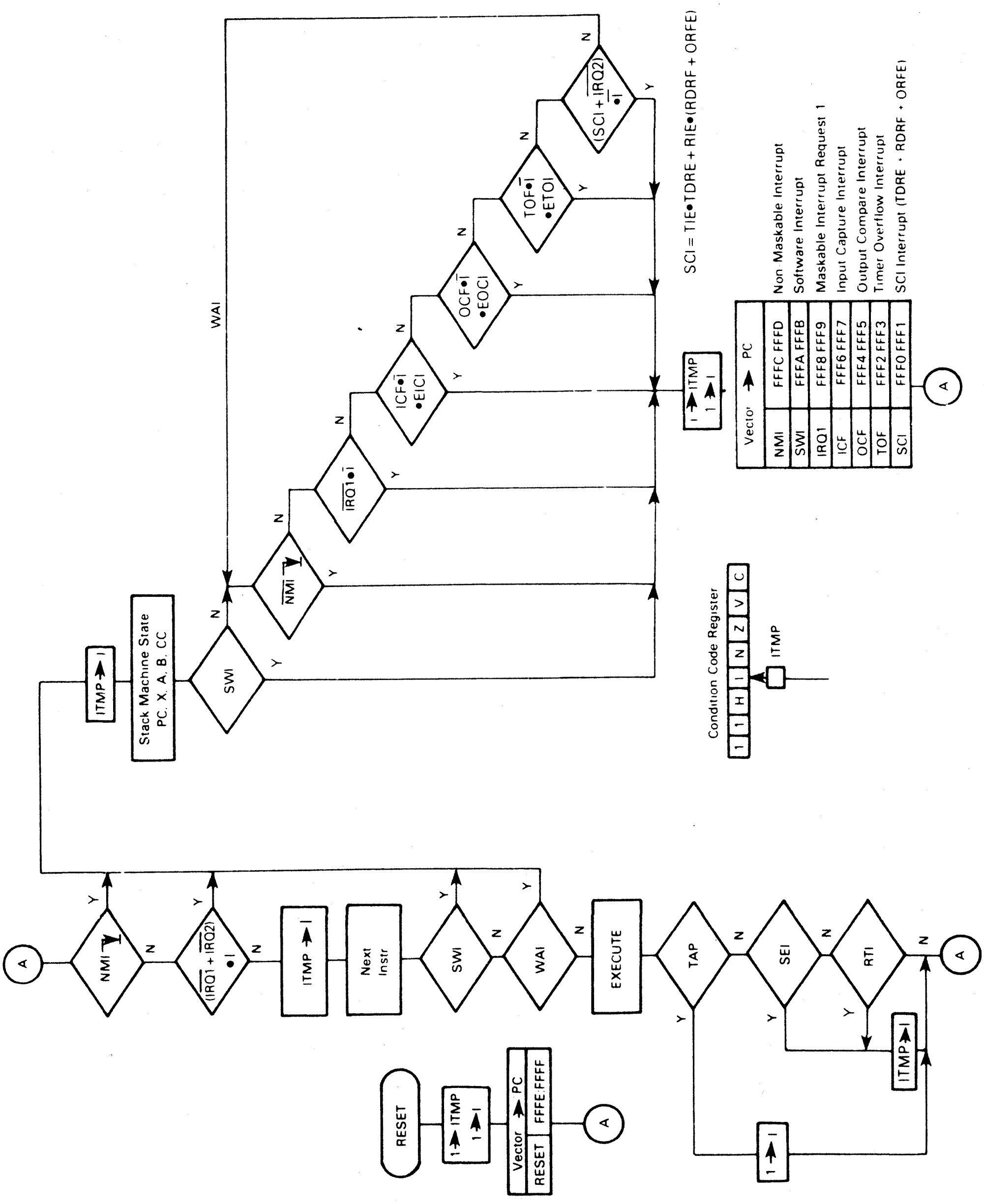
***1 = Output, 0 = Input

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

*IRQ2 Interrupt

FIGURE 19 — INTERRUPT FLOWCHART



Vector	→ PC
RESET	FFFE FFFF

Vector	→ PC
NMI	FFFC FFFD
SWI	FFFA FFFB
IRQ1	FFF8 FFF9
ICF	FFF6 FFF7
OCF	FFF4 FFF5
TOF	FFF2 FFF3
SCI	FFF0 FFF1

SCI = TIE•TDRE + RIE•(RDRF + ORFE)

NMI Non Maskable Interrupt
SWI Software Interrupt
IRQ1 Maskable Interrupt Request 1
ICF Input Capture Interrupt
OCF Output Compare Interrupt
TOF Timer Overflow Interrupt
SCI SCI Interrupt (TDRE + RDRF + ORFE)

FIGURE 20 — INTERRUPT SEQUENCE

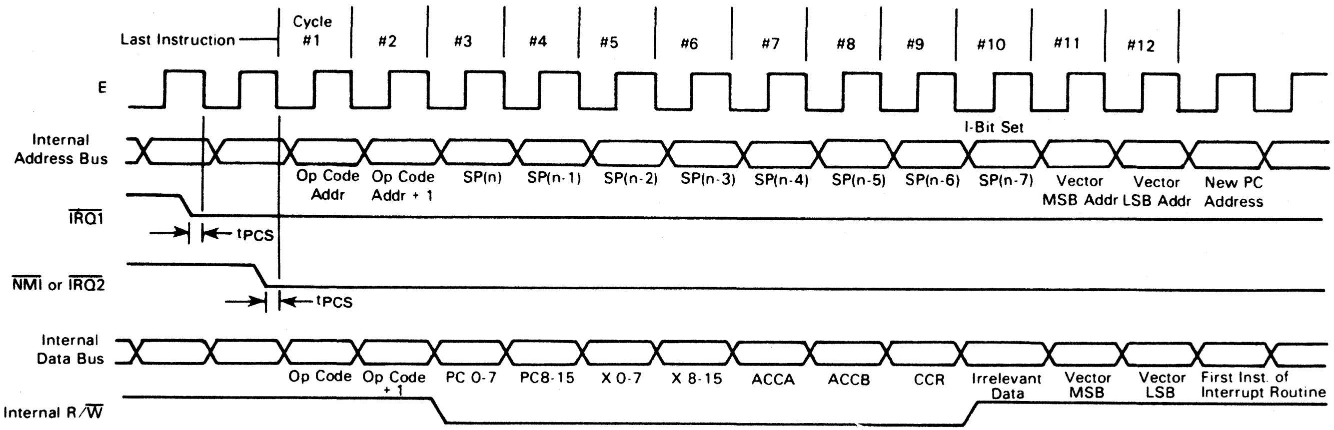
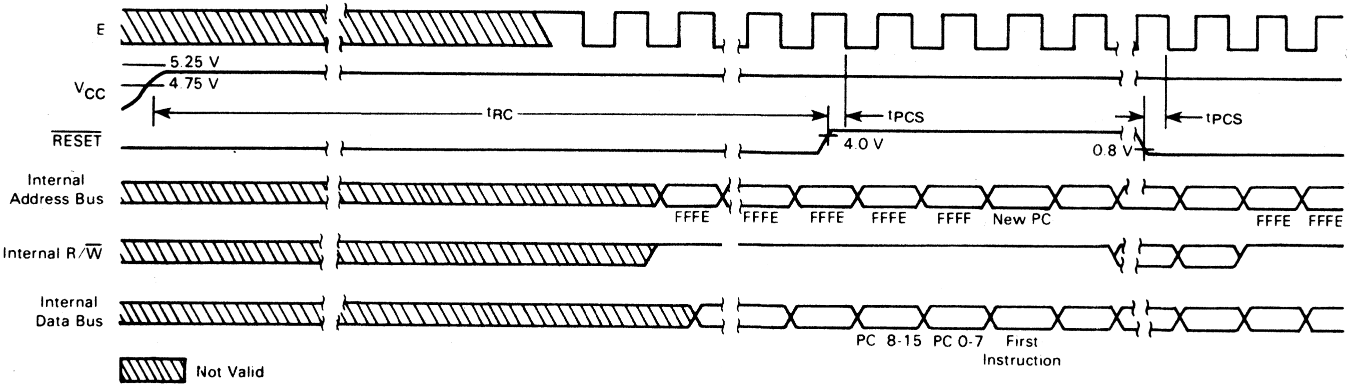


FIGURE 21 — RESET TIMING



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 22.

$\overline{\text{RESET}}$

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.

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

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.

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

$\overline{\text{IRQ1}}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the 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.

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 $\overline{\text{IS3}}$ 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 description. 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 5.

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

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 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	$0003

---

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

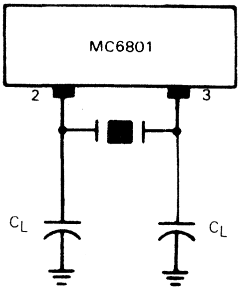
FIGURE 22 — M6801 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

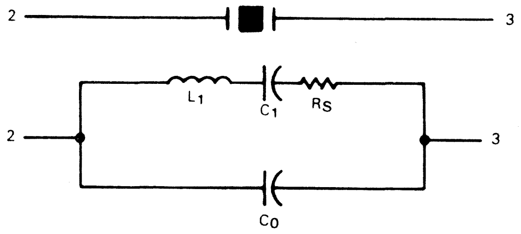
Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	5.0 MHz	6.0 MHz	8.0 MHz
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 cut may also be used.



$C_L = 20$ pF (typical)

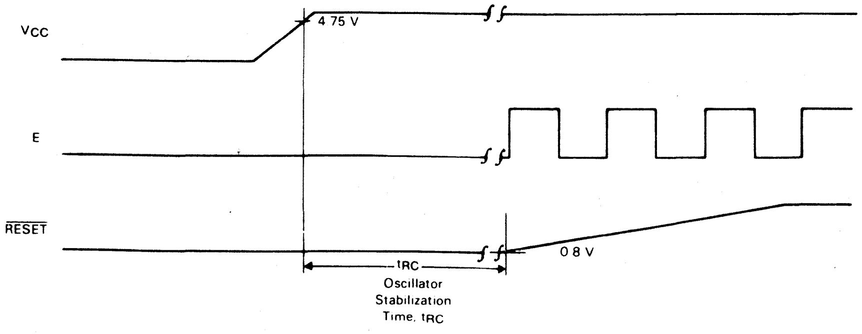


Equivalent Circuit

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})



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, $\overline{\text{IS3}}$ and $\overline{\text{OS3}}$, 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 $\overline{\text{IS3}}$ as a control signal, (2) $\overline{\text{OS3}}$ 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 $\overline{\text{IS3}}$ negative edge. Port 3 latch timing is shown in Figure 6.

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 $\overline{\text{IS3}}$ negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENALBLE is cleared during reset.
Bit 4	OSS (Output Strobe Select). This bit determines whether $\overline{\text{OS3}}$ 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 $\overline{\text{IRQ1}}$ 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 $\overline{\text{IS3}}$ 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.

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 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-board ROM and 128 bytes of on-board RAM.

One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} 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 is cleared whenever V_{CC} Standby decreases below V_{SBB} (min). It can be set only by software and is not affected during reset.

PROGRAMMABLE TIMER

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 23.

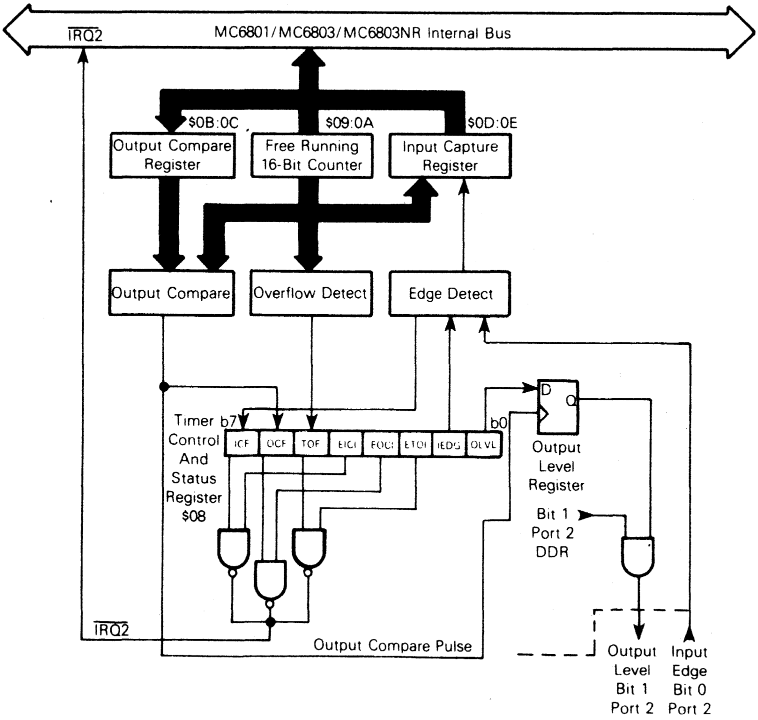
COUNTER ($09:0A)

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

FIGURE 23 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P21 and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF at $\overline{\text{RESET}}$.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20 even when configured as an output. An input capture can occur independently of ICF: the register always 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 occured 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 $\overline{\text{IRQ2}}$ 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 1's. 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 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 (×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 24. 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)

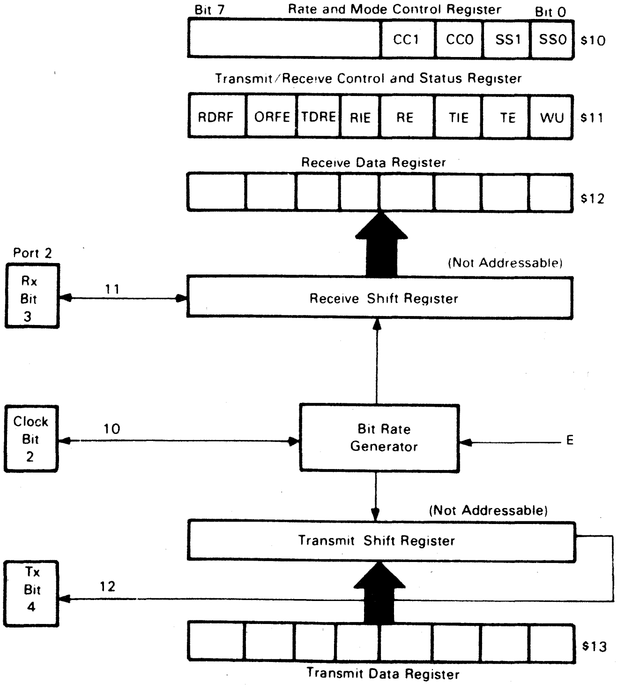
7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$0010

Bit 1:Bit 0	SS1:SS0 Speed Select. These two bits select the Baud 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.

FIGURE 24 — 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 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 $\overline{\text{IRQ2}}$ 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.

---

*Devices made with mask numbers M5G, M8D, and T5P do not transfer unframed data to the Receive Data Register.

TABLE 6 — SCI BIT TIMES AND RATES

SS1:SS0		$4f_o$ →	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

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 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 25.

INSTRUCTION SET

The MC6801/03/03NR 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 MC6801/03/03NR is shown in Figure 11. 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.

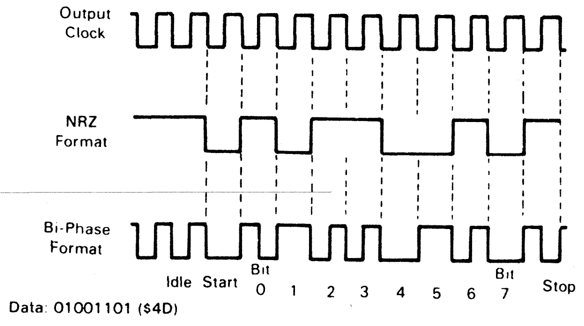
Stack Pointer – The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

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 25 – SCI DATA FORMATS



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, 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 26.

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	40	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	EF	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	CD	LDX	IMMED	3	3		*UNDEFINED OP CODE			
33	PULB	↓	4	1	67	ASR	INDXD	6	2	9B	ADDA		3	2	CF	•								

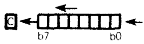
NOTES:

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

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

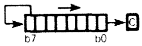
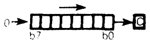
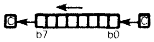
Pointer Operations	Mnemonic	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X - M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09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

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS

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		●	●	↕	↕	↕	↕

— Continued —

TABLE 10 – ACCUMULATOR AND MEMORY INSTRUCTIONS (CONTINUED)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The Condition Code Register notes are listed after Table 10

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

		Direct			Relative			Index			Extnd			Inherent				Cond. Code Reg.					
																		5	4	3	2	1	0
Operations	**Mnemonic**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**OP**	**~**	**#**	**Branch Test**	**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										See Special Operations - Figure 26	●	●	●	●	●	●
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 26	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

	Inherent					Cond. Code Reg.					
						5	4	3	2	1	0
Operations	**Mnemonic**	**OP**	**~**	**#**	**Boolean Operation**	**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
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
$\overline{M}$	Complement of M
→	Transfer Into
0	Bit = Zero
00	Byte = Zero

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E-CYCLES

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

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

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

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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

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 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

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)

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

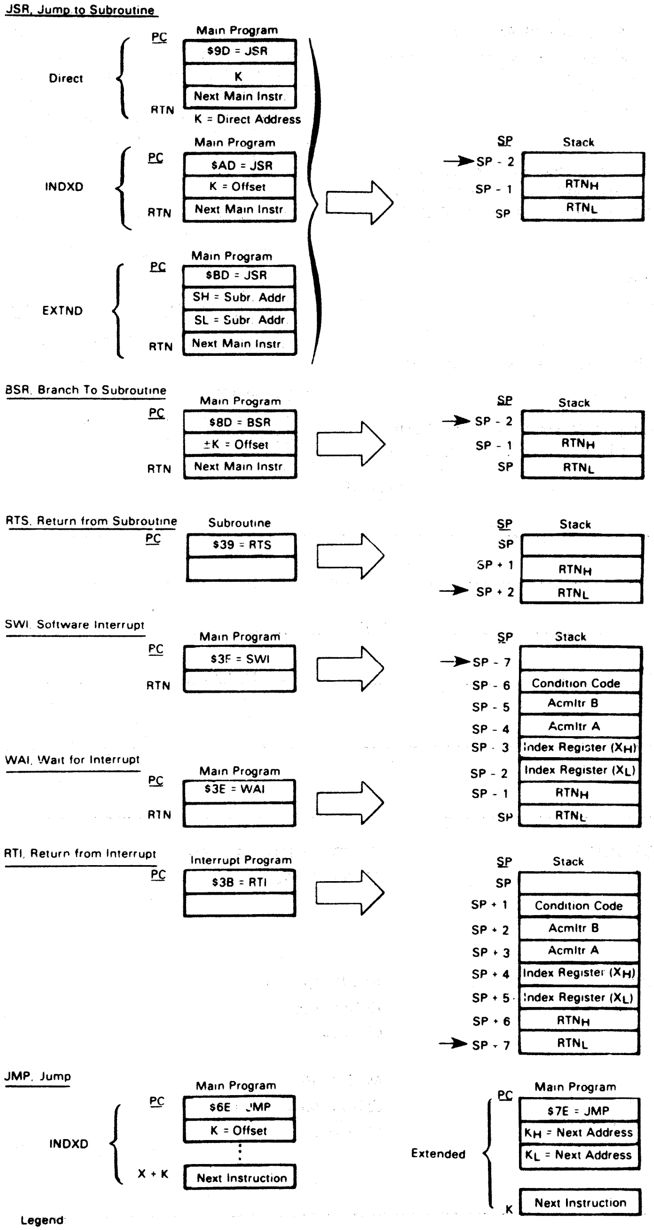
Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
RTS	5	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Address of Next Instruction (High Order Byte)
		5	Stack Pointer +2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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONCLUDED)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	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 26 — 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.
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 ($X800-$XFFF). 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.

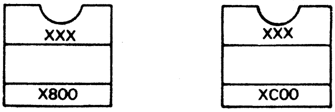


FIGURE A-1 XXX = Customer ID

A.2 DISKETTE (MDOS)

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

FIGURE A-2

CUSTOMER NAME ______________________________

ADDRESS ______________________________

STATE ______________ CITY ______________ ZIP ________

PHONE (____) ______________ EXTENSION ______________

CONTACT MS/MR ______________________________

CUSTOMER PART # ______________________________

ROM START ADDRESS OPTION
- ☐ $C800
- ☐ $D800
- ☐ $E800
- ☐ $F800
- ☐ A12 and A13 don't care

TEMPERATURE RANGE
- ☐ 0° to 70°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

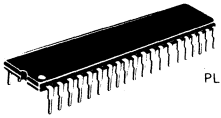
MICROCOMPUTER/MICROPROCESSOR (MCU/MPU)

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.

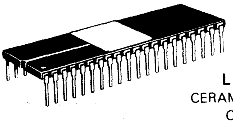
- Enhanced MC6800 Instruction Set
- Upward Source and Object Code Compatibility with the MC6800 and MC6801
- Bus Compatibility with the M6800 Family
- 8 × 8 Multiply Instruction
- Single-Chip or Expanded Operation to 64K Byte Address Space
- Internal Clock Generator with Divide-by-Four Output
- Serial 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
- $\overline{\text{NMI}}$ Inhibited Until Stack Load

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

Package Type	Frequency	Order Number
Plastic – G Suffix	1.0 MHz 1.25 MHz	MC6801U4G/MC6803U4G MC6801U4G-1/MC6803U4G-1
Ceramic – L Suffix	1.0 MHz 1.25 MHz	MC6801U4L/MC6803U4L MC6801U4L-1/MC6803U4L-1

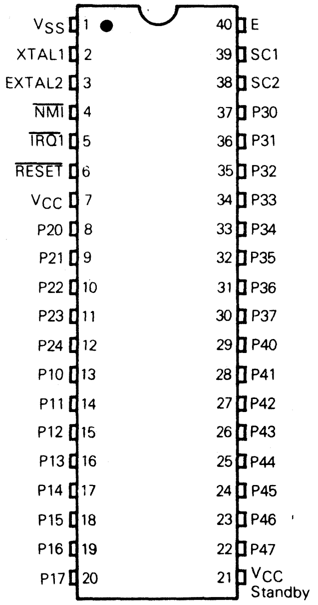


G SUFFIX
PLASTIC PACKAGE
CASE 711



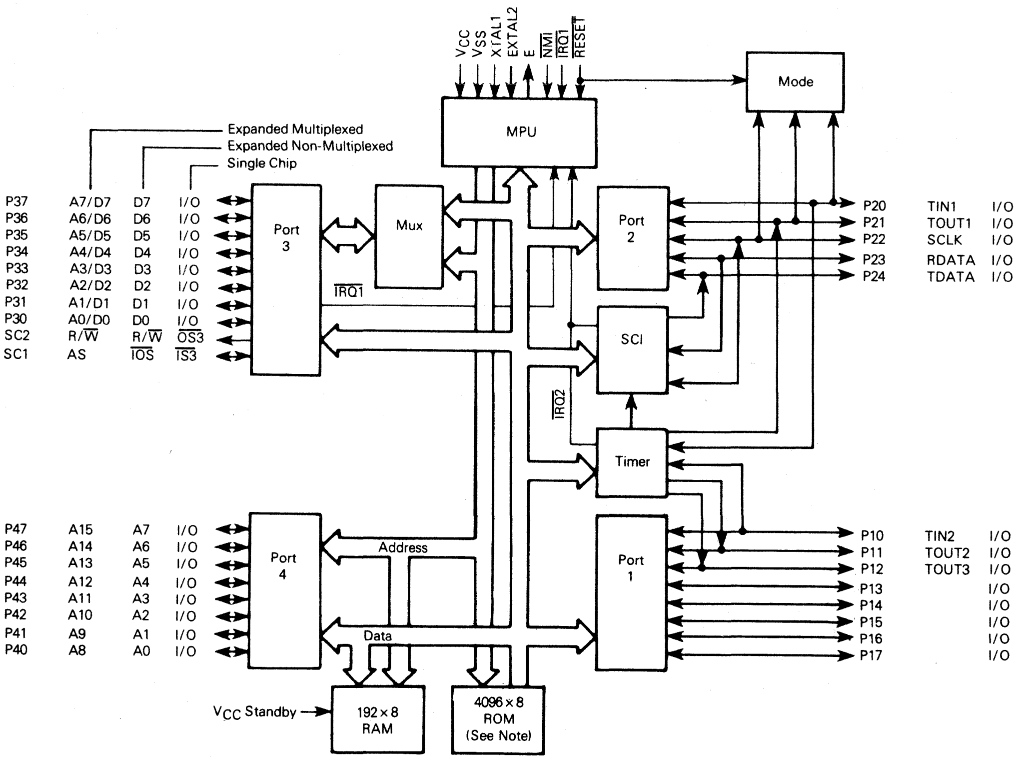
L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT



ADI-896

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

THERMAL CHARACTERISTICS

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

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 $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{CC}$. Input protection is enhanced by connecting unused inputs to either V_{DD} or V_{SS}.

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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}	3.579	4.0	3.579	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	$\overline{\text{RESET}}$ 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 SC1	I_{in}	– –	– –	0.5 0.8	mA
Input Leakage Current (V_{in} = 0 to 5.25 V)	$\overline{\text{NMI}}$, $\overline{\text{IRQ1}}$, $\overline{\text{RESET}}$	I_{in}	–	1.5	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	P10-P17, P30-P37 P20-P24	I_{TSI}	– –	2.0 10.0	10 100	μA
Output High Voltage (I_{load} = −100 μA, V_{CC} = min) (I_{load} = −65 μA, V_{CC} = min)** (I_{load} = −100 μA, V_{CC} = min)	E, P30-P37 P40-P47, SC1, SC2 Other Outputs	V_{OH}	$V_{SS}+2.4$ $V_{SS}+2.4$ $V_{SS}+2.4$	– – –	– – –	V
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Darlington Drive Current (V_O = 1.5 V)	P10-P17	I_{OH}	1.0	2.5	10.0	mA
Internal Power Dissipation (Measured at T_A = 0°C in Steady-State Operation)		P_{INT}	–	–	1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_O = 1.0 MHz)	P30-P37, P40-P47, SC1 Other Inputs	C_{in}	– –	– –	12.5 10.0	pF
V_{CC} Standby	Power Down Power Up	V_{SBB} V_{SB}	4.0 4.75	– –	5.25 5.25	V
Standby Current	Power Down	I_{SBB}	–	–	6.0	mA

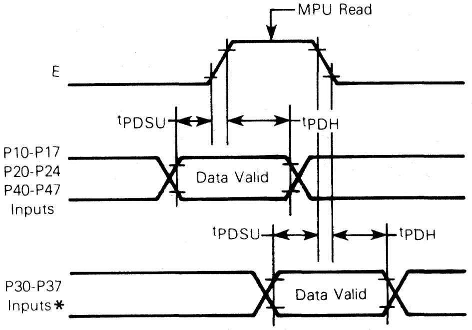
*Except mode programming levels; see Figure 16.

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

PERIPHERAL PORT TIMING (Refer to Figures 1-4)

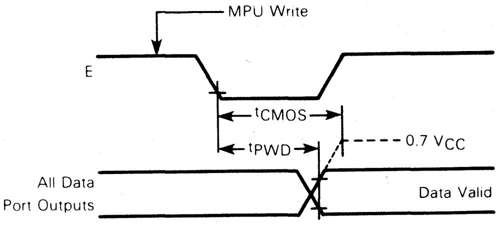
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{\text{OS3}}$ Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{\text{OS3}}$ Positive Transition	t_{OSD2}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid Port 1 Port 2, 3, 4	t_{PWD}	 – –	 – –	 350 350	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 (MC6801U4 SINGLE-CHIP MODE)

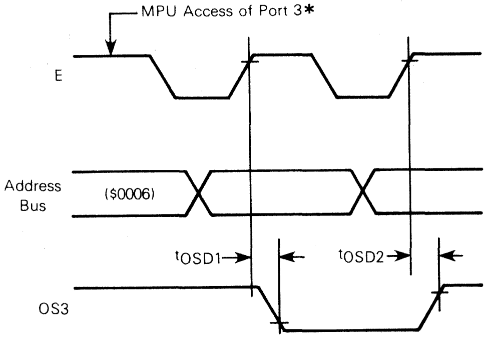
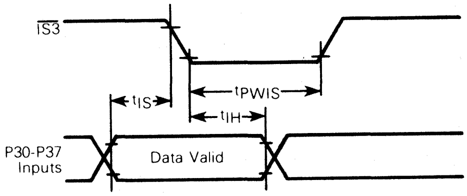


FIGURE 4 — PORT 3 LATCH TIMING (MC6801U4 SINGLE-CHIP MODE)



*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

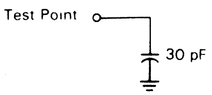
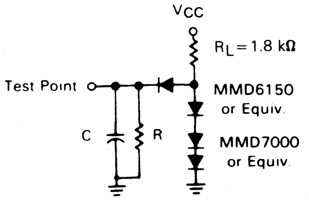


FIGURE 6 — TIMING TEST LOAD PORTS 1, 2, 3, AND 4



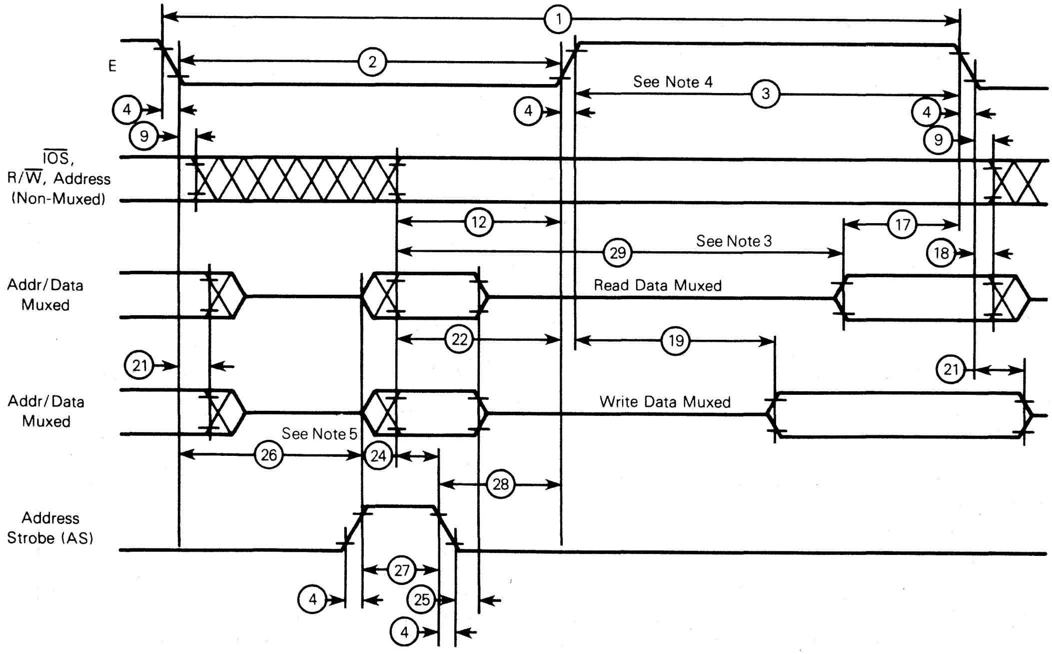
C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P10-P17, P20-P24
= 24 kΩ for P30-P37, E

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}	530	–	410	–	ns

*At specified cycle time.

FIGURE 7 — BUS TIMING



NOTES:

1. Voltage levels shown are $V_L \leq 0.6$ 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 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.

FIGURE 8 — PROGRAMMING MODEL

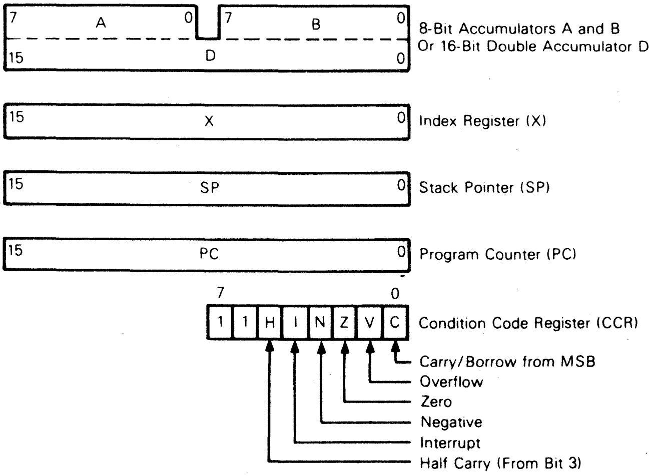


TABLE 1 — NEW INSTRUCTIONS

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.

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

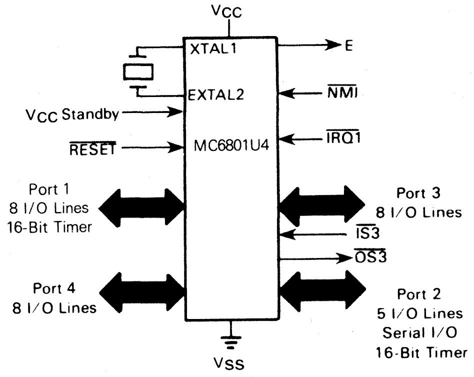
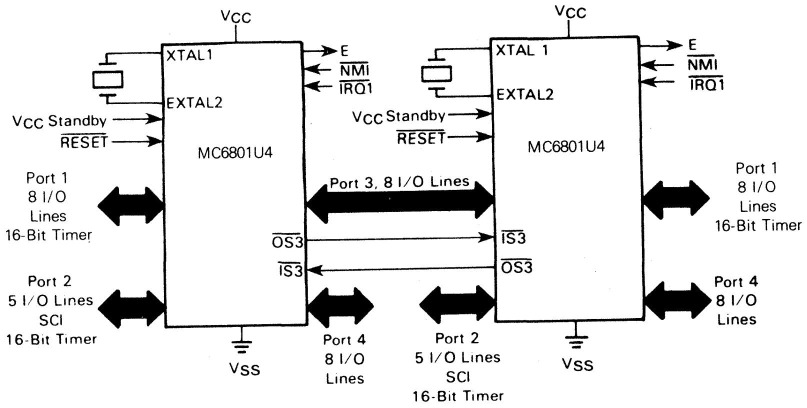


FIGURE 10 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION



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.

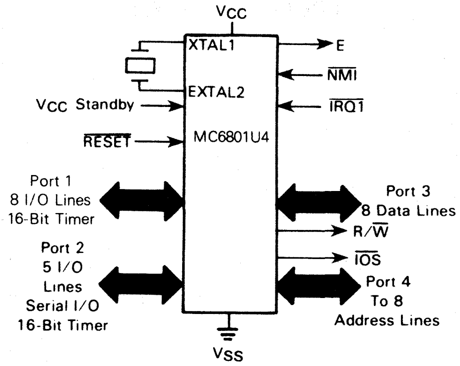
EXPANDED MULTIPLEXED MODES (0, 1, 2, 3, 6) — A 64K byte memory space is provided in the expanded multiplexed modes. In each of the expanded multiplexed modes, port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E 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



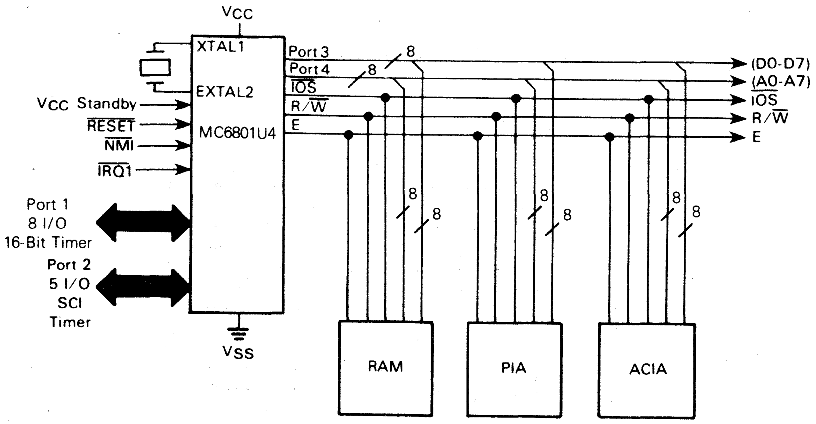
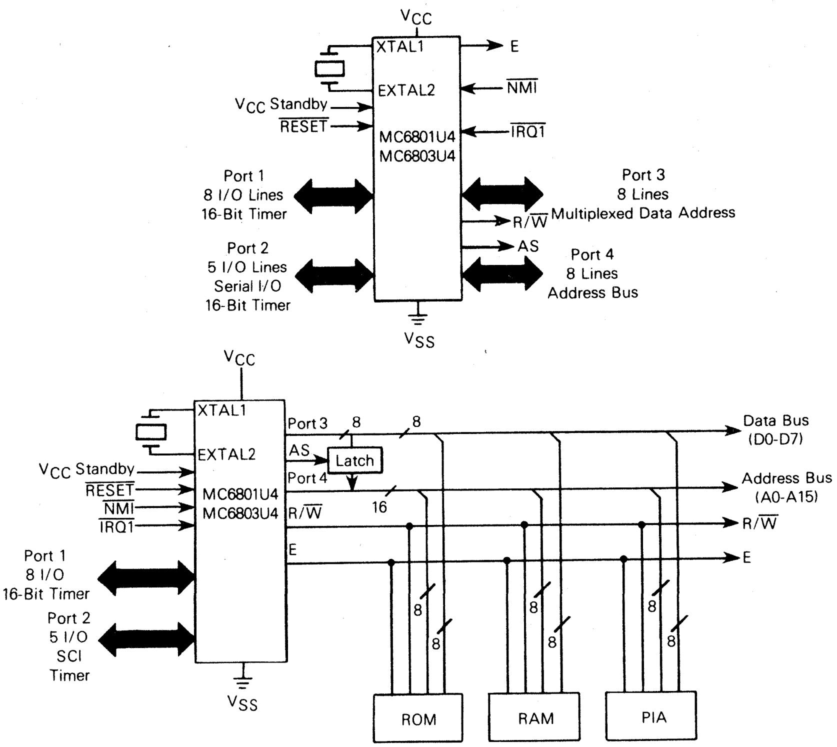
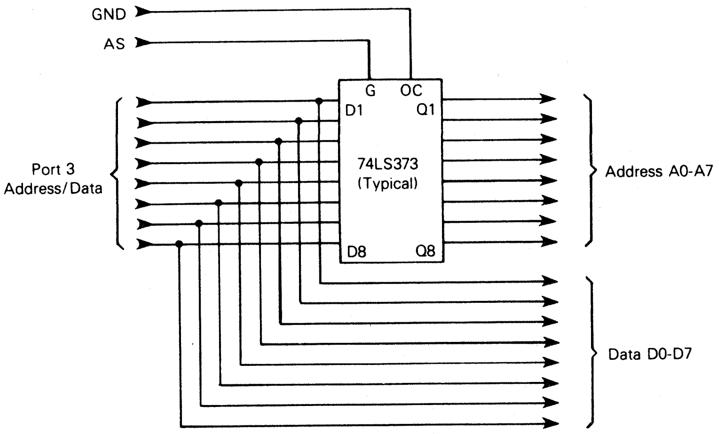


FIGURE 12 — EXPANDED MULTIPLEXED CONFIGURATION



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

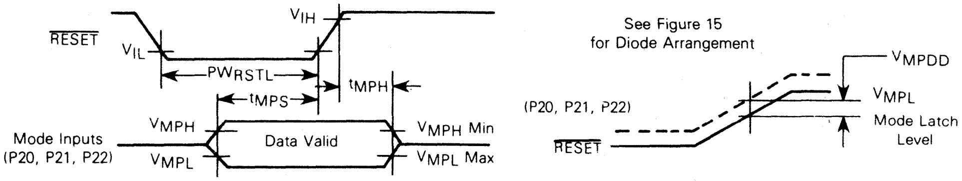
7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	\$03

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

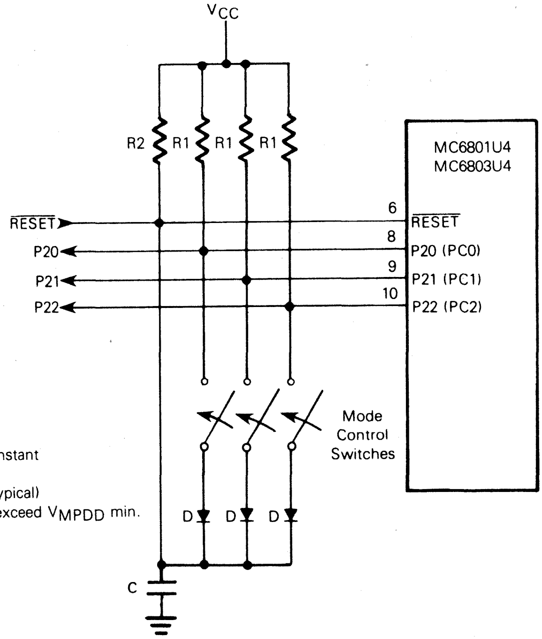
Mode*	P22 PC2	P21 PC1	P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX$^{(2,3)}$	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX$^{(2,3)}$	Non-Multiplexed/Partial Decode
4	H	L	L	–	–	–	–	Undefined$^{(4)}$
3	L	H	H	E	I	E	MUX$^{(1,5)}$	Multiplexed/RAM
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.

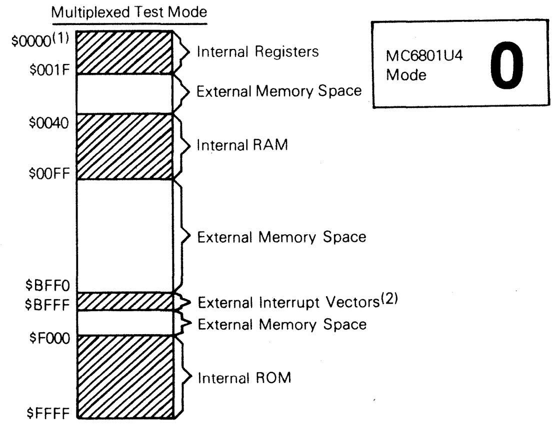
FIGURE 15 — TYPICAL MODE PROGRAMMING CIRCUIT



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.

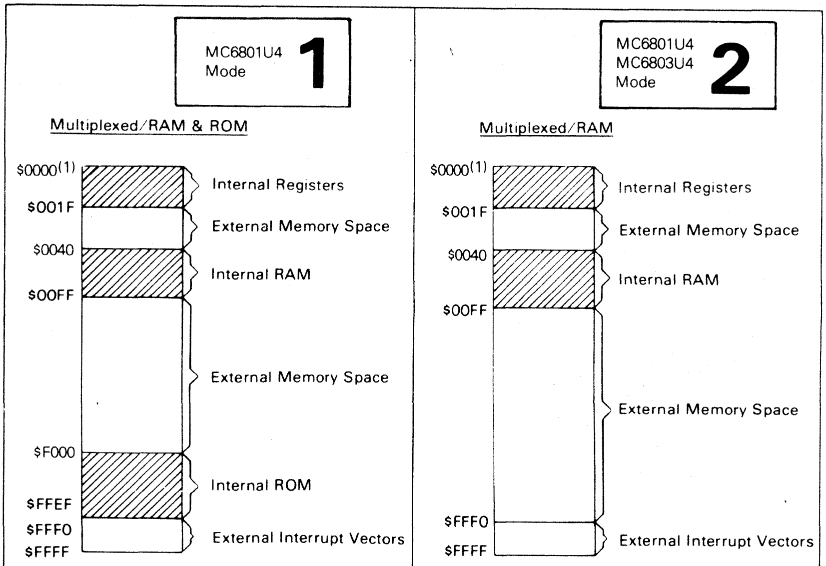
FIGURE 16 — MC6801U4/MC6803U4 MEMORY MAPS (Sheet 1 of 4)



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 1-7 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)



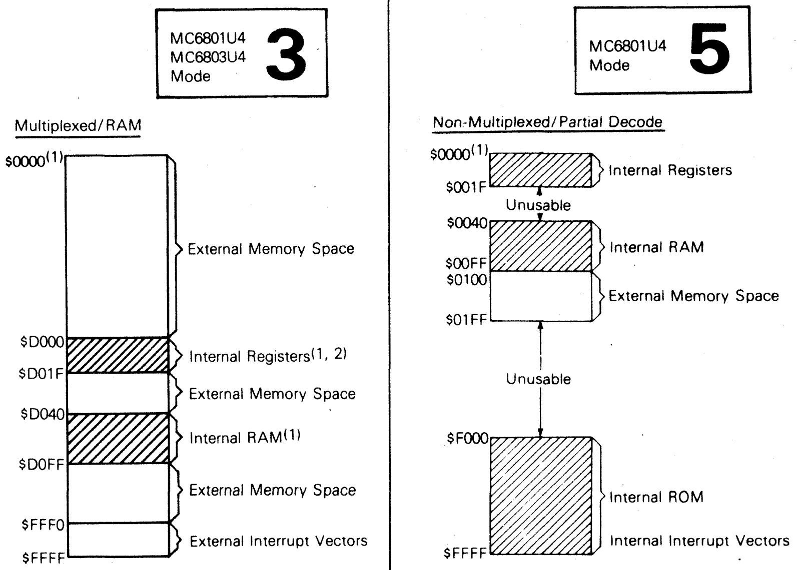
NOTES:

1) Excludes the following addresses which may be used externally: $05 and $07.
2) Internal ROM 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.

NOTE:

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

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



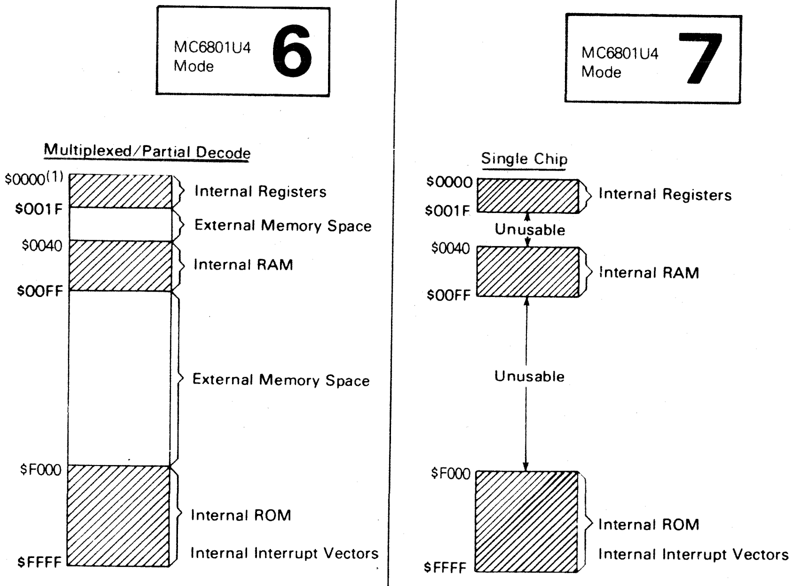
NOTES:

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.

NOTES:

1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{\text{IOS}}$).
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
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

MC6801U4/MC6803U4 INTERRUPTS

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

NOTE

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

All $\overline{\text{IRQ2}}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order and each is vectored to a separate location. All interrupt vector locations are shown in Table 5. 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 $\overline{\text{RESET}}$ timing are illustrated in Figures 18 and 19.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

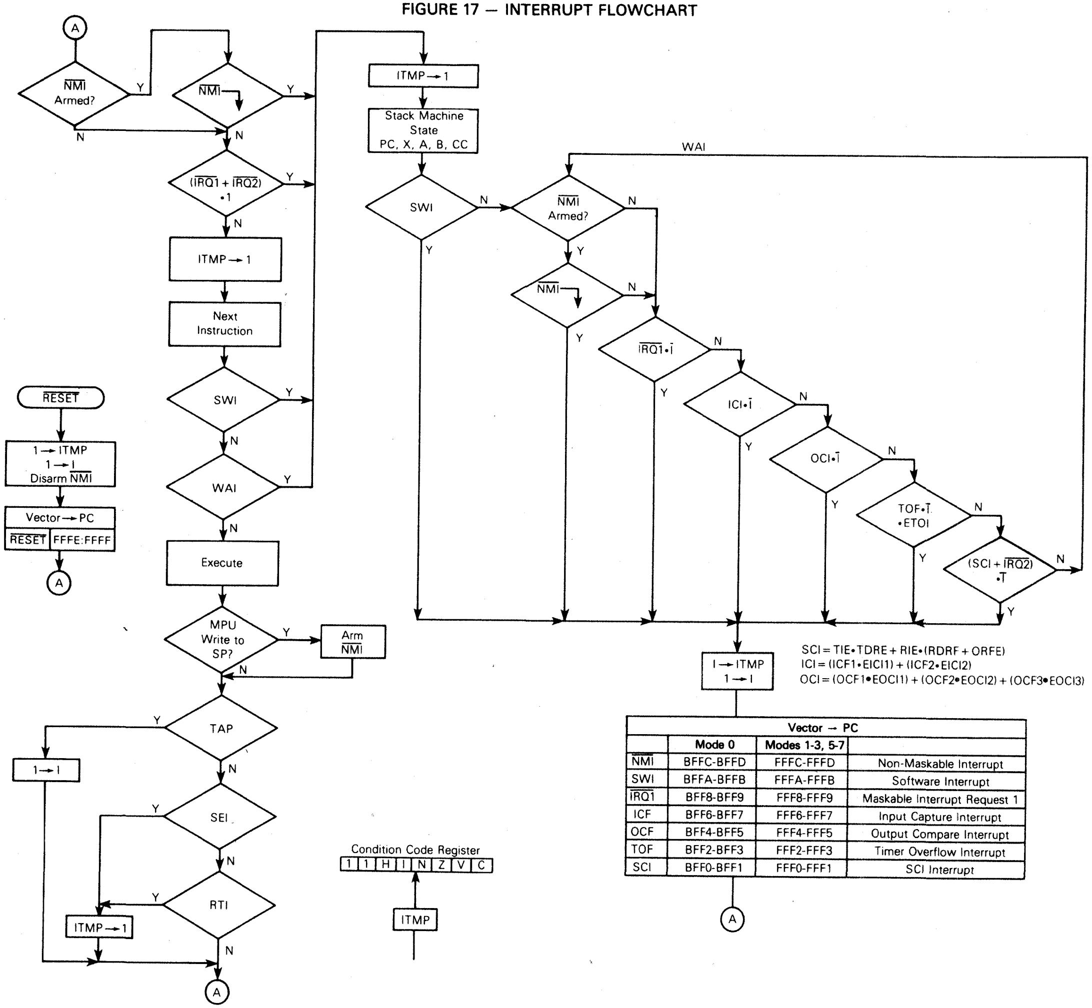
Mode 0		Modes 1-3, 5-7		Interrupt***
MSB	LSB	MSB	LSB	
BFFE	BFFF	FFFE	FFFF	$\overline{\text{RESET}}$
BFFC	BFFD	FFFC	FFFD	Non-Maskable Interrupt**
BFFA	BFFB	FFFA	FFFB	Software Interrupt
BFF8	BFF9	FFF8	FFF9	Maskable Interrupt Request 1
BFF6	BFF7	FFF6	FFF7	Input Capture Flag*
BFF4	BFF5	FFF4	FFF5	Output Compare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

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

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

***Mode 4 interrupt vectors are undefined.

FIGURE 17 — INTERRUPT FLOWCHART



SCI = TIE•TDRE + RIE•(RDRF + ORFE)
ICI = (ICF1•EICI1) + (ICF2•EICI2)
OCI = (OCF1•EOCI1) + (OCF2•EOCI2) + (OCF3•EOCI3)

	Vector → PC		
	Mode 0	**Modes 1-3, 5-7**	
$\overline{\text{NMI}}$	BFFC-BFFD	FFFC-FFFD	Non-Maskable Interrupt
SWI	BFFA-BFFB	FFFA-FFFB	Software Interrupt
$\overline{\text{IRQ1}}$	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

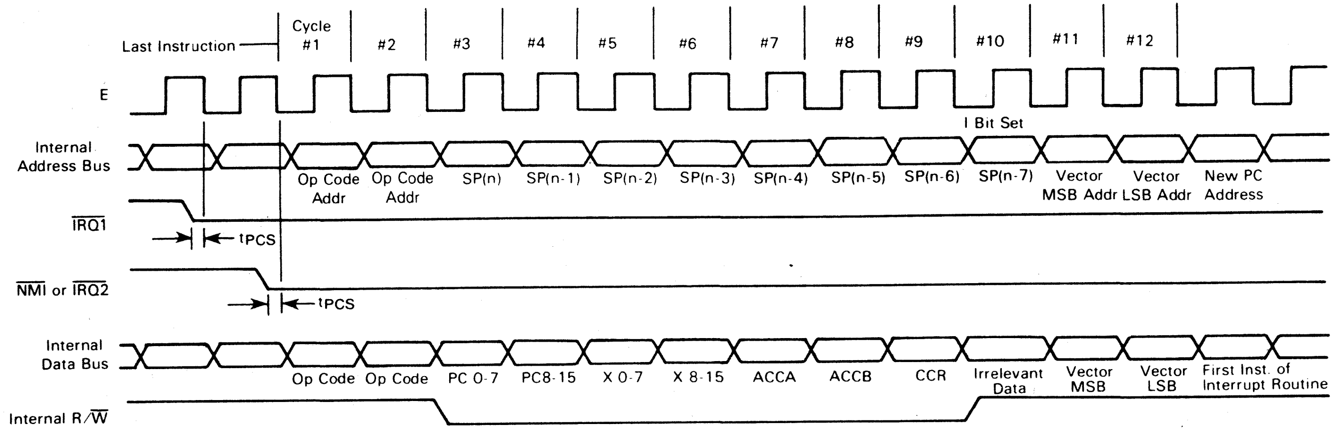
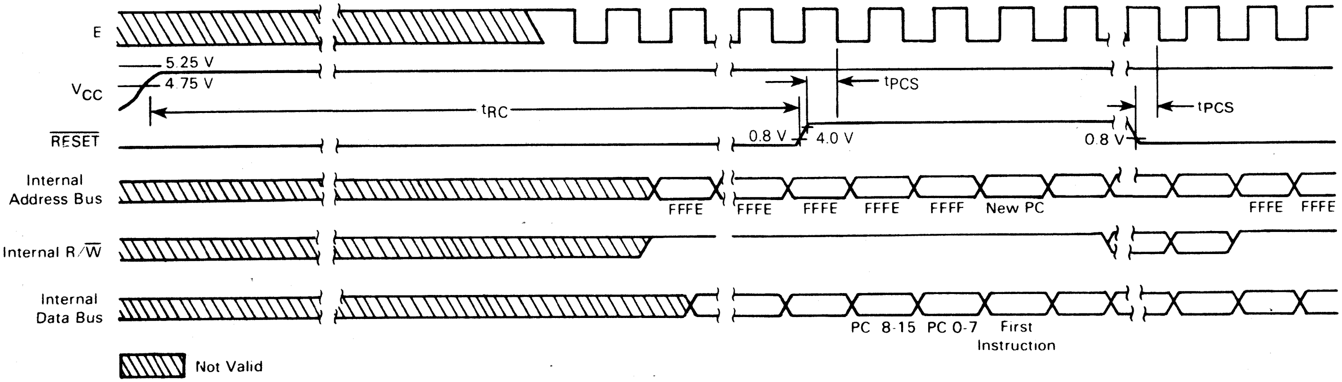


FIGURE 19 — $\overline{\text{RESET}}$ TIMING



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

$\overline{\text{RESET}}$

This input is used to reset the internal state of the device and provide an orderly startup procedure. During power up, $\overline{\text{RESET}}$ 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 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.

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

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 $FFFD ($BFFC and $BFFD in mode 0), 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.

NOTE

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

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

$\overline{\text{IRQ1}}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I bit) in the condition code register is clear, the 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.

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

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.

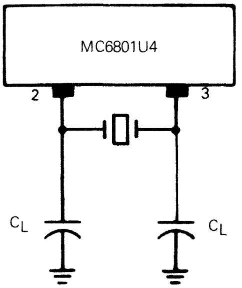
FIGURE 20 — MC6801U4/MC6803U4 FAMILY OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters

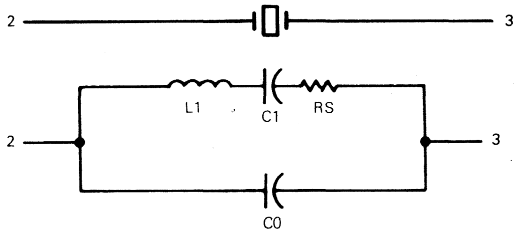
Nominal Crystal Parameters*

	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)

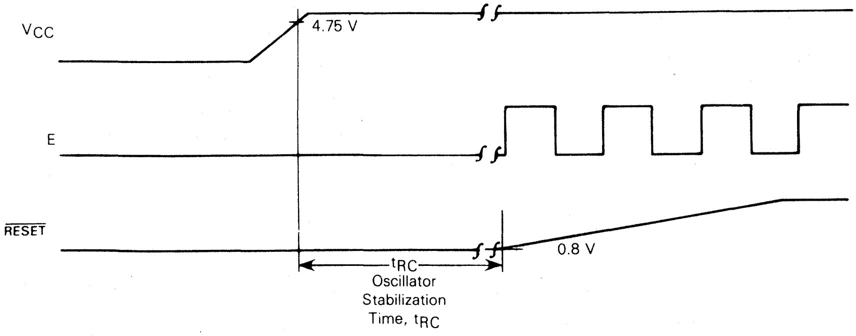


Equivalent Circuit

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 1.

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.

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 and not in mode 3, 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 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 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.

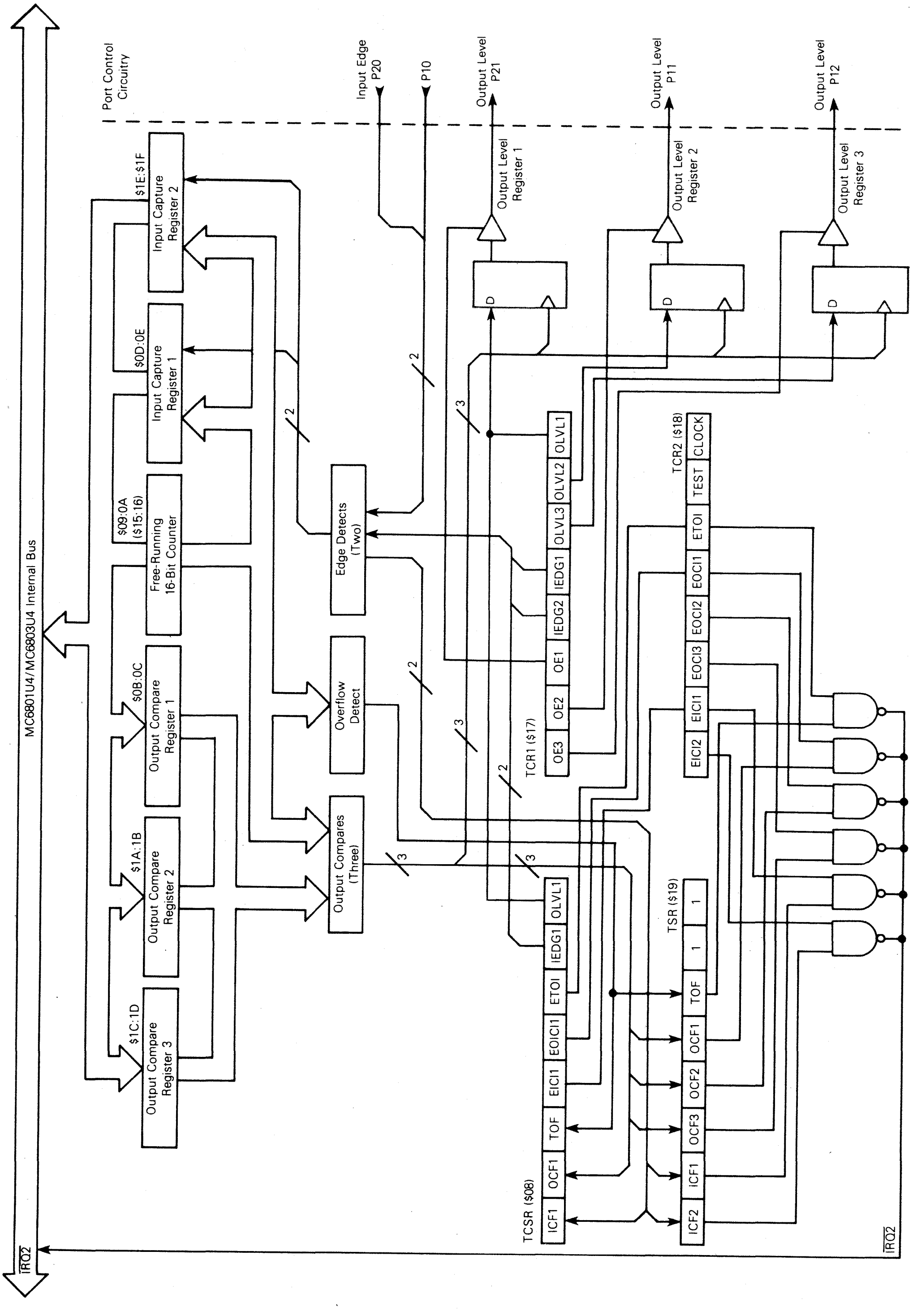
An input capture can occur independently of ICF; the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte MPU read. The input pulse width must be at least two E cycles to ensure an input capture under all conditions.

TIMER CONTROL AND STATUS REGISTERS

Four registers are used to provide the 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 $\overline{\text{IRQ2}}$ 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	OLVL	$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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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

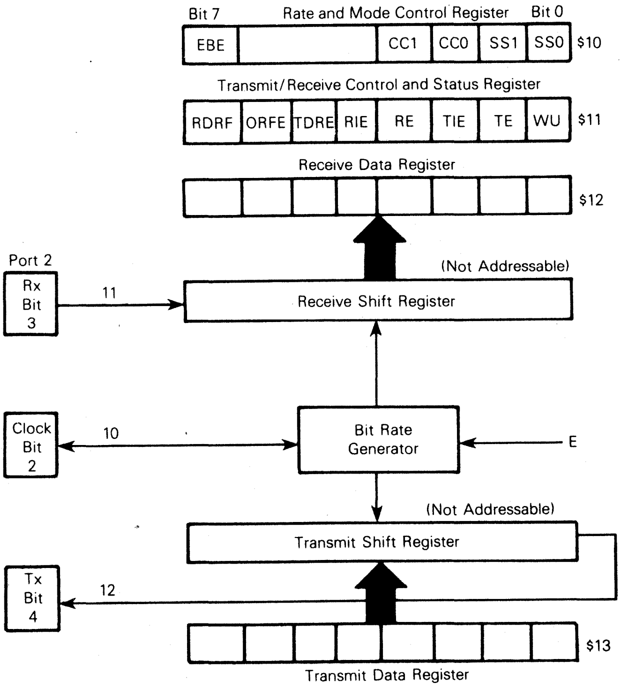
The following features of the SCI are programmable:

- 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		$4f_o \rightarrow$	2.4576 MHz		4.0 MHz		4.9152 MHz	
				614.4 kHz		1.0 MHz		1.2288 MHz	
			E	Baud	Time	Baud	Time	Baud	Time
0	0	0	÷16	38400.0	26 μs	62500.0	16.0 μs	76800.0	13.0 μs
0	0	1	÷128	4800.0	208.3 μs	7812.5	128.0 μs	9600.0	104.2 μs
0	1	0	÷1024	600.0	1.67 ms	976.6	1.024 ms	1200.0	833.3 μs
0	1	1	÷4096	150.0	6.67 ms	244.1	4.096 ms	300.0	3.33 ms
1	0	0	÷64	9600.0	104.2 μs	15625.0	64 μs	19200.0	52.0 μs
1	0	1	÷256	2400.0	416.6 μs	3906.3	256 μs	4800.0	208.3 μs
1	1	0	÷512	1200.0	833.3 μs	1953.1	512 μs	2400.0	416.6 μs
1	1	1	÷2048	300.0	3.33 ms	488.3	2.05 ms	600.0	01.67 ms
External (P22)*				76800.0	13.0 μs	125000.0	8.0 μs	153600.0	6.5 μs

*Using maximum clock rate

TABLE 7 — SCI FORMAT AND CLOCK SOURCE CONTROL

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

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) — The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{\text{RESET}}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

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

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

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

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

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

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

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

Bit 6 **Overrun Framing Error** — If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the 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.

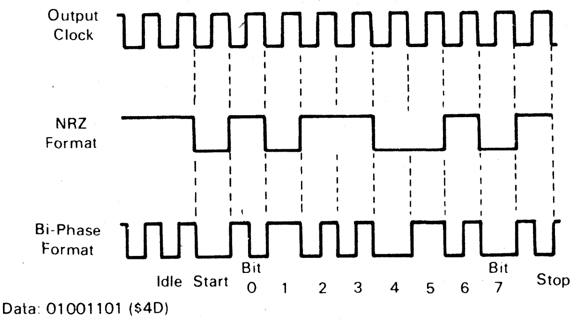
SERIAL OPERATIONS

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

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

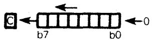
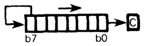
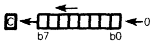
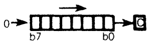
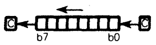
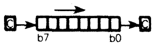
Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry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

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
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				 C b7 b0 0	●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3				 0 b7 b0 C	●	●	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				 C b7 b0 C	●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3				 C b7 b0 C	●	●	↕	↕	↕	↕
	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

		Direct			Relative			Index			Extnd			Inherent				Cond. Code Reg.					
																		5	4	3	2	1	0
Operations	Mnemonic	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	Branch Test	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										See Special Operations - Figure 24	●	●	●	●	●	●
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 24	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

	Inherent					Cond. Code Reg.					
						5	4	3	2	1	0
Operations	Mnemonic	OP	~	#	Boolean Operation	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
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
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E-CYCLES

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

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

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

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		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)

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

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

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

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

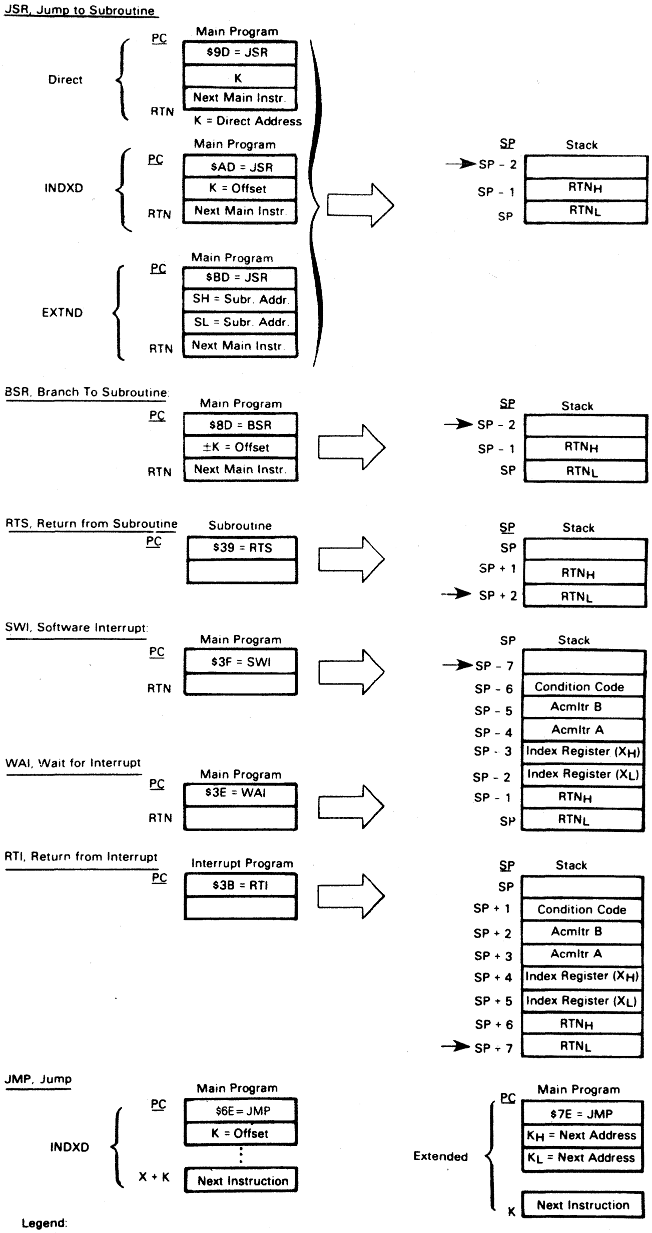
TABLE 14 — CYCLE-BY-CYCLE OPERATION (Sheet 5 of 6)

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

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

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

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.

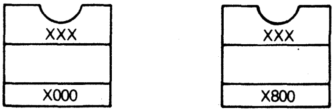


FIGURE A-1 XXX = Customer ID

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

MC6802
MC6808
MC6802NS

MICROPROCESSOR WITH CLOCK AND OPTIONAL RAM

The MC6802 is a monolithic 8-bit microprocessor that contains all the registers and accumulators of the present MC6800 plus an internal clock oscillator and driver on the same chip. In addition, the MC6802 has 128 bytes of on-board RAM located at hex addresses $0000 to $007F. The first 32 bytes of RAM, at hex addresses $0000 to $001F, may be retained in a low power mode by utilizing V_{CC} standby; thus, facilitating memory retention during a power-down situation.

The MC6802 is completely software compatible with the MC6800 as well as the entire M6800 family of parts. Hence, the MC6802 is expandable to 64K words.

The MC6802NS is identical to the MC6802 without standby RAM feature. The MC6808 is identical to the MC6802 without on-board RAM.

- On-Chip Clock Circuit
- 128 × 8 Bit On-Chip RAM
- 32 Bytes of RAM are Retainable
- Software-Compatible with the MC6800
- Expandable to 64K Words
- Standard TTL-Compatible Inputs and Outputs
- 8-Bit Word Size
- 16-Bit Memory Addressing
- Interrupt Capability

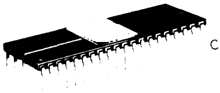
PART NUMBER DESIGNATION BY SPEED

MC6802 (1.0 MHz)	MC6808 (1.0 MHz)	MC6802NS (1.0 MHz)
MC68A02 (1.5 MHz)	MC68A08 (1.5 MHz)	
MC68B02 (2.0 MHz)	MC68B08 (2.0 MHz)	

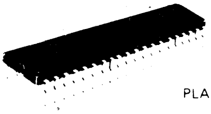
MOS

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

MICROPROCESSOR WITH CLOCK AND OPTIONAL RAM

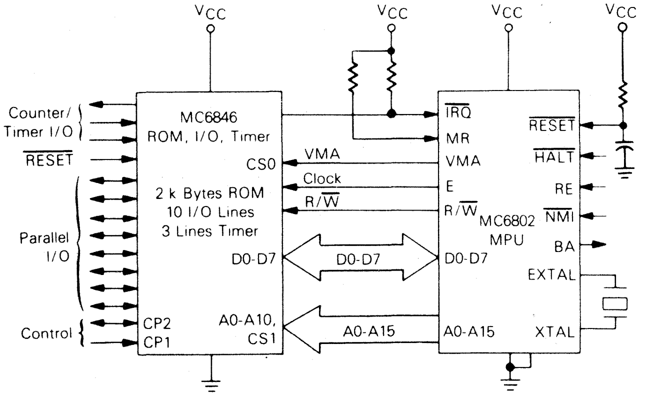


L SUFFIX
CERAMIC PACKAGE
CASE 715



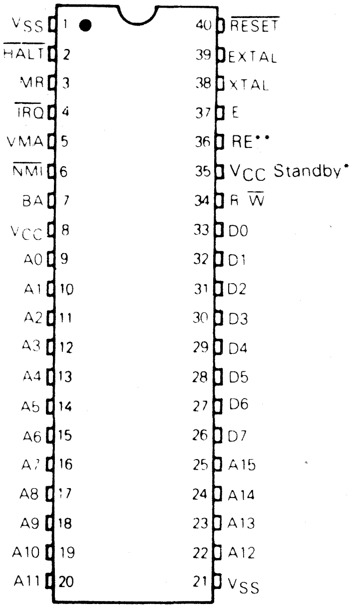
P SUFFIX
PLASTIC PACKAGE
CASE 711

TYPICAL MICROCOMPUTER



This block diagram shows a typical cost effective microcomputer. The MPU is the center of the microcopuier system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts in the M6800 Microcomputer family.

PIN ASSIGNMENT



*Pin 35 must be tied to 5 V on the 6802NS
**Pin 36 must be tied to ground for the 6808

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	Unit
Average Thermal Resistance (Junction to Ambient) Plastic Ceramic	θ_{JA}	 100 50	°C/W

This input 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. 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:

$T_A \equiv$ 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.

OPERATING TEMPERATURE RANGE

Device	Speed	Symbol	Value	Unit
MC6802P,L MC6802CP,CL	(1.0 MHz) (1.0 MHz)	T_A	0 to +70 −40 to +85	°C
MC68A02P,L MC68A02CP,CL	(1.5 MHz) (1.5 MHz)	T_A	0 to +70 −40 to +85	°C
MC68B02P,L MC68B02CP,CL	(2.0 MHz) (2.0 MHz)	T_A	0 to +70 −40 to +85	°C
MC6802NSP,L	(1.0 MHz)	T_A	0 to +70	°C
MC6808P,L MC68A08P,L MC68B08P,L	(1.0 MHz) (1.5 MHz) (2.0 MHz)	T_A	0 to +70	°C

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	Logic, EXTAL, $\overline{RESET}$	V_{IH}	$V_{SS}+2.0$ $V_{SS}+4.0$	– –	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, $\overline{RESET}$	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current ($V_{in}=0$ to 5.25 V, $V_{CC}=$ max)	Logic	I_{in}	–	1.0	2.5	μA
Output High Voltage ($I_{Load}=-205$ μA, $V_{CC}=$ min) ($I_{Load}=-145$ μA, $V_{CC}=$ min) ($I_{Load}=-100$ μA, $V_{CC}=$ min)	 D0-D7 A0-A15, $R/\overline{W}$, VMA, E BA	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$ $V_{SS}+2.4$	 – – –	 – – –	V
Output Low Voltage ($I_{Load}=1.6$ mA, $V_{CC}=$ min)		V_{OL}	–	–	$V_{SS}+0.4$	V
Internal Power Dissipation (Measured at $T_A=0$°C)		P_{INT}	–	0.600	1.0	W
V_{CC} Standby	Power Down Power Up	V_{SBB} V_{SB}	4.0 4.75	– –	5.25 5.25	V
Standby Current		I_{SBB}		–	8.0	mA
Capacitance # ($V_{in}=0$, $T_A=25$°C, $f=1.0$ MHz)	D0-D7 Logic Inputs, EXTAL	C_{in}		10 6.5	12.5 10	pF
	A0-A15, $R/\overline{W}$, VMA	C_{out}			12	pF

*In power-down mode, maximum power dissipation is less than 42 mW.

#Capacitances are periodically sampled rather than 100% tested.

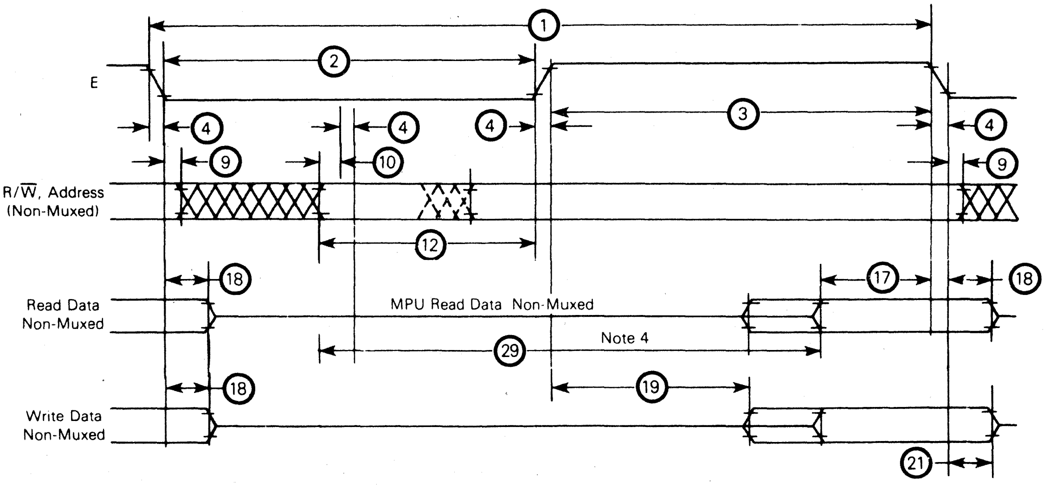
CONTROL TIMING ($V_{CC}=5.0$ V ±5%, $V_{SS}=0$, $T_A=T_L$ to T_H, unless otherwise noted)

Characteristics	Symbol	MC6802NS, MC6808		MC68A02 MC68A08		MC68B02 MC68B08		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	0.1	1.0	0.1	1.5	0.1	2.0	MHz
Crystal Frequency	f_{XTAL}	1.0	4.0	1.0	6.0	1.0	8.0	MHz
External Oscillator Frequency	$4xf_o$	0.4	4.0	0.4	6.0	0.4	8.0	MHz
Crystal Oscillator Start Up Time	t_{rc}	100	–	100	–	100	–	ms
Processor Controls (HALT, MR, RE, $\overline{RESET}$, $\overline{IRQ}$ $\overline{NMI}$)								
Processor Control Setup Time	t_{PCS}	200	–	140	–	110	–	ns
Processor Control Rise and Fall Time (Does Not Apply to $\overline{RESET}$)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

BUS TIMING CHARACTERISTICS

Ident. Number	Characteristic	Symbol	MC6802NS MC6802 MC6808		MC68A02 MC68A08		MC68B02 MC68B08		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	450	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9700	220	9700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	ns
12	Non-Muxed Address Valid Time to E (See Note 5)	t_{AV1} t_{AV2}	160 –	– 270	100 –	– –	50 –	– –	ns
17	Read Data Setup Time	t_{DSR}	100	–	70	–	60	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	ns
19	Write Data Delay Time	t_{DDW}	–	225	–	170	–	160	ns
21	Write Data Hold Time	t_{DHW}	30	–	20	–	20	–	ns
29	Usable Access Time (See Note 4)	t_{ACC}	605	–	310	–	235	–	ns

FIGURE 2 — BUS TIMING



NOTES:

1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.
3. All electricals shown for the MC6802 apply to the MC6802NS and MC6808, unless otherwise noted.
4. Usable access time is computed by: 12+3+4−17.
5. If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs cannot be executed from on-board RAM when using A and B parts (MC68A02, MC68A08, MC68B02, MC68B08). On-board RAM can be used for data storage with all parts.

FIGURE 3 — BUS TIMING TEST LOAD

C = 130 pF for D0-D7, E
= 90 pF for A0-A15, R/$\overline{W}$, and VMA
= 30 pF for BA
R = 11.7 kΩ for D0-D7, E
= 16.5 kΩ for A0-A15, R/$\overline{W}$, and VMA
= 24 kΩ for BA

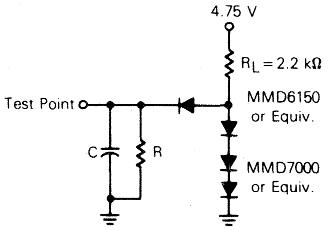


FIGURE 4 — TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

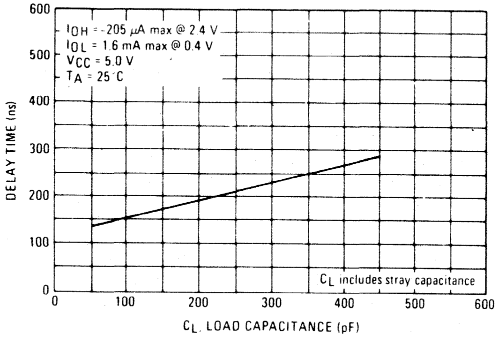


FIGURE 5 — TYPICAL READ/WRITE, VMA AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

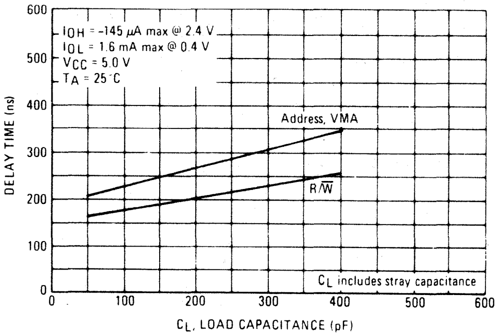
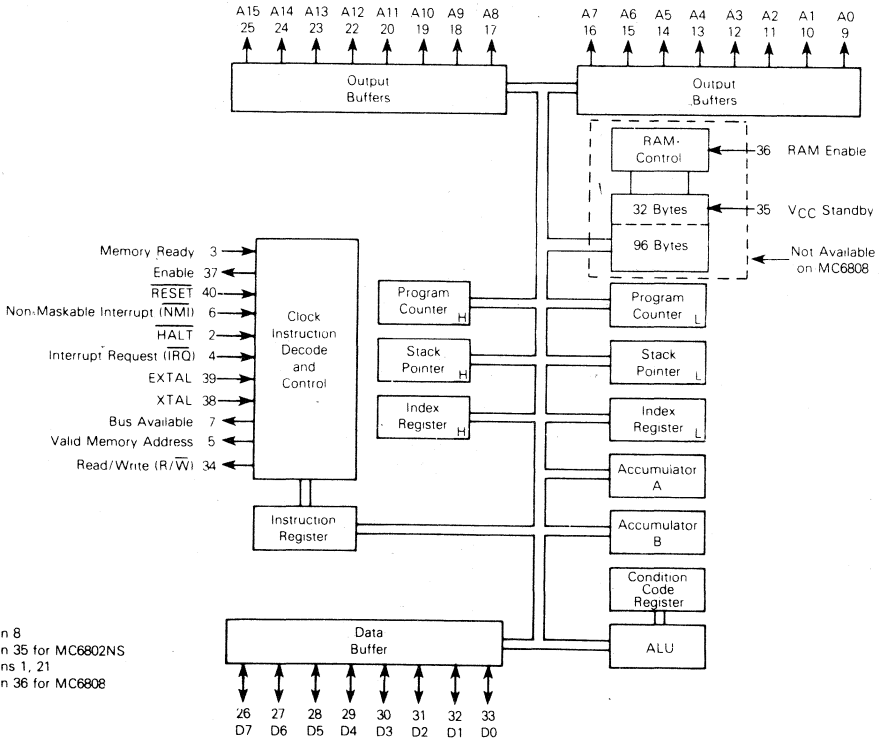


FIGURE 6 — EXPANDED BLOCK DIAGRAM



V_{CC} = Pin 8
V_{CC} = Pin 35 for MC6802NS
V_{SS} = Pins 1, 21
V_{SS} = Pin 36 for MC6808

MPU REGISTERS

A general block diagram of the MC6802 is shown in Figure 6. As shown, the number and configuration of the registers are the same as for the MC6800. The 128×8-bit RAM* has been added to the basic MPU. The first 32 bytes can be retained during power-up and power-down conditions via the RE signal.

The MC6802NS is identical to the MC6802 except for the standby feature on the first 32 bytes of RAM. The standby feature does not exist on the MC6802NS and thus pin 35 must be tied to 5 V.

The MC6808 is identical to the MC6802 except for on-board RAM. Since the MC6808 does not have on-board RAM pin 36 must be tied to ground allowing the processor to utilize up to 64K bytes of external memory.

The MPU has three 16-bit registers and three 8-bit registers available for use by the programmer (Figure 7).

PROGRAM COUNTER

The program counter is a two byte (16-bit) register that points to the current program address.

STACK POINTER

The stack pointer is a two byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access read/write memory that may have any location (address) that is convenient. In those applications that require storage of information in the stack when power is lost, the stack must be non-volatile.

INDEX REGISTER

The index register is a two byte register that is used to store data or a 16-bit memory address for the indexed mode of memory addressing.

ACCUMULATORS

The MPU contains two 8-bit accumulators that are used to hold operands and results from an arithmetic logic unit (ALU).

CONDITION CODE REGISTER

The condition code register indicates the results of an Arithmetic Logic Unit operation: Negative (N), Zero (Z), Overflow (V), Carry from bit 7 (C), and Half Carry from bit 3 (H). These bits of the Condition Code Register are used as testable conditions for the conditional branch instructions. Bit 4 is the interrupt mask bit (I). The unused bits of the Condition Code Register (b6 and b7) are ones.

Figure 8 shows the order of saving the microprocessor status within the stack.

---

*If programs are not executed from on-board RAM, TAV1 applies. If programs are to be stored and executed from on-board RAM, TAV2 applies. For normal data storage in the on-board RAM, this extended delay does not apply. Programs cannot be executed from on-board RAM when using A and B parts (MC68A02, MC68A08, MC68B02, and MC68B08). On-board RAM can be used for data storage with all parts.

FIGURE 7 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT

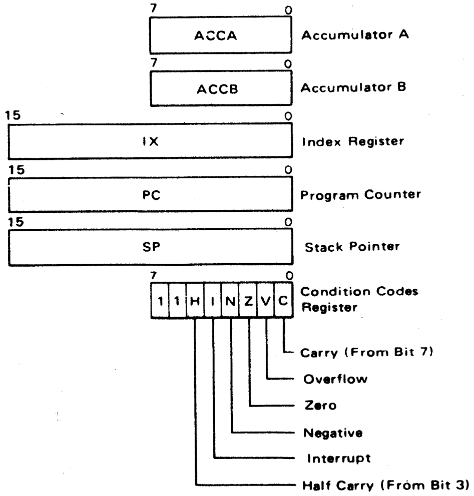
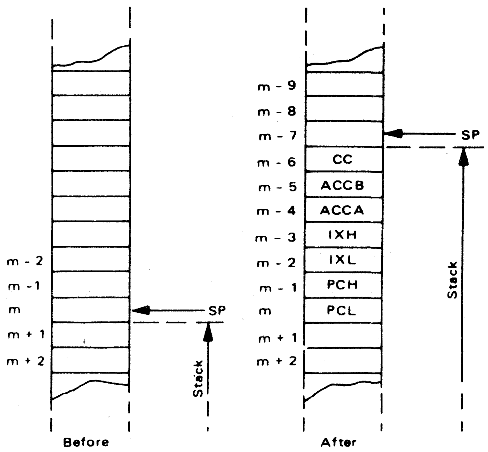


FIGURE 8 — SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK

SP = Stack Pointer
CC = Condition Codes (Also called the Processor Status Byte)
ACCB = Accumulator B
ACCA = Accumulator A
IXH = Index Register, Higher Order 8 Bits
IXL = Index Register, Lower Order 8 Bits
PCH = Program Counter, Higher Order 8 Bits
PCL = Program Counter, Lower Order 8 Bits



MPU SIGNAL DESCRIPTION

Proper operation of the MPU requires that certain control and timing signals be provided to accomplish specific functions and that other signal lines be monitored to determine the state of the processor. These control and timing signals are similar to those of the MC6800 except that TSC, DBE, $\phi 1$, $\phi 2$ input, and two unused pins have been eliminated, and the following signal and timing lines have been added:

RAM Enable (RE)
Crystal Connections EXTAL and XTAL
Memory Ready (MR)
V_{CC} Standby
Enable $\phi 2$ Output (E)

The following is a summary of the MPU signals:

ADDRESS BUS (A0-A15)

Sixteen pins are used for the address bus. The outputs are capable of driving one standard TTL load and 90 pF. These lines do not have three-state capability.

DATA BUS (D0-D7)

Eight pins are used for the data bus. It is bidirectional, transferring data to and from the memory and peripheral devices. It also has three-state output buffers capable of driving one standard TTL load and 130 pF.

Data bus will be in the output mode when the internal RAM is accessed and RE will be high. This prohibits external data entering the MPU. It should be noted that the internal RAM is fully decoded from $0000 to $007F. External RAM at $0000 to $007F must be disabled when internal RAM is accessed.

$\overline{\text{HALT}}$

When this input is in the low state, all activity in the machine will be halted. This input is level sensitive. In the $\overline{\text{HALT}}$ mode, the machine will stop at the end of an instruction, bus available will be at a high state, valid memory address will be at a low state. The address bus will display the address of the next instruction.

To ensure single instruction operation, transition of the $\overline{\text{HALT}}$ line must occur t_{PCS} before the falling edge of E and the $\overline{\text{HALT}}$ line must go high for one clock cycle.

$\overline{\text{HALT}}$ should be tied high if not used. This is good engineering design practice in general and necessary to ensure proper operation of the part.

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

This TTL-compatible output signals the peripherals and memory devices whether the MPU is in a read (high) or write (low) state. The normal standby state of this signal is read (high). When the processor is halted, it will be in the read state. This output is capable of driving one standard TTL load and 90 pF.

VALID MEMORY ADDRESS (VMA)

This output indicates to peripheral devices that there is a valid address on the address bus. In normal operation, this signal should be utilized for enabling peripheral interfaces such as the PIA and ACIA. This signal is not three-state. One standard TTL load and 90 pF may be directly driven by this active high signal.

BUS AVAILABLE (BA) — The bus available signal will normally be in the low state; when activated, it will go to the high state indicating that the microprocessor has stopped and that the address bus is available (but not in a three-state condition). This will occur if the $\overline{\text{HALT}}$ line is in the low state or the processor is in the WAIT state as a result of the execution of a WAIT instruction. At such time, all three-state output drivers will go to their off-state and other outputs to their normally inactive level. The processor is removed from the

WAIT state by the occurrence of a maskable (mask bit I = 0) or nonmaskable interrupt. This output is capable of driving one standard TTL load and 30 pF.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$)

A low level on this input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being excuted before it recognizes the request. At that time, if the interrupt mask bit in the condition code register is not set, the machine will begin an interrupt sequence. The index register, program counter, accumulators, and condition code register are stored away on the stack. Next the MPU will respond to the interrupt request by setting the interrupt mask bit high so that no further interrupts may occur. At the end of the cycle, a 16-bit vectoring address which is located in memory locations \$FFF8 and \$FFF9 is loaded which causes the MPU to branch to an interrupt routine in memory.

The $\overline{\text{HALT}}$ line must be in the high state for interrupts to be serviced. Interrupts will be latched internally while $\overline{\text{HALT}}$ is low.

A nominal 3 kΩ pullup resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. $\overline{\text{IRQ}}$ may be tied directly to V_{CC} if not used.

$\overline{\text{RESET}}$

This input is used to reset and start the MPU from a power-down condition, resulting from a power failure or an initial start-up of the processor. When this line is low, the MPU is inactive and the information in the registers will be lost. If a high level is detected on the input, this will signal the MPU to begin the restart sequence. This will start execution of a routine to initialize the processor from its reset condition. All the higher order address lines will be forced high. For the restart, the last two (\$FFFE, \$FFFF) locations in memory will be used to load the program that is addressed by the program counter. During the restart routine, the interrupt mask bit is set and must be reset before the MPU can be interrupted by $\overline{\text{IRQ}}$. Power-up and reset timing and power-down sequences are shown in Figures 9 and 10, respectively.

$\overline{\text{RESET}}$, when brought low, must be held low at least three clock cycles. This allows adequate time to respond internally to the reset. This is independent of the t_{rc} power-up reset that is required.

When $\overline{\text{RESET}}$ is released it *must* go through the low-to-high threshold without bouncing, oscillating, or otherwise causing an erroneous reset (less than three clock cycles). This may cause improper MPU operation until the next valid reset.

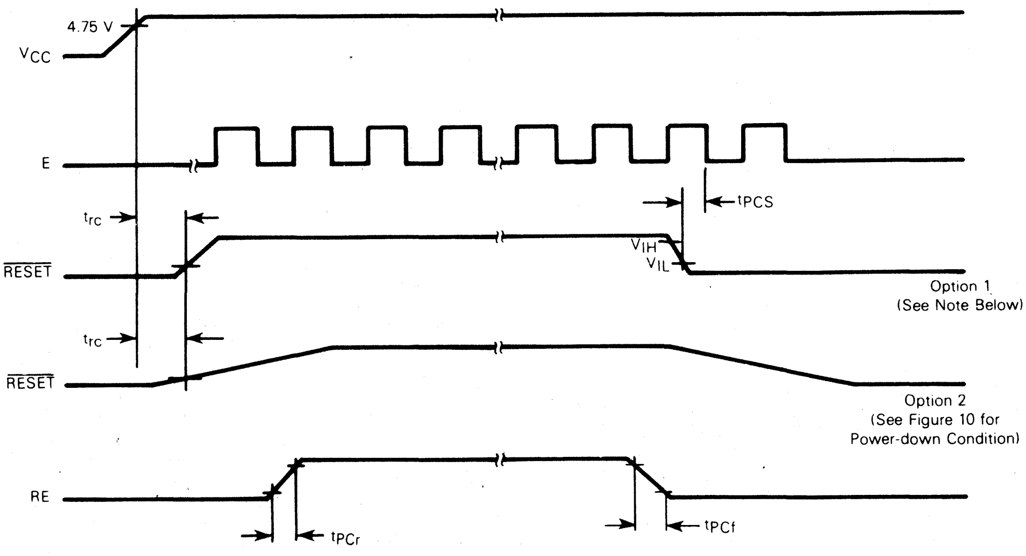
NON-MASKABLE INTERRUPT ($\overline{\text{NMI}}$)

A low-going edge on this input requests that a non-maskable interrupt sequence be generated within the processor. As with the interrupt request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{\text{NMI}}$ signal. The interrupt mask bit in the condition code register has no effect on $\overline{\text{NMI}}$.

The index register, program counter, accumulators, and condition code registers are stored away on the stack. At the end of the cycle, a 16-bit vectoring address which is located in memory locations \$FFFC and \$FFFD is loaded causing the MPU to branch to an interrupt service routine in memory.

A nominal 3 kΩ pullup resistor to V_{CC} should be used for wire-OR and optimum control of interrupts. $\overline{\text{NMI}}$ may be tied

FIGURE 9 — POWER-UP AND RESET TIMING



NOTE: If option 1 is chosen, $\overline{\text{RESET}}$ and RE pins can be tied together.

directly to V_{CC} if not used.

Inputs $\overline{IRQ}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled when E is high and will start the interrupt routine on a low E following the completion of an instruction.

Figure 11 is a flowchart describing the major decision paths and interrupt vectors of the microprocessor. Table 1 gives the memory map for interrupt vectors.

TABLE 1 — MEMORY MAP FOR INTERRUPT VECTORS

Vector		Description
MS	LS	
$FFFE	$FFFF	Restart
$FFFC	$FFFD	Non-Maskable Interrupt
$FFFA	$FFFB	Software Interrupt
$FFF8	$FFF9	Interrupt Request

FIGURE 10 — POWER-DOWN SEQUENCE

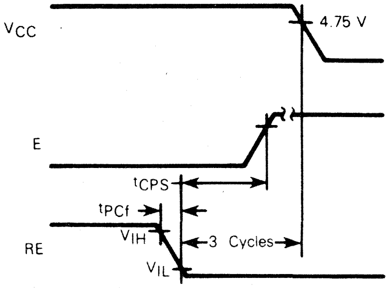


FIGURE 11 — MPU FLOWCHART

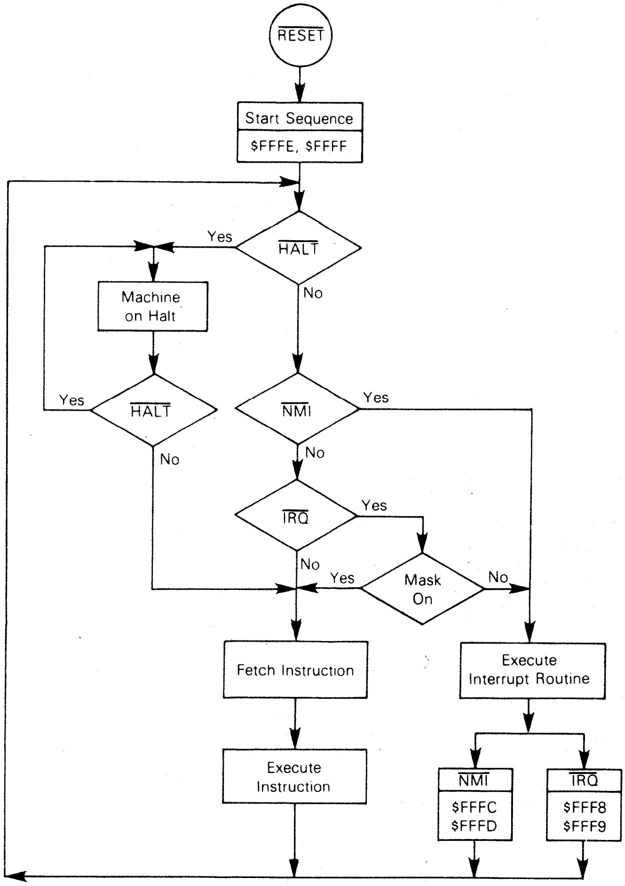
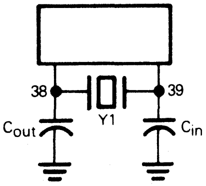
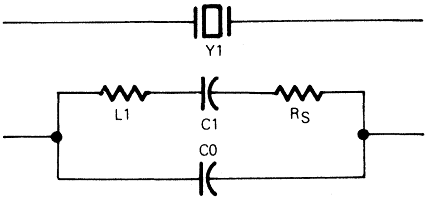


FIGURE 12 — CRYSTAL SPECIFICATIONS



Y1	C_{in}	C_{out}
3.58 MHz	27 pF	27 pF
4 MHz	27 pF	27 pF
6 MHz	20 pF	20 pF
8 MHz	18 pF	18 pF

Crystal Loading



Nominal Crystal Parameters*

	3.58 MHz	4.0 MHz	6.0 MHz	8.0 MHz
R_S	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C0	3.5 pF	6.5 pF	4-6 pF	4-6 pF
C1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40K	>30K	>20K	>20K

*These are representative AT-cut parallel resonance crystal parameters only. Crystals of other types of cuts may also be used.

Figure 13 — SUGGESTED PC BOARD LAYOUT

Example of Board Design Using the Crystal Oscillator

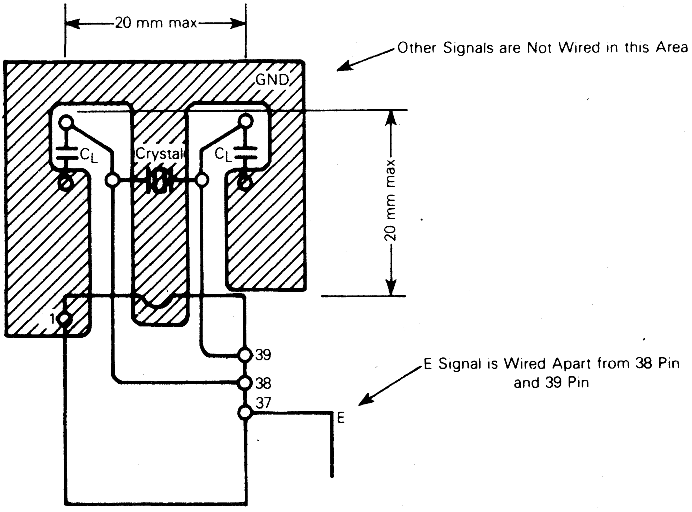


FIGURE 14 — MEMORY READY SYNCHRONIZATION

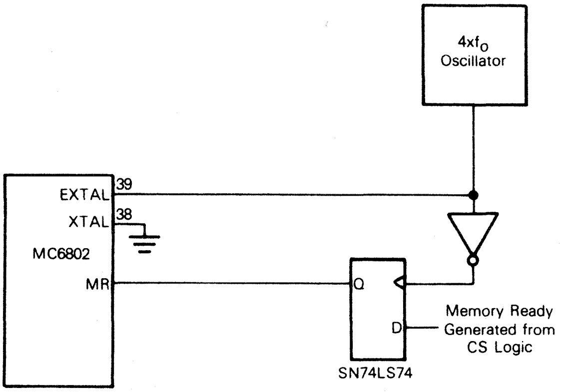
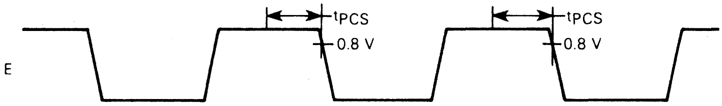


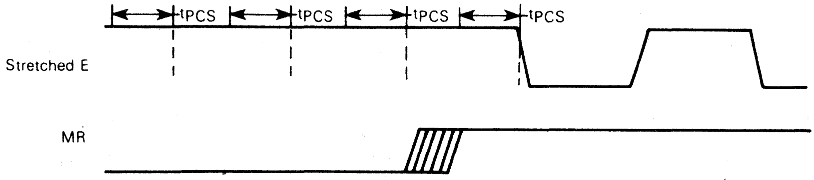
FIGURE 15 — MR NEGATIVE SETUP TIME REQUIREMENT

E Clock Stretch



The E clock will be stretched at end of E high of the cycle during which MR negative meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to the fall of E. If the t_{PCS} setup time is not met, E will be stretched at the end of the next E-high ½ cycle. E will be stretched in integral multiples of ½ cycles.

Resuming E Clocking



The E clock will resume normal operation at the end of the ½ cycle during which MR assertion meets the t_{PCS} setup time. The t_{PCS} setup time is referenced to transitions of E were it not stretched. If t_{PCS} setup time is not met, E will fall at the second possible transition time after MR is asserted. There is no direct means of determining when the t_{PCS} references occur, unless the synchronizing circuit of Figure 14 is used.

RAM ENABLE (RE — MC6802 + MC6802NS ONLY)

A TTL-compatible RAM enable input controls the on-chip RAM of the MC6802. When placed in the high state, the on-chip memory is enabled to respond to the MPU controls. In the low state, RAM is disabled. This pin may also be utilized to disable reading and writing the on-chip RAM during a power-down situation. RAM Enable must be low three cycles before V_{CC} goes below 4.75 V during power-down. RAM enable must be tied low on the MC6808. RE should be tied to the correct high or low state if not used.

EXTAL AND XTAL

These inputs are used for the internal oscillator that may be crystal controlled. These connections are for a parallel resonant fundamental crystal (see Figure 12). (AT-cut.) A divide-by-four circuit has been added so a 4 MHz crystal may be used in lieu of a 1 MHz crystal for a more cost-effective system. An example of the crystal circuit layout is shown in Figure 13. Pin 39 may be driven externally by a TTL input signal four times the required E clock frequency. Pin 38 is to be grounded.

An RC network is not directly usable as a frequency source on pins 38 and 39. An RC network type TTL or CMOS oscillator will work well as long as the TTL or CMOS output drives the on-chip oscillator.

LC networks are not recommended to be used in place of the crystal.

If an external clock is used, it may not be halted for more than $t_{PW\phi L}$. The MC6802, MC6808 and MC6802NS are dynamic parts except for the internal RAM, and require the external clock to retain information.

MEMORY READY (MR)

MR is a TTL-compatible input signal controlling the stretching of E. Use of MR requires synchronization with the $4xf_o$ signal, as shown in Figure 14. When MR is high, E will be in normal operation. When MR is low, E will be stretched integral numbers of half periods, thus allowing interface to slow memories. Memory Ready timing is shown in Figure 15.

MR should be tied high (connected directly to V_{CC}) if not used. This is necessary to ensure proper operation of the part. A maximum stretch is t_{cyc}.

ENABLE (E)

This pin supplies the clock for the MPU and the rest of the system. This is a single-phase, TTL-compatible clock. This clock may be conditioned by a memory read signal. This is equivalent to $\phi 2$ on the MC6800. This output is capable of driving one standard TTL load and 130 pF.

V_{CC} STANDBY (MC6802 ONLY)

This pin supplies the dc voltage to the first 32 bytes of RAM as well as the RAM Enable (RE) control logic. Thus, retention of data in this portion of the RAM on a power-up, power-down, or standby condition is guaranteed. Maximum current drain at V_{SB} maximum is I_{SBB}. For the MC6802NS this pin must be connected to V_{CC}.

MPU INSTRUCTION SET

The instruction set has 72 different instructions. Included are binary and decimal arithmetic, logical, shift, rotate, load, store, conditional or unconditional branch, interrupt and stack manipulation instructions (Tables 2 through 6). The instruction set is the same as that for the MC6800.

MPU ADDRESSING MODES

There are seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 7 along with the associated instruction execution time that is given in machine cycles. With a bus frequency of 1 MHz, these times would be microseconds.

ACCUMULATOR (ACCX) ADDRESSING

In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

IMMEDIATE ADDRESSING

In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The MPU addresses this location when it fetches the immediate instruction for execution. These are two- or three-byte instructions.

DIRECT ADDRESSING

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine, i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random-access memory. These are two-byte instructions.

EXTENDED ADDRESSING

In extended addressing, the address contained in the second byte of the instruction is used as the higher eight bits of the address of the operand. The third byte of the instruction is used as the lower eight bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

INDEXED ADDRESSING

In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest eight bits in the MPU. The carry is then added to the higher order eight bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

IMPLIED ADDRESSING

In the implied addressing mode, the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

RELATIVE ADDRESSING

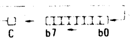
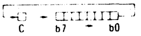
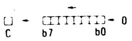
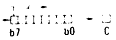
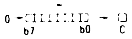
In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest eight bits plus two. The carry or borrow is then added to the high eight bits. This allows the user to address data within a range of − 125 to + 129 bytes of the present instruction. These are two-byte instructions.

TABLE 2 — MICROPROCESSOR INSTRUCTION SET — ALPHABETIC SEQUENCE

ABA	Add Accumulators
ADC	Add with Carry
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero
BGE	Branch if Greater or Equal Zero
BGT	Branch if Greater than Zero
BHI	Branch if Higher
BIT	Bit Test
BLE	Branch if Less or Equal
BLS	Branch if Lower or Same
BLT	Branch if Less than Zero
BMI	Branch if Minus
BNE	Branch if Not Equal to Zero
BPL	Branch if Plus
BRA	Branch Always
BSR	Branch to Subroutine
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CBA	Compare Accumulators
CLC	Clear Carry
CLI	Clear Interrupt Mask
CLR	Clear
CLV	Clear Overflow
CMP	Compare
COM	Complement
CPX	Compare Index Register
DAA	Decimal Adjust
DEC	Decrement
DES	Decrement Stack Pointer
DEX	Decrement Index Register
EOR	Exclusive OR
INC	Increment
INS	Increment Stack Pointer
INX	Increment Index Register
JMP	Jump
JSR	Jump to Subroutine
LDA	Load Accumulator
LDS	Load Stack Pointer
LDX	Load Index Register
LSR	Logical Shift Right
NEG	Negate
NOP	No Operation
ORA	Inclusive OR Accumulator
PSH	Push Data
PUL	Pull Data
ROL	Rotate Left
ROR	Rotate Right
RTI	Return from Interrupt
RTS	Return from Subroutine
SBA	Subtract Accumulators
SBC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Register
STX	Store Index Register
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumulators
TAP	Transfer Accumulators to Condition Code Reg.
TBA	Transfer Accumulators
TPA	Transfer Condition Code Reg. to Accumulator
TST	Test
TSX	Transfer Stack Pointer to Index Register
TXS	Transfer Index Register to Stack Pointer
WAI	Wait for Interrupt

TABLE 3 — ACCUMULATOR AND MEMORY INSTRUCTIONS

ADDRESSING MODES | BOOLEAN/ARITHMETIC OPERATION | COND. CODE REG.

		IMMED			DIRECT			INDEX			EXTND			IMPLIED			(All register labels refer to contents)	5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	=	OP	~	=	OP	~	=	OP	~	=	OP	~	=		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	●	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	●	●	↕	↕	R	●
Clear	CLR							6F	7	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	5	2	B1	4	3				A − M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	●	●	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	●	●	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				M̄ → M	●	●	↕	↕	R	S
	COMA													43	2	1	Ā → A	●	●	↕	↕	R	S
	COMB													53	2	1	B → B	●	●	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	●	●	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	●	●	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	●	●	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	●	●	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	●	●	↕	↕	④	●
	DECA													4A	2	1	A − 1 → A	●	●	↕	↕	④	●
	DECB													5A	2	1	B − 1 → B	●	●	↕	↕	④	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	●	●	↕	↕	R	●
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	●	●	↕	↕	⑤	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	⑤	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	⑤	●
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	●	●	↕	↕	R	●
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	●	●	●	●	●	●
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	●	●	●	●	●	●
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	●	●	●	●	●	●
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	●	●	●	●	●	●
Rotate Left	ROL							69	7	2	79	6	3				M 	●	●	↕	↕	⑥	↕
	ROLA													49	2	1	A	●	●	↕	↕	⑥	↕
	ROLB													59	2	1	B	●	●	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M 	●	●	↕	↕	⑥	↕
	RORA													46	2	1	A	●	●	↕	↕	⑥	↕
	RORB													56	2	1	B	●	●	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M 	●	●	↕	↕	⑥	↕
	ASLA													48	2	1	A	●	●	↕	↕	⑥	↕
	ASLB													58	2	1	B	●	●	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M 	●	●	↕	↕	⑥	↕
	ASRA													47	2	1	A	●	●	↕	↕	⑥	↕
	ASRB													57	2	1	B	●	●	↕	↕	⑥	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M 	●	●	R	↕	⑥	↕
	LSRA													44	2	1	A	●	●	R	↕	⑥	↕
	LSRB													54	2	1	B	●	●	R	↕	⑥	↕
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	●	●	↕	↕	R	●
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	●	●	↕	↕	↕	↕
Subtract Acmltrs	SBA													10	2	1	A − B → A	●	●	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	●	●	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A − 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B − 00	●	●	↕	↕	R	R
																		H	I	N	Z	V	C

LEGEND

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

Note: Accumulator addressing mode instructions are included in the column for IMPLIED addressing

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 from bit 7
R	Reset Always
S	Set Always
↕	Test and set if true, cleared otherwise
●	Not Affected

TABLE 4 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

		IMMED			DIRECT			INDEX			EXTND			IMPLIED				COND. CODE REG. 5	4	3	2	1	0
POINTER OPERATIONS	**MNEMONIC**	**OP**	**~**	**=**	**OP**	**~**	**=**	**OP**	**~**	**=**	**OP**	**~**	**=**	**OP**	**~**	**=**	**BOOLEAN/ARITHMETIC OPERATION**	**H**	**I**	**N**	**Z**	**V**	**C**
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				X_H M, X_L - (M + 1)	●	●	⑦	↕	⑧	●
Decrement Index Reg	DEX													09	4	1	X 1 · X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	4	1	SP 1 · SP	●	●	●	●	●	●
Increment Index Reg	INX													08	4	1	X + 1 · X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	4	1	SP + 1 · SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				M · X_H, (M + 1) · X_L	●	●	⑨	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				M · SP_H, (M + 1) · SP_L	●	●	⑨	↕	R	●
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				X_H · M, X_L · (M + 1)	●	●	⑨	↕	R	●
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				SP_H · M, SP_L · (M + 1)	●	●	⑨	↕	R	●
Indx Reg · Stack Pntr	TXS													35	4	1	X 1 · SP	●	●	●	●	●	●
Stack Pntr · Indx Reg	TSX													30	4	1	SP + 1 · X	●	●	●	●	●	●

TABLE 5 — JUMP AND BRANCH INSTRUCTIONS

		RELATIVE			INDEX			EXTND			IMPLIED				COND. CODE REG. 5	4	3	2	1	0
OPERATIONS	**MNEMONIC**	**OP**	**~**	**=**	**OP**	**~**	**=**	**OP**	**~**	**=**	**OP**	**~**	**=**	**BRANCH TEST**	**H**	**I**	**N**	**Z**	**V**	**C**
Branch Always	BRA	20	4	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC	24	4	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS	25	4	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ	27	4	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE	2C	4	2										N ⊕ V = 0	●	●	●	●	●	●
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	●	●	●	●	●	●
Branch If Higher	BHI	22	4	2										C + Z = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	●	●	●	●	●	●
Branch If Minus	BMI	2B	4	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC	28	4	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS	29	4	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL	2A	4	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR	8D	8	2										See Special Operations (Figure 16)	●	●	●	●	●	●
Jump	JMP				6E	4	2	7E	3	3					●	●	●	●	●	●
Jump To Subroutine	JSR				AD	8	2	BD	9	3					●	●	●	●	●	●
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	●	●	●	●	●	●
Return From Interrupt	RTI										3B	10	1		⑩					
Return From Subroutine	RTS										39	5	1	See Special Operations (Figure 16)	●	●	●	●	●	●
Software Interrupt	SWI										3F	12	1		●	●	●	●	●	●
Wait for Interrupt	WAI										3E	9	1		●	⑪	●	●	●	●

FIGURE 16 — SPECIAL OPERATIONS

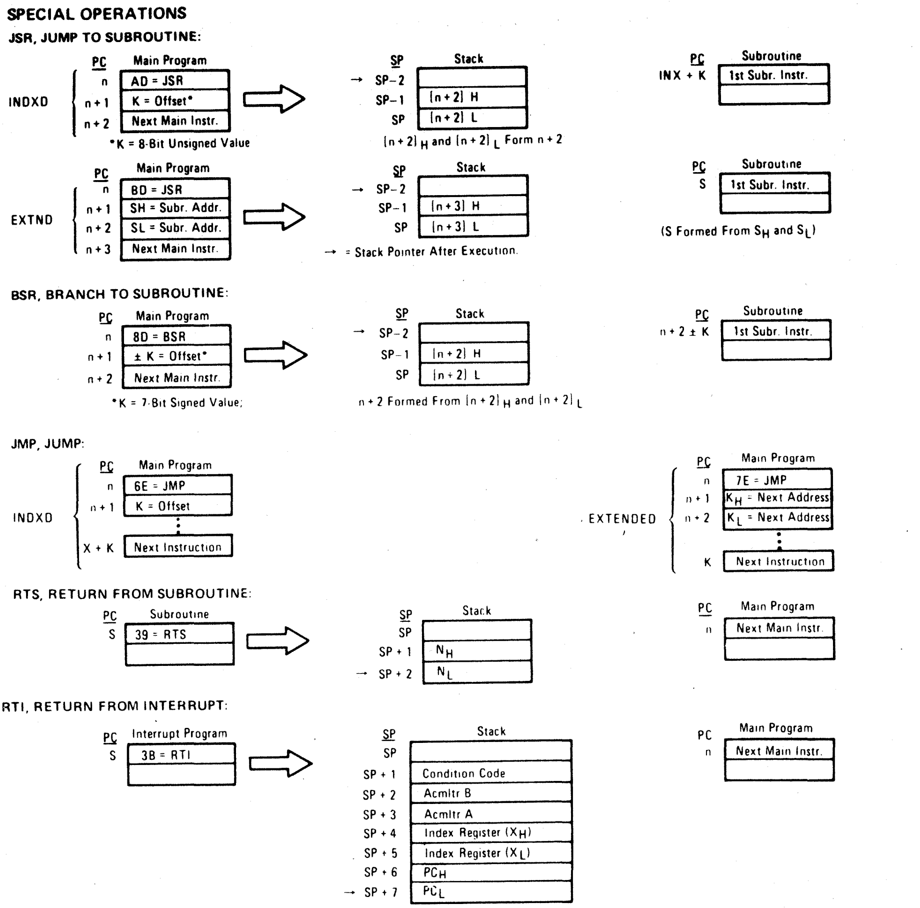


TABLE 6 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

		IMPLIED				COND. CODE REG.					
						5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	=	BOOLEAN OPERATION	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	●
Acmltr A → CCR	TAP	06	2	1	A → CCR	⑫					
CCR → Acmltr A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

CONDITION CODE REGISTER NOTES: (Bit set if test is true and cleared otherwise)

1	(Bit V)	Test: Result = 10000000?
2	(Bit C)	Test: Result ≠ 00000000?
3	(Bit C)	Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)
4	(Bit V)	Test: Operand = 10000000 prior to execution?
5	(Bit V)	Test: Operand = 01111111 prior to execution?
6	(Bit V)	Test: Set equal to result of N⊕C after shift has occurred.
7	(Bit N)	Test: Sign bit of most significant (MS) byte = 1?
8	(Bit V)	Test: 2's complement overflow from subtraction of MS bytes?
9	(Bit N)	Test: Result less than zero? (Bit 15 = 1)
10	(All)	Load Condition Code Register from Stack. (See Special Operations)
11	(Bit I)	Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
12	(All)	Set according to the contents of Accumulator A.

TABLE 7 — INSTRUCTION ADDRESSING MODES AND ASSOCIATED EXECUTION TIMES
(Times in Machine Cycle)

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA		•	•	•	•	•	2	•
ADC	x	•	2	3	4	5	•	•
ADD	x	•	2	3	4	5	•	•
AND	x	•	2	3	4	5	•	•
ASL		2	•	•	6	7	•	•
ASR		2	•	•	6	7	•	•
BCC		•	•	•	•	•	•	4
BCS		•	•	•	•	•	•	4
BEA		•	•	•	•	•	•	4
BGE		•	•	•	•	•	•	4
BGT		•	•	•	•	•	•	4
BHI		•	•	•	•	•	•	4
BIT	x	•	2	3	4	5	•	•
BLE		•	•	•	•	•	•	4
BLS		•	•	•	•	•	•	4
BLT		•	•	•	•	•	•	4
BMI		•	•	•	•	•	•	4
BNE		•	•	•	•	•	•	4
BPL		•	•	•	•	•	•	4
BRA		•	•	•	•	•	•	4
BSR		•	•	•	•	•	•	8
BVC		•	•	•	•	•	•	4
BVS		•	•	•	•	•	•	4
CBA		•	•	•	•	•	2	•
CLC		•	•	•	•	•	2	•
CLI		•	•	•	•	•	2	•
CLR		2	•	•	6	7	•	•
CLV		•	•	•	•	•	2	•
CMP	x	•	2	3	4	5	•	•
COM		2	•	•	6	7	•	•
CPX		•	3	4	5	6	•	•
DAA		•	•	•	•	•	2	•
DEC		2	•	•	6	7	•	•
DES		•	•	•	•	•	4	•
DEX		•	•	•	•	•	4	•
EOR	x	•	2	3	4	5	•	•

	(Dual Operand)	ACCX	Immediate	Direct	Extended	Indexed	Implied
INC		2	•	•	6	7	•
INS		•	•	•	•	•	4
INX		•	•	•	•	•	4
JMP		•	•	•	3	4	•
JSR		•	•	•	9	8	•
LDA	x	•	2	3	4	5	•
LDS		•	3	4	5	6	•
LDX		•	3	4	5	6	•
LSR		2	•	•	6	7	•
NEG		2	•	•	6	7	•
NOP		•	•	•	•	•	2
ORA	x	•	2	3	4	5	•
PSH		•	•	•	•	•	4
PUL		•	•	•	•	•	4
ROL		2	•	•	6	7	•
ROR		2	•	•	6	7	•
RTI		•	•	•	•	•	10
RTS		•	•	•	•	•	5
SBA		•	•	•	•	•	2
SBC	x	•	2	3	4	5	•
SEC		•	•	•	•	•	2
SEI		•	•	•	•	•	2
SEV		•	•	•	•	•	2
STA	x	•	•	4	5	6	•
STS		•	•	5	6	7	•
STX		•	•	5	6	7	•
SUB	x	•	2	3	4	5	•
SWI		•	•	•	•	•	12
TAB		•	•	•	•	•	2
TAP		•	•	•	•	•	2
TBA		•	•	•	•	•	2
TPA		•	•	•	•	•	2
TST		2	•	•	6	7	•
TSX		•	•	•	•	•	4
TSX		•	•	•	•	•	4
WAI		•	•	•	•	•	9

NOTE Interrupt time is 12 cycles from the end of the instruction being executed, except following a WAI instruction Then it is 4 cycles

SUMMARY OF CYCLE-BY-CYCLE OPERATION

Table 8 provides a detailed description of the information present on the address bus, data bus, valid memory address line (VMA), and the read/write line ($R/\overline{W}$) 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 modes 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.)

TABLE 8 — OPERATIONS SUMMARY

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	$R/\overline{W}$ Line	Data Bus
IMMEDIATE						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data
CPX LDS LDX	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Operand Data (High Order Byte)
		3	1	Op Code Address + 2	1	Operand Data (Low Order Byte)
DIRECT						
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data
CPX LDS LDX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	1	Address of Operand	1	Operand Data (High Order Byte)
		4	1	Operand Address + 1	1	Operand Data (Low Order Byte)
STA	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address
		3	0	Destination Address	1	Irrelevant Data (Note 1)
		4	1	Destination Address	0	Data from Accumulator
STS STX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand
		3	0	Address of Operand	1	Irrelevant Data (Note 1)
		4	1	Address of Operand	0	Register Data (High Order Byte)
		5	1	Address of Operand + 1	0	Register Data (Low Order Byte)
INDEXED						
JMP	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data
CPX LDS LDX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Operand Data (High Order Byte)
		6	1	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)

TABLE 8 — OPERATIONS SUMMARY (CONTINUED)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED (Continued)						
STA	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	1	Index Register Plus Offset	1	Current Operand Data
		6	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		7	1/0 (Note 3)	Index Register Plus Offset	0	New Operand Data (Note 3)
STS STX	7	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
		5	0	Index Register Plus Offset	1	Irrelevant Data (Note 1)
		6	1	Index Register Plus Offset	0	Operand Data (High Order Byte)
		7	1	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
JSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Offset
		3	0	Index Register	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Index Register	1	Irrelevant Data (Note 1)
		8	0	Index Register Plus Offset (w/o Carry)	1	Irrelevant Data (Note 1)
EXTENDED						
JMP	3	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	1	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data
CPX LDS LDX	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Operand Data (High Order Byte)
		5	1	Address of Operand + 1	1	Operand Data (Low Order Byte)
STA A STA B	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	1	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	0	Operand Destination Address	1	Irrelevant Data (Note 1)
		5	1	Operand Destination Address	0	Data from Accumulator
ASL LSR ASR NEG CLR ROL COM ROR DEC TST INC	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	1	Address of Operand	1	Current Operand Data
		5	0	Address of Operand	1	Irrelevant Data (Note 1)
		6	1/0 (Note 3)	Address of Operand	0	New Operand Data (Note 3)

TABLE 8 — OPERATIONS SUMMARY (CONTINUED)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED (Continued)						
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	1	Irrelevant Data (Note 1)
		5	1	Address of Operand	0	Operand Data (High Order Byte)
		6	1	Address of Operand + 1	0	Operand Data (Low Order Byte)
JSR	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	1	Subroutine Starting Address	1	Op Code of Next Instruction
		5	1	Stack Pointer	0	Return Address (Low Order Byte)
		6	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		7	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		8	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		9	1	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
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	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
DES DEX INS INX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Previous Register Contents	1	Irrelevant Data (Note 1)
		4	0	New Register Contents	1	Irrelevant Data (Note 1)
PSH	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Accumulator Data
		4	0	Stack Pointer – 1	1	Accumulator Data
PUL	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Operand Data from Stack
TSX	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	0	New Index Register	1	Irrelevant Data (Note 1)
TXS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	0	Index Register	1	Irrelevant Data
		4	0	New Stack Pointer	1	Irrelevant Data
RTS	5	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Address of Next Instruction (High Order Byte)
		5	1	Stack Pointer + 2	1	Address of Next Instruction (Low Order Byte)

TABLE 8 — OPERATIONS SUMMARY (CONCLUDED)

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT (Continued)						
WAI	9	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Op Code of Next Instruction
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6	1	Contents of Cond. Code Register
RTI	10	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 2)
		3	0	Stack Pointer	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer + 1	1	Contents of Cond. Code Register from Stack
		5	1	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	1	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	1	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	1	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	1	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	1	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Irrelevant Data (Note 1)
		3	1	Stack Pointer	0	Return Address (Low Order Byte)
		4	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	1	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	1	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	1	Stack Pointer – 4	0	Contents of Accumulator A
		8	1	Stack Pointer – 5	0	Contents of Accumulator B
		9	1	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	0	Stack Pointer – 7	1	Irrelevant Data (Note 1)
		11	1	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	1	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE						
BCC BHI BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMI BVS	4	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Op Code Address + 2	1	Irrelevant Data (Note 1)
		4	0	Branch Address	1	Irrelevant Data (Note 1)
BSR	8	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Branch Offset
		3	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		4	1	Stack Pointer	0	Return Address (Low Order Byte)
		5	1	Stack Pointer – 1	0	Return Address (High Order Byte)
		6	0	Stack Pointer – 2	1	Irrelevant Data (Note 1)
		7	0	Return Address of Main Program	1	Irrelevant Data (Note 1)
		8	0	Subroutine Address (Note 4)	1	Irrelevant Data (Note 1)

NOTES:

1. If device which is addressed during this cycle uses VMA, then the Data Bus will go to the high-impedance three-state condition. Depending on bus capacitance, data from the previous cycle may be retained on the Data Bus.
2. Data is ignored by the MPU.
3. For TST, VMA = 0 and Operand data does not change.
4. MS Byte of Address Bus = MS Byte of Address of BSR instruction and LS Byte of Address Bus = LS Byte of Sub-Routine Address.

MOTOROLA

MC6803E

Advance Information

8-BIT MICROPROCESSOR

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

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

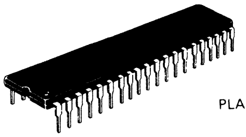
ORDERING INFORMATION ($T_A = 0°C$ to $70°C$)

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

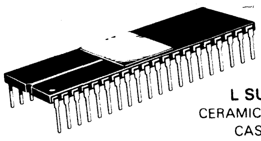
HMOS

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

8-BIT
MICROPROCESSOR

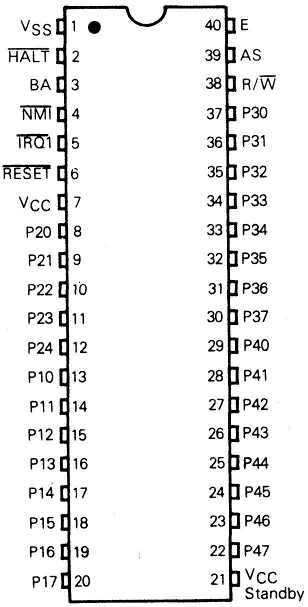


G SUFFIX
PLASTIC PACKAGE
CASE 711



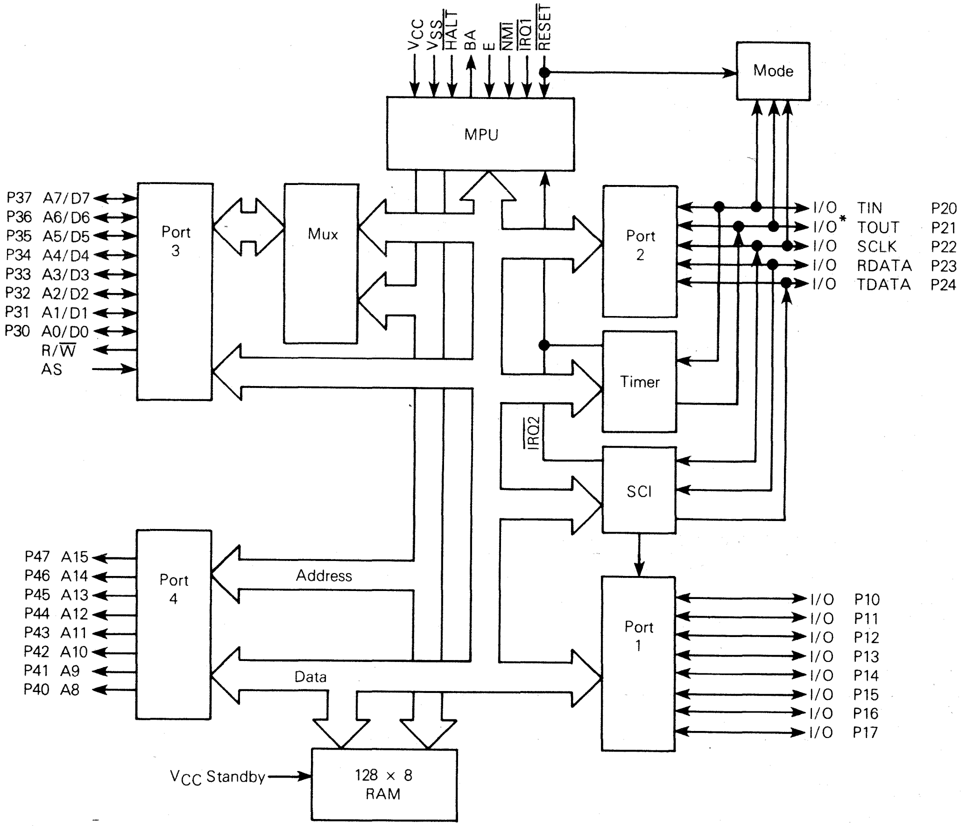
L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT



ADI-912

BLOCK DIAGRAM



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

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	− 0.3 to + 7.0	V
Input Voltage	V_{in}	− 0.3 to + 7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	− 55 to + 150	°C

THERMAL CHARACTERISTICS

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

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}.

POWER CONSIDERATIONS

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

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

$P_D = K \div (T_J + 273°C)$ (2)

Solving equations 1 and 2 for K gives:

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

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

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

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

* Except mode programming levels; see Figure 8.

PERIPHERAL PORT TIMING (Refer to Figures 1 and 2)

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

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

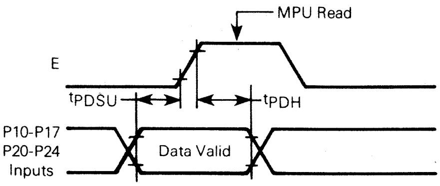
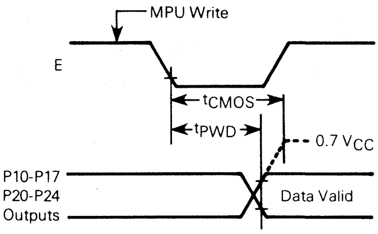


FIGURE 2 — DATA SETUP AND HOLD TIMES (MPU WRITE)



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

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

FIGURE 3 — CMOS LOAD

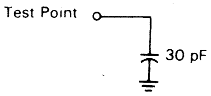
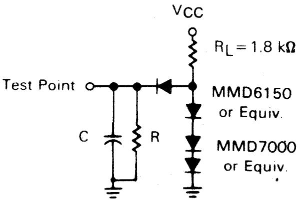


FIGURE 4 — TIMING TEST LOAD PORTS 1, 2, 3, 4

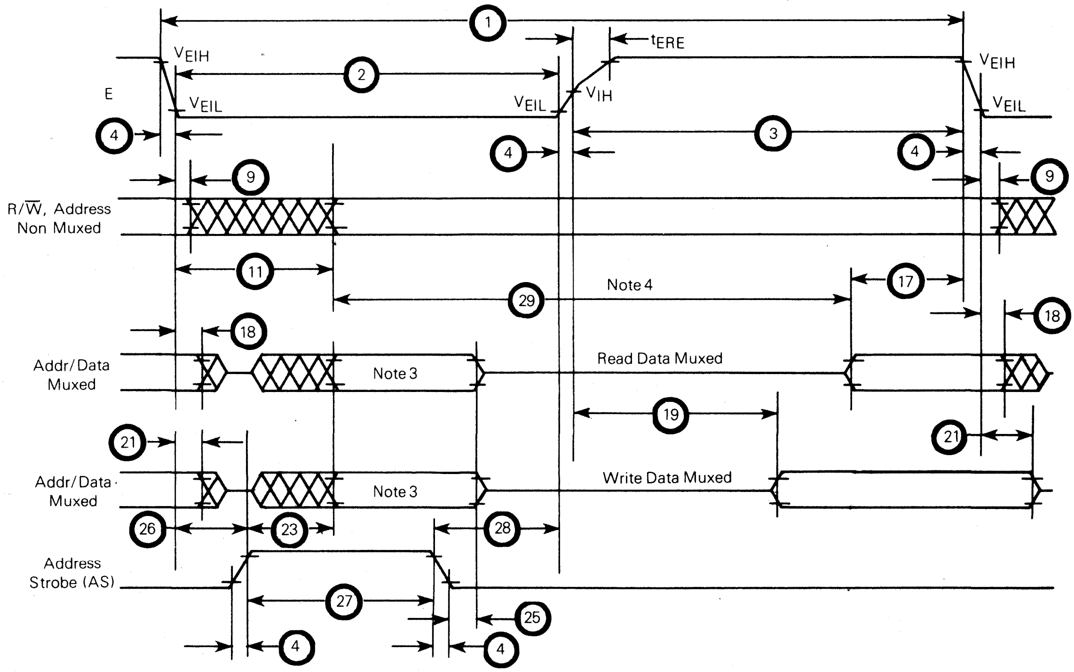


C = 90 pF for P30-P37, P40-P47, R/$\overline{W}$
= 30 pF for P10-P17, P20-P24, BA
R = 24 kΩ for P10-P17, P20-P24,
P30-P37, P40-P47, R/$\overline{W}$, BA

BUS TIMING (See Notes 1 and 2)

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

FIGURE 5 — BUS TIMING DIAGRAM



NOTES:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Address valid on the occurrence of the latter of 11 or 23.
4. Usable access time is computed by smaller of 1 – (4 + 11 + 17) or 1 – (4 + 17 + 23 + 26).

INTRODUCTION

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

The MC6803E has one 10-bit port, two 8-bit ports, and one 5-bit port. Each port except port 3 and port 4 consists of at least a write-only data direction register and a data register. The data direction register is used to define whether corresponding bits in the data register are configured as an input (clear) or output (set).

The term "port," by itself, refers to all of the hardware associated with the port. When the port is used as a "data port" or an "I/O port," it is controlled by the port data direction register and the programmer has direct access to the port pins using the port data register. Port 3 functions as a time multiplexed address/data bus and does not contain either a data direction register or a data register. Port 4 functions as a non-multiplexed high order address bus and does not contain either a data direction register or a data register. Port pins are labeled as Pij, where i identifies one of four ports and j indicates the particular bit.

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

FIGURE 6 — PROGRAMMING MODEL

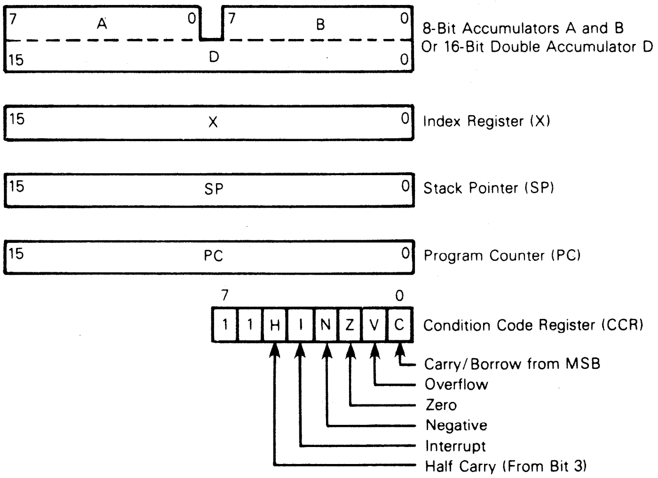


TABLE 1 — NEW INSTRUCTIONS

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

OPERATING MODES

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

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

Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of address strobe (AS) and data valid while E is high. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 7. This allows port 3 to function as a data bus when E is high.

Figure 8 depicts a typical operating configuration.

PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

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

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

FIGURE 7 — TYPICAL LATCH ARRANGEMENT

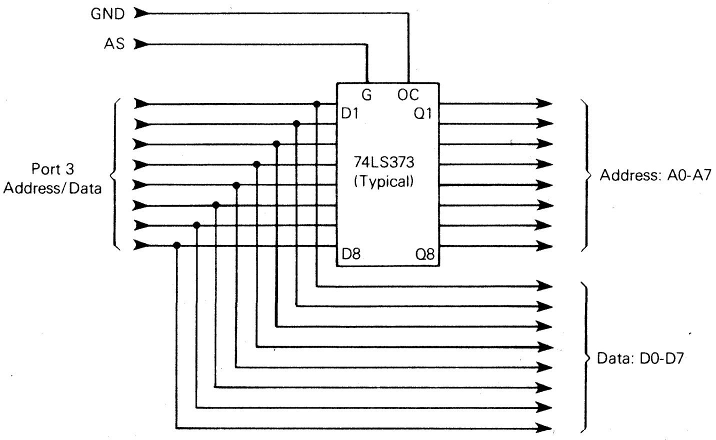
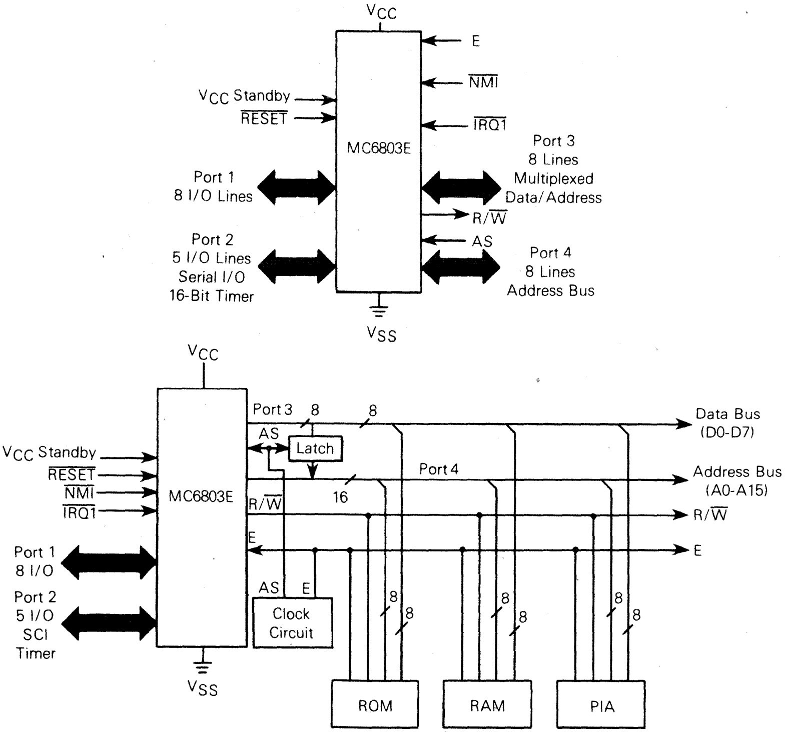
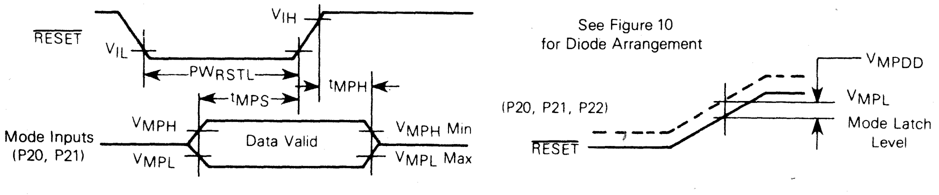


FIGURE 8 — EXPANDED MULTIPLEXED CONFIGURATION



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

FIGURE 9 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 9)

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

TABLE 2 — SUMMARY OF MC6803E OPERATING MODES

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

TABLE 3 — MODE SELECTION SUMMARY

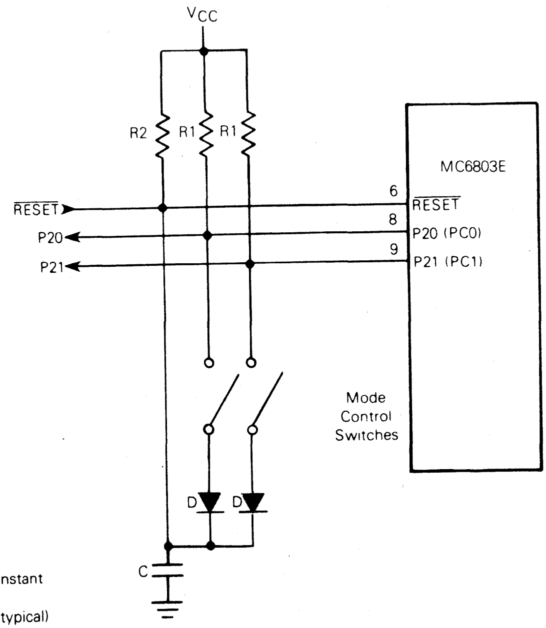
Mode	P21 PC1	P20 PC0	RAM	Interrupt Vectors	Bus Mode	Operating Mode
3	H	H	E	E	MUX	Multiplexed/No RAM
2	H	L	I	E	MUX	Multiplexed/RAM
1	L	H				Undefined*
0	L	L				Undefined*

Legend:

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

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

FIGURE 10 — TYPICAL MODE PROGRAMMING CIRCUIT

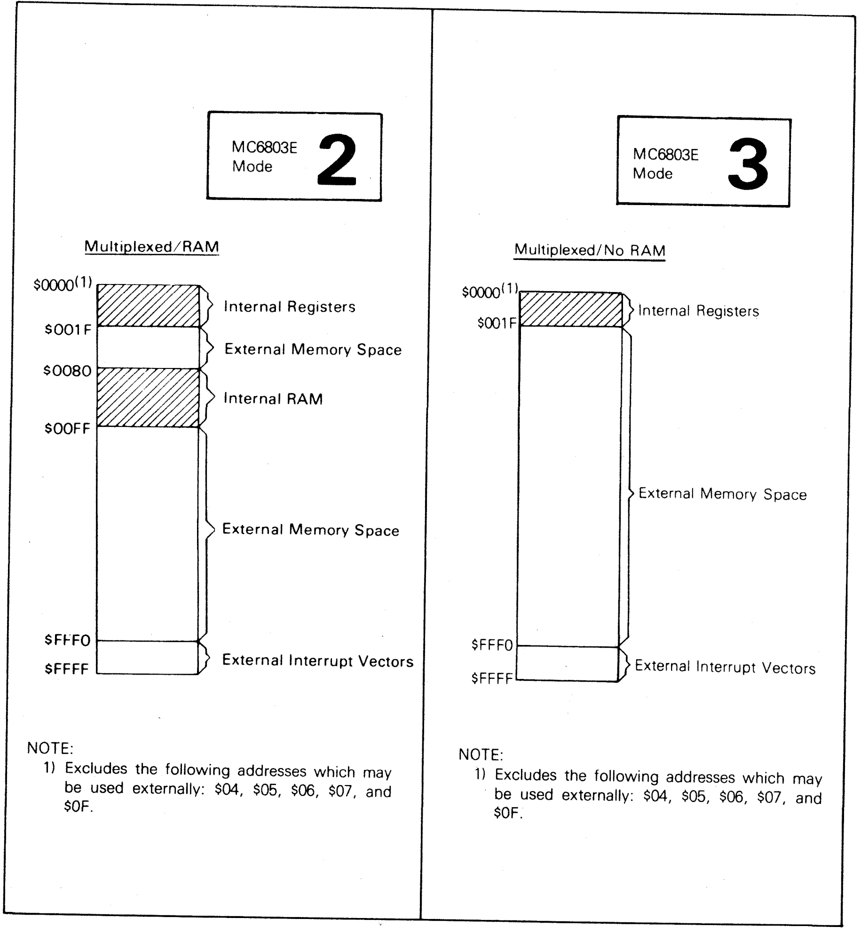


NOTES:
1. Mode 3 as shown
2. R2•C = reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

MEMORY MAPS

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

FIGURE 11 — MC6803E MEMORY MAPS



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

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

TABLE 4 — INTERNAL REGISTER AREA

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

*1 = Output, 0 = Input

MC6803E INTERRUPTS

The MC6803E supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt ($\overline{\text{NMI}}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I bit and by individual enable bits. The I bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$. The programmable timer and serial communications interface use an internal $\overline{\text{IRQ2}}$ interrupt line, as shown in the block diagram. External devices use $\overline{\text{IRQ1}}$. An $\overline{\text{IRQ1}}$ interrupt is serviced before $\overline{\text{IRQ2}}$ if both are pending.

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

The interrupt flowchart is depicted in Figure 12 and is common to every interrupt excluding reset. During interrupt servicing, the program counter, A accumulator, B accumulator, and condition code register are pushed onto the stack. The I bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. Interrupt and $\overline{\text{RESET}}$ timing are illustrated in Figures 13 and 14.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

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

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

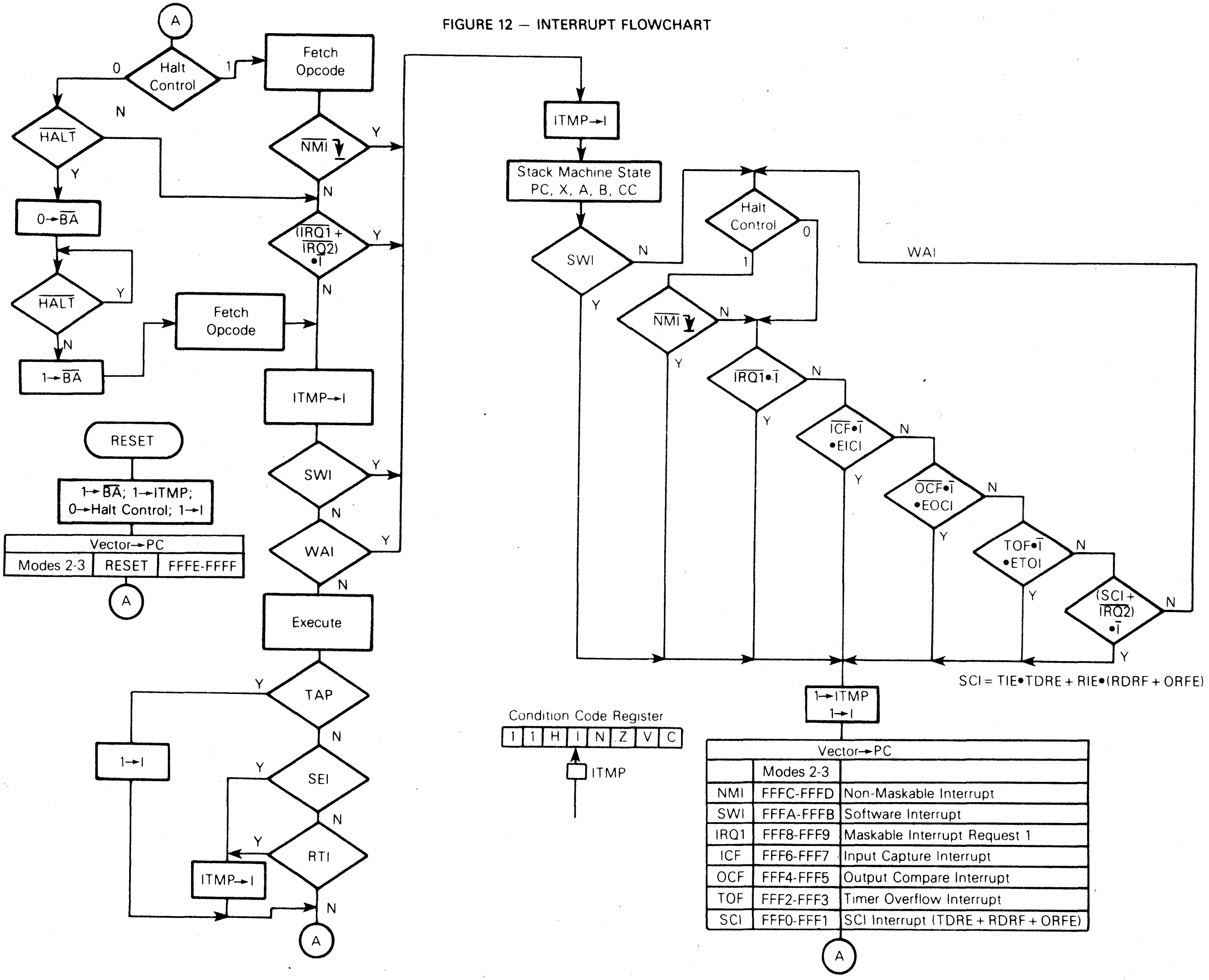
V_{CC} standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide +5 volts (± 5%)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

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

AS (ADDRESS STROBE)

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

FIGURE 12 — INTERRUPT FLOWCHART



Vector→PC		
	Modes 2-3	
NMI	FFFC-FFFD	Non-Maskable Interrupt
SWI	FFFA-FFFB	Software Interrupt
IRQ1	FFF8-FFF9	Maskable Interrupt Request 1
ICF	FFF6-FFF7	Input Capture Interrupt
OCF	FFF4-FFF5	Output Compare Interrupt
TOF	FFF2-FFF3	Timer Overflow Interrupt
SCI	FFF0-FFF1	SCI Interrupt (TDRE + RDRF + ORFE)

FIGURE 13 — INTERRUPT SEQUENCE

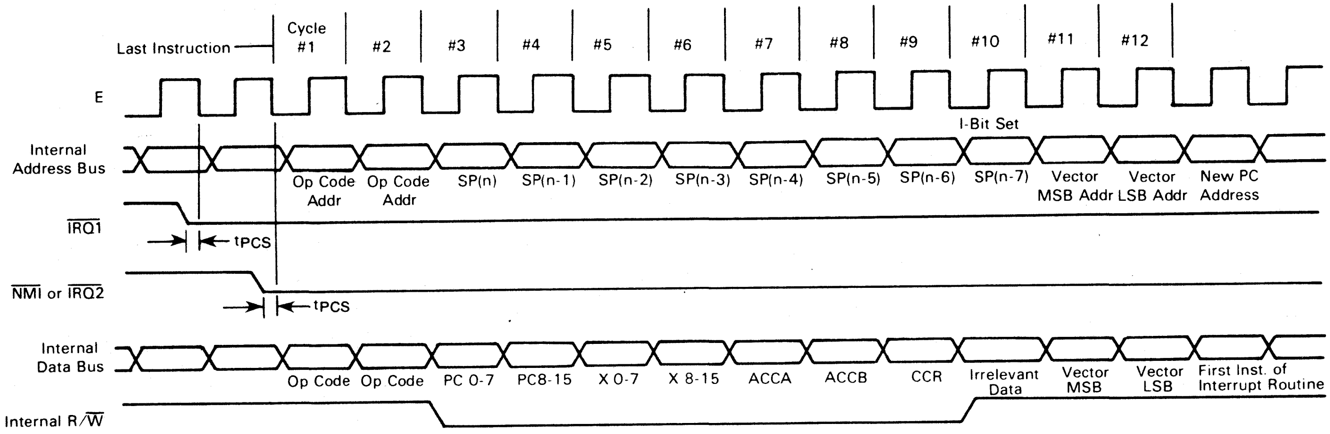
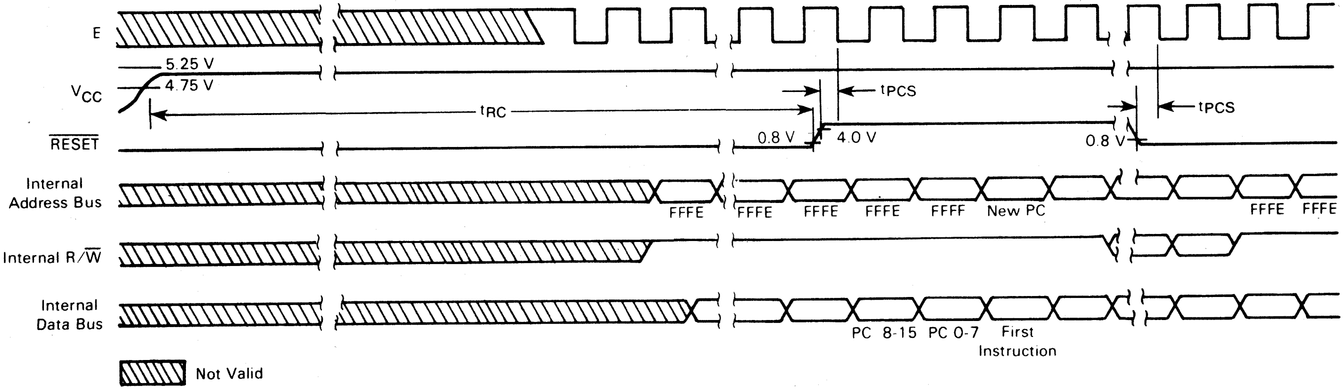


FIGURE 14 — $\overline{\text{RESET}}$ TIMING



$\overline{\text{HALT}}$

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

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

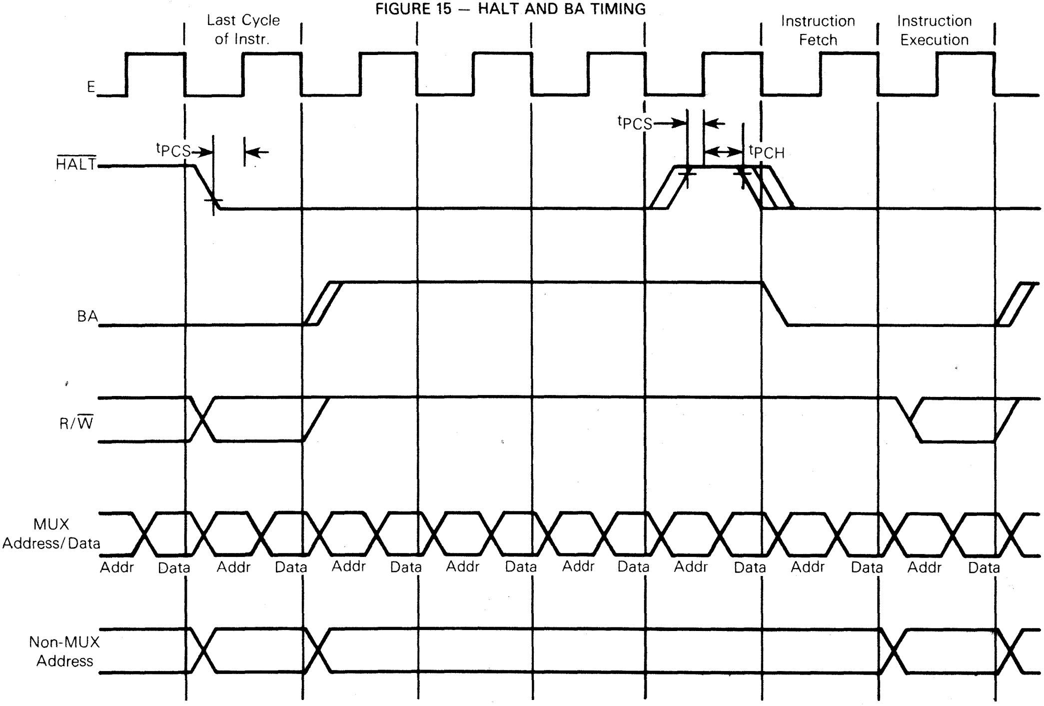
BA (BUS AVAILABLE)

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

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

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

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



$\overline{\text{RESET}}$

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

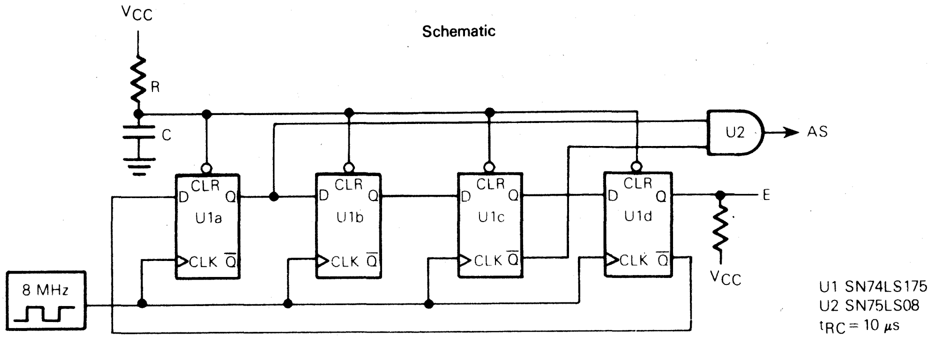
E (ENABLE)

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

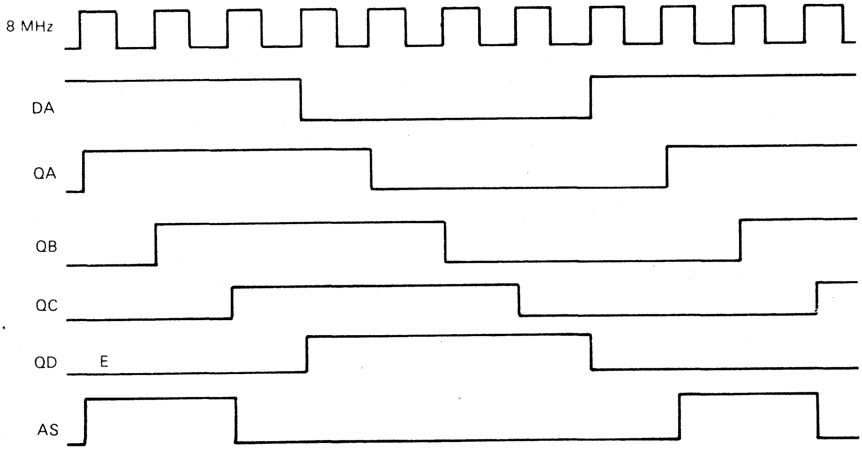
Enable is the primary MC6803E system timing signal and all timing data specified as cycles is assumed to be referenced to this clock unless otherwise noted.

FIGURE 16 — CLOCK CIRCUIT EXAMPLE 1

Schematic



Timing



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

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

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

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

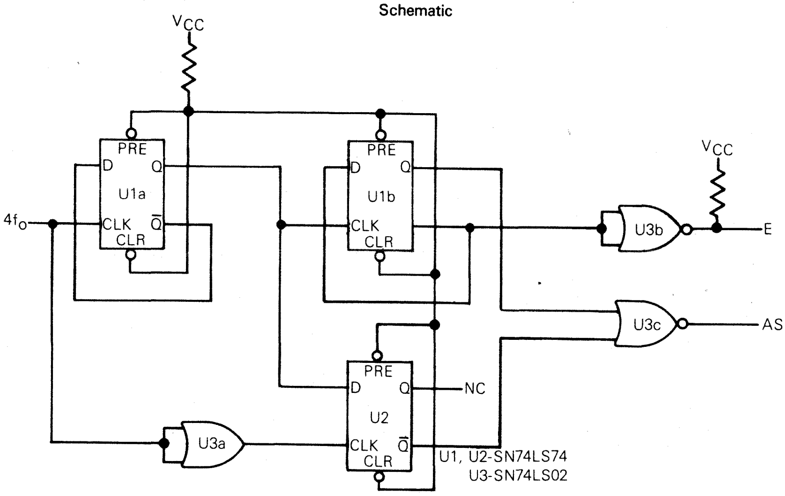
$\overline{\text{IRQ1}}$ typically requires an external 3.3 kilohm (nominal) resistor to V_{CC} for wire-OR applications. $\overline{\text{IRQ1}}$ has no internal pullup resistors.

P10-P17 (PORT 1)

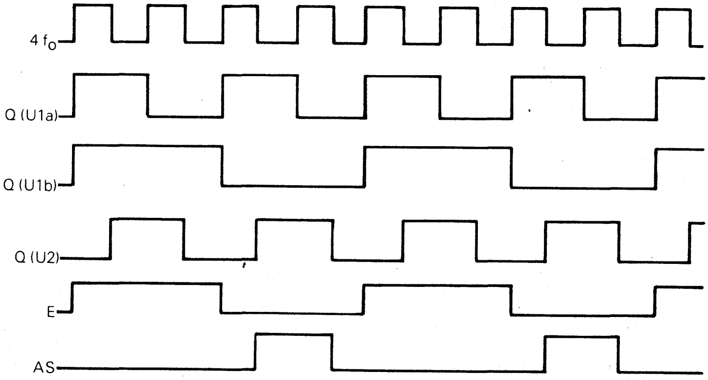
Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the port 1 data direction register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 picofarads, Darlington transistors, or CMOS devices using external pullup resistors. It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

FIGURE 17 — CLOCK CIRCUIT EXAMPLE 2

Schematic



Timing



P20-P24 (PORT 2)

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

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

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

PORT 2 DATA REGISTER

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

P30-P37 (PORT 3)

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

P40-P47 (PORT 4)

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

RESIDENT MEMORY

The MC6803E provides 128 bytes of on-board RAM. One half of the RAM is powered through the V_{CC} standby pin and is maintainable during V_{CC} power down. This standby portion of the RAM consists of 64 bytes located from $80 through $BF.

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

The RAM is controlled by the RAM control register.

RAM CONTROL REGISTER ($14)

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

RAM CONTROL REGISTER

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

Bit 0-5 Not used.

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

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

PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 18.

COUNTER ($09:0A)

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

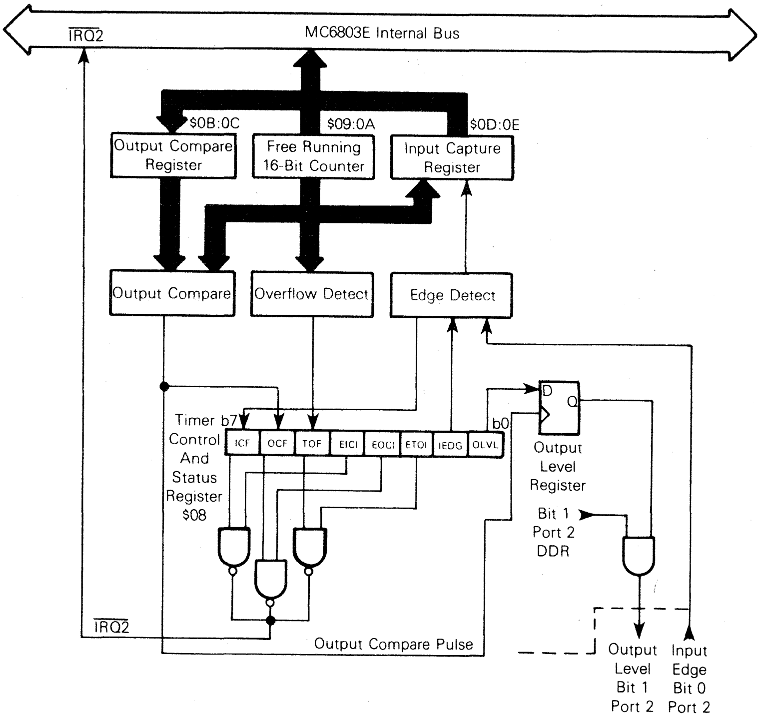
OUTPUT COMPARE REGISTER ($0B:0C)

The output compare register is a 16-bit read/write register used to control an output waveform or to provide an arbitrary timeout flag. It is compared with the free-running counter on each E cycle. When a match occurs, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1 is configured as an output, OLVL will appear at P21 and the output compare register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte ($0B) to ensure a valid compare. The output compare register is set to $FFFF at $\overline{\text{RESET}}$.

INPUT CAPTURE REGISTER ($0D:0E)

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

FIGURE 18 — BLOCK DIAGRAM OF PROGRAMMABLE TIMER



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

TIMER CONTROL AND STATUS REGISTERS ($08)

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

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

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

TIMER CONTROL AND STATUS REGISTER (TCSR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$08

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

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

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

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

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

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

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

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

SERIAL COMMUNICATIONS INTERFACE

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

WAKE-UP FEATURE

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

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

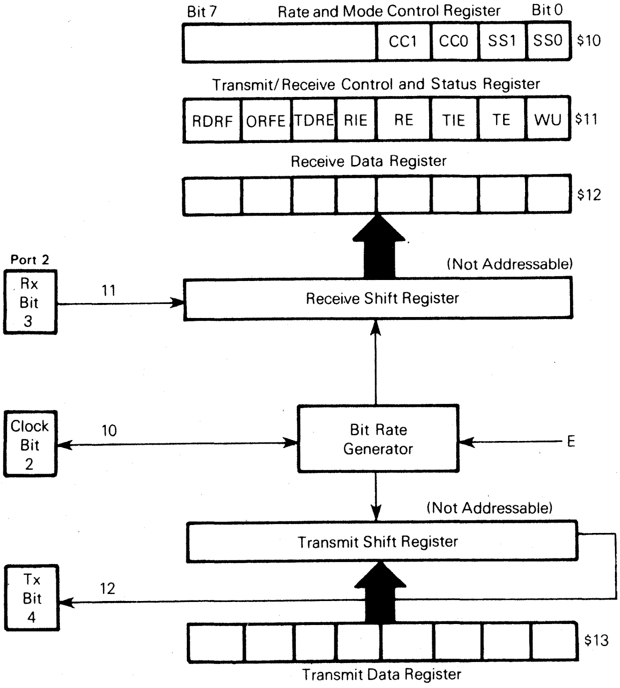
- format : standard mark/space (NRZ) or bi-phase
- clock: external or internal bit rate clock
- baud: one of 4 per E clock frequency, or external clock (8x desired baud)
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal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

SERIAL COMMUNICATIONS REGISTERS

The serial communications interface includes four addressable registers as depicted in Figure 19. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and a read-only receive register. The shift registers are not accessible to software.

RATE AND MODE CONTROL REGISTER (RMCR) ($10) — The rate and mode control register controls the SCI bit rate, format, clock source, and under certain conditions the configuration of P22. The register consists of four write-only bits which are cleared during reset. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

FIGURE 19 — SCI REGISTERS



RATE AND MODE CONTROL REGISTER

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

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

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

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

NOTE

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

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR) ($11) – The transmit/receive control and status register controls the transmitter, receiver, wake-up feature, and two individual interrupts, and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{\text{RESET}}$.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER

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

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

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

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

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

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

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

TABLE 6 – SCI BIT TIMES AND RATES

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

*Using maximum clock rate

TABLE 7 – SCI FORMAT AND CLOCK SOURCE CONTROL

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

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

Bit 7 **Receive Data Register Full (RDRF)** — RDRF is set when the input serial shift register is transferred to the receive data register, or during reset.

SERIAL OPERATIONS

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

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

The start bit (0), eight data bits (beginning with bit 0), and a stop bit (1) will be transmitted. If TDRE is still set when the next byte transfer should occur, ones will be sent until more data is provided. In bi-phase format, the output toggles at the start of each bit and at half-bit time when a one is sent. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. SCI data formats are illustrated in Figure 20.

INSTRUCTION SET

As stated earlier, the MC6803E is upward source and object code compatible with the MC6800. Execution times of key instructions have been reduced and several new instructions have been added, including a hardware multiply.

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

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

PROGRAMMING MODEL

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

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

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

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

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

FIGURE 20 — SCI DATA FORMATS

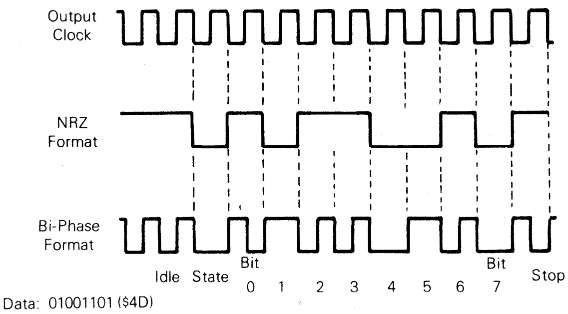


TABLE 8 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	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.

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

ADDRESSING MODES

Six addressing modes can be used to reference memory. A summary of the addressing modes for all instructions is presented in Tables 9 through 12, where execution times are provided in E cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 megahertz, one E cycle is equivalent to one microsecond. A description of selected instructions is shown in Figure 21.

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

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

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

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

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

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

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

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

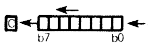
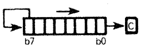
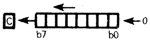
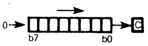
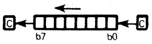
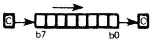
Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry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

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

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The condition code register notes are listed after Table 12

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	Mnemonic	Direct			Relative			Index			Extnd			Inherent			Branch Test	Cond. Code Reg.					
																		5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA				20	3	2										None	●	●	●	●	●	●
Branch Never	BRN				21	3	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC				24	3	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS				25	3	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ				27	3	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE				2C	3	2										N⊕V = 0	●	●	●	●	●	●
Branch If > Zero	BGT				2E	3	2										Z + (N⊕V) = 0	●	●	●	●	●	●
Branch If Higher	BHI				22	3	2										C + Z = 0	●	●	●	●	●	●
Branch If Higher or Same	BHS				24	3	2										C = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE				2F	3	2										Z + (N⊕V) = 1	●	●	●	●	●	●
Branch If Carry Set	BLO				25	3	2										C = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT				2D	3	2										N⊕V = 1	●	●	●	●	●	●
Branch If Minus	BMI				2B	3	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC				28	3	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS				29	3	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL				2A	3	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR				8D	6	2										See Special Operations - Figure 21	●	●	●	●	●	●
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		↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1	See Special Operations - Figure 21	●	●	●	●	●	●
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	Cond. Code Reg.					
						5	4	3	2	1	0
	Mnemonic	OP	~	#		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
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
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

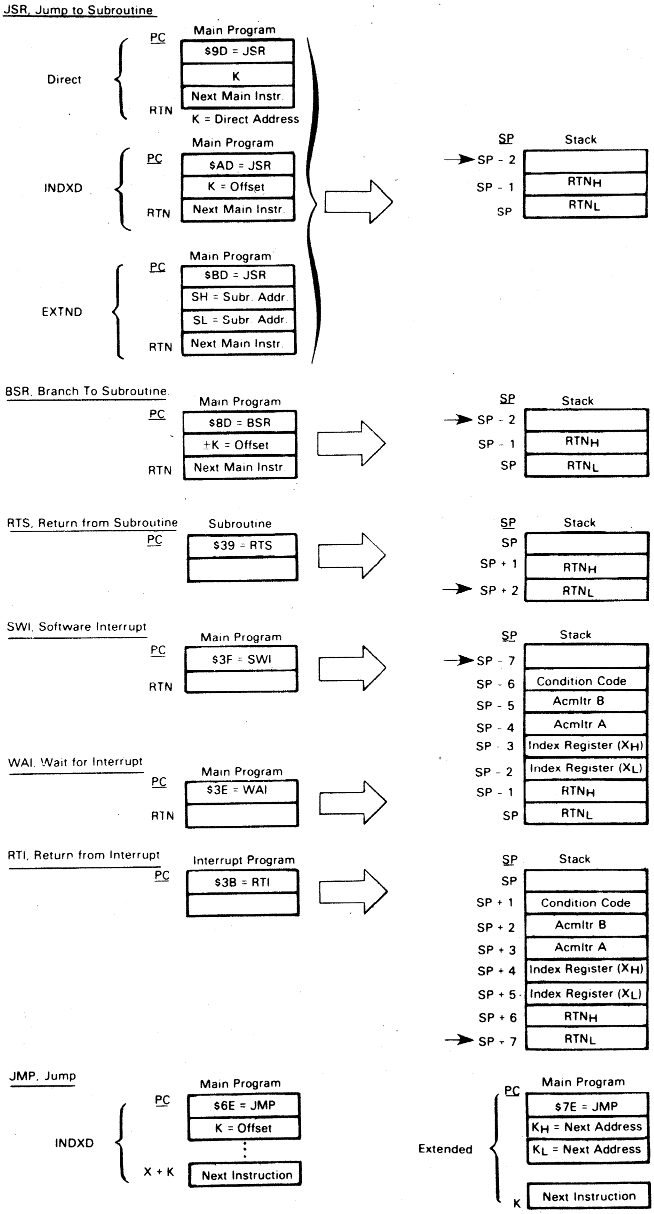
H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

FIGURE 21 — SPECIAL OPERATIONS



Legend:

RTN = Address of next instruction in main program to be executed upon return from subroutine
$\mathrm{RTN_H}$ = Most significant byte of return address
$\mathrm{RTN_L}$ = Least significant byte of return address
→ = Stack pointer after execution
K = 8-bit unsigned value

SUMMARY OF CYCLE-BY-CYCLE OPERATION

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

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

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

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

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

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

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		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)

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

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

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

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

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

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

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

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	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)

MOTOROLA

MC6804P2

Product Preview

8-BIT MICROCOMPUTER UNIT

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

HARDWARE FEATURES

- 8-Bit Architecture
- Pin Compatible with the MC6805P2, MC68705P3, and MC68HC04P2
- 32 Bytes of RAM
- Memory Mapped I/O
- 1024 Bytes of User ROM
- 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

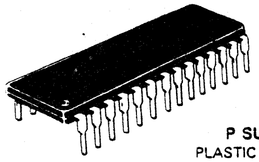
- 20 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Vectored Interrupts: Timer, Software, and External
- Mask Selectable Edge- or Level-Sensitive Interrupt Pin

EXORciser is a registered trademark of Motorola Inc.

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

**8-BIT
MICROCOMPUTER**



P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENTS

V_{SS}	1	28	$\overline{\text{RESET}}$
$\overline{\text{INT}}$	2	27	PA7
V_{CC}	3	26	PA6
EXTAL	4	25	PA5
XTAL	5	24	PA4
NUM	6	23	PA3
TIMER	7	22	PA2
PC0	8	21	PA1
PC1	9	20	PA0
PC2	10	19	PB7
PC3	11	18	PB6
PB0	12	17	PB5
PB1	13	16	PB4
PB2	14	15	PB3

NP-369

MOTOROLA

MC6805P2

Advance Information

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT MICROCOMPUTER

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. A comparison of the key features of several members of the M6805 Family is shown on the last page of this data sheet. The following are some of the hardware and software highlights of the MC6805P2 MCU.

HARDWARE FEATURES

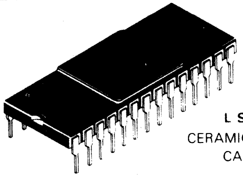
- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 1100 Bytes of User ROM
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Self-Check Mode
- Zero Crossing Detection
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply

SOFTWARE FEATURES

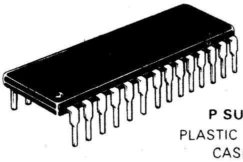
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instruction
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O

USER SELECTABLE OPTIONS

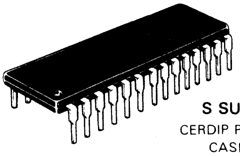
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits, 2^n)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vectored Interrupts: Timer, Software, and External



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

FIGURE 1 — PIN ASSIGNMENTS

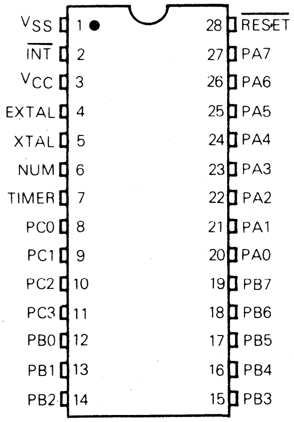
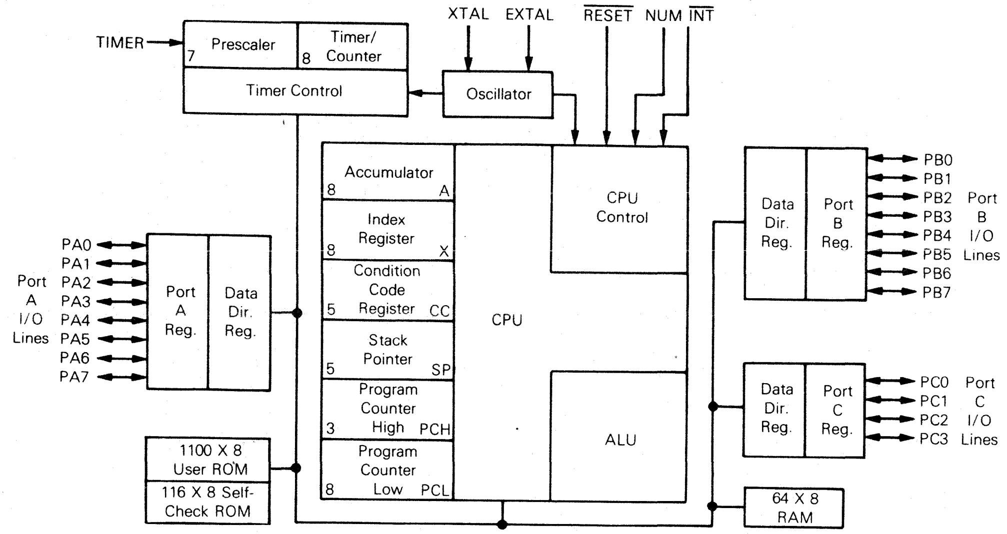


FIGURE 2 — MC6805P2 HMOS MICROCOMPUTER BLOCK DIAGRAM



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		150	
Ceramic	T_J	175	°C
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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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		120	
Ceramic	θ_{JA}	50	°C/W
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	V_{IH}				V
$\overline{\text{RESET}}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	–	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{\text{INT}}$ ($4.75 \le V_{CC} \le 5.75$)		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}	V
Self-Check Mode		–	9.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{\text{RESET}}$		−0.3	–	0.8	
$\overline{\text{INT}}$		−0.3	*	1.5	
All Other		−0.3	–	0.8	
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figures 11, 12, and 13)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{\text{INT}}$ 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	C_{in}				pF
XTAL		–	25	–	
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	–	3.5	–	V
Input Current	I_{in}				µA
TIMER ($V_{in}=0.4$ V)		–	–	20	
$\overline{\text{INT}}$ ($V_{in}=2.4$ 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	
$\overline{\text{RESET}}$ ($V_{in}=0.8$ V) (External Capacitor Charging Current)		−4.0	–	−50	

*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}	3.5	–	–	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}	0.3	–	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	µA

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 MC68A05P2 MC68B05P2	f_{osc}	0.4 0.4 0.4	– – –	4.2 6.0 8.0	MHz
Cycle Time ($4/f_{osc}$)		t_{cyc}	0.95	–	10	µs
$\overline{\text{INT}}$ and TIMER Pulse Width (See Interrupt Section)		t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{\text{RESET}}$ Pulse Width		t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Capacitance = 1.0 µF)		t_{RHL}	–	100	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (±5° Accuracy)		f_{INT}	0.03	–	1.0	kHz
External Clock Input Duty Cycle (EXTAL)		–	40	50	60	%

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

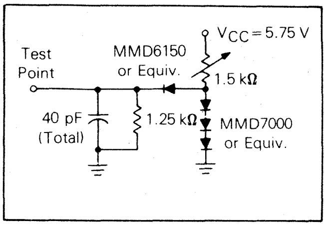


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

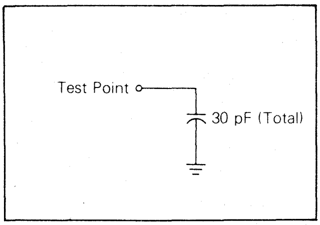
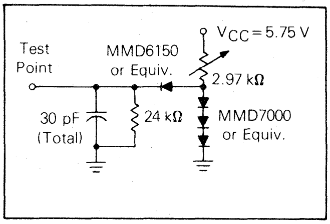


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$

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.

$\overline{\text{RESET}}$

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 6, 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 7. 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.

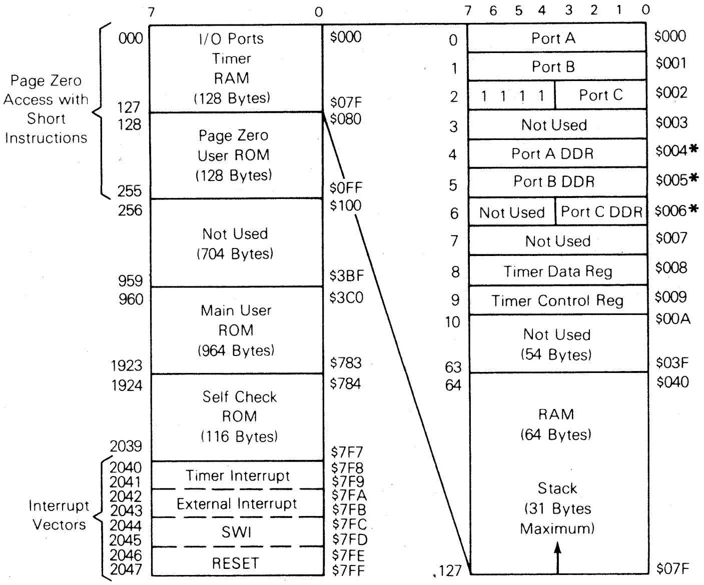
CENTRAL PROCESSING UNIT

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

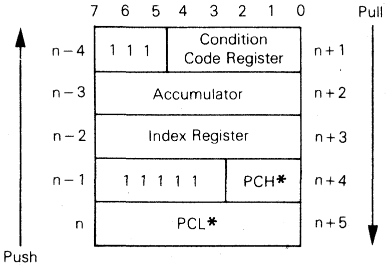
The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 8 and are explained in the following paragraphs.

FIGURE 6 — MCU ADDRESS MAP



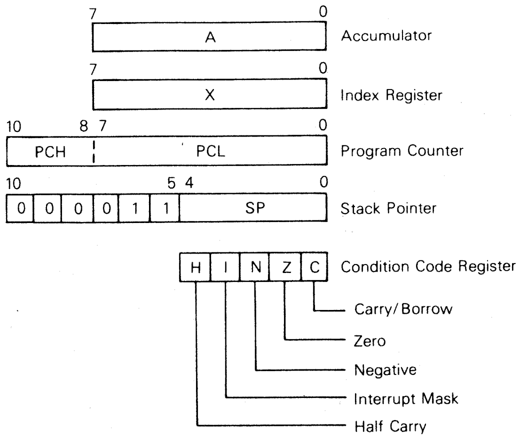
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 7 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH are stacked.

FIGURE 8 — 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 ($\overline{\text{INT}}$). 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 9. 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.

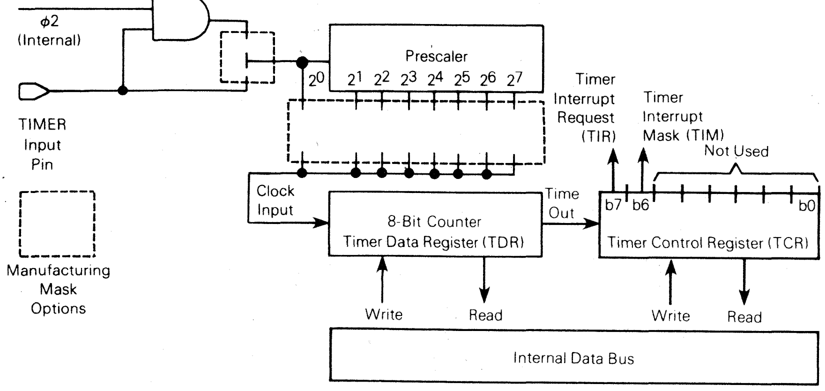
The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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 9 — 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 10 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 ($\overline{\text{RESET}}$), and by an optional internal low voltage detect circuit; see Figure 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 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 12.

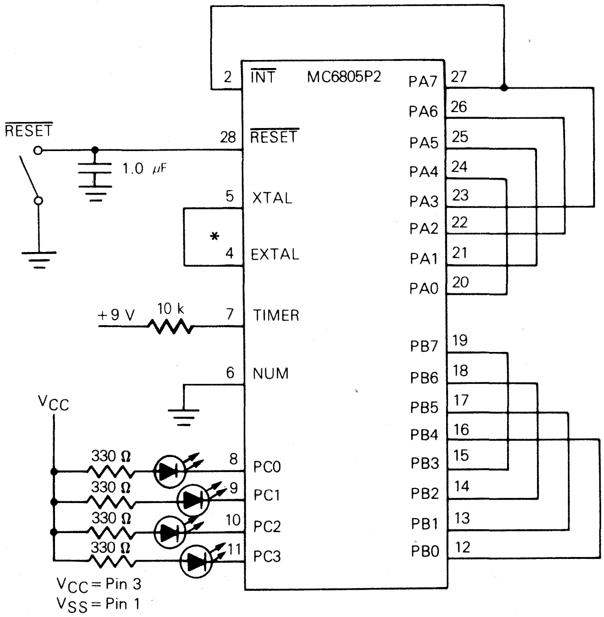
During power-up, a delay of t_{RHL} is needed before allowing the $\overline{\text{RESET}}$ input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the $\overline{\text{RESET}}$ input, as shown in Figure 13, typically provides sufficient delay. See Figure 17 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 14. The crystal specifications and suggested PC board layouts

FIGURE 10 — SELF-CHECK CONNECTIONS



*This connection depends on the clock oscillator user selectable mask option. Use crystal if that 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 11 — POWER AND $\overline{\text{RESET}}$ TIMING

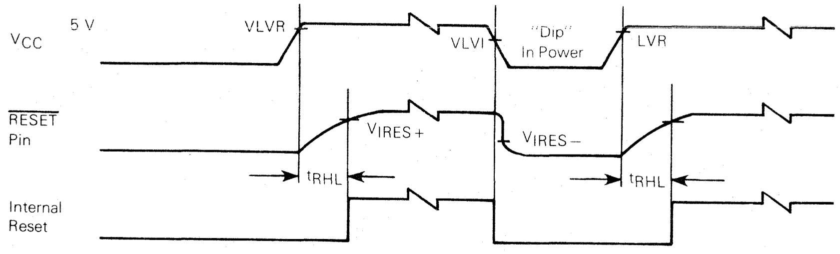


FIGURE 12 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

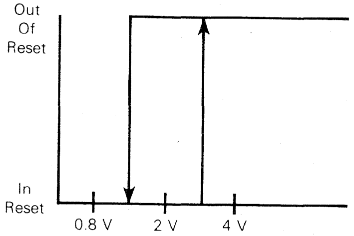


FIGURE 13 — POWER-UP RESET DELAY CIRCUIT

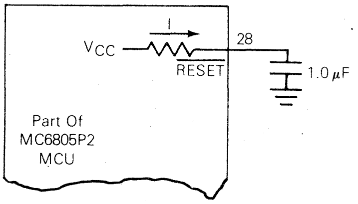
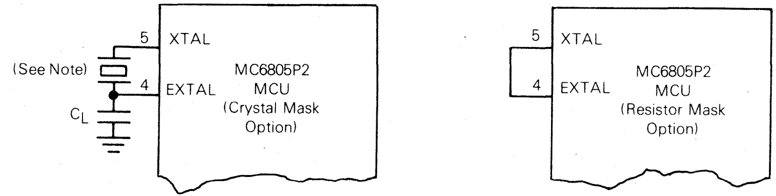
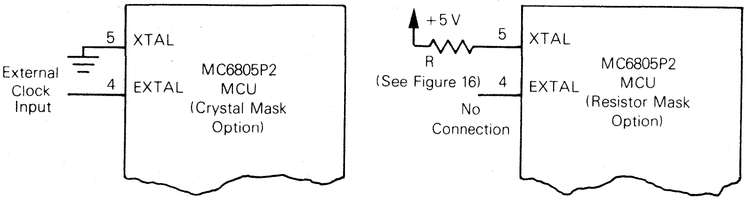


FIGURE 14 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% to 50% Accuracy
Typical $t_{cyc} = 1.25\ \mu s$
External Jumper

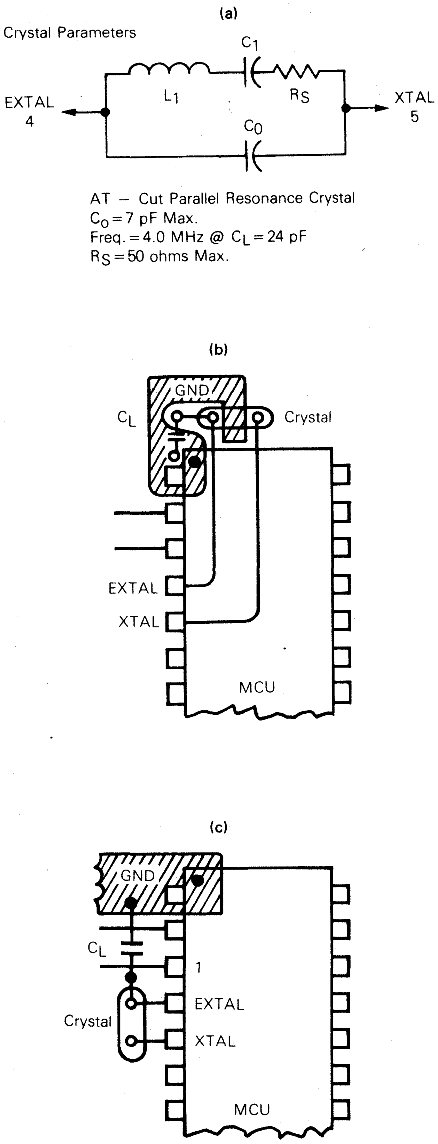


External Clock

Approximately 10% to 25% Accuracy
External Resistor
(Excludes Resistor Tolerence)

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.

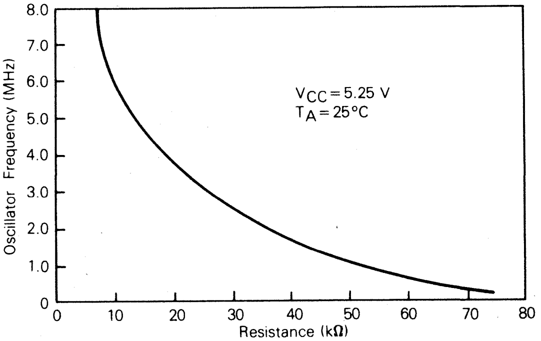
FIGURE 15 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



AT — Cut Parallel Resonance Crystal
$C_0 = 7$ 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 16 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P2 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 17. 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 $\overline{\text{RESET}}$, 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 (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 18(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. 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 $\overline{\text{INT}}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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 17 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

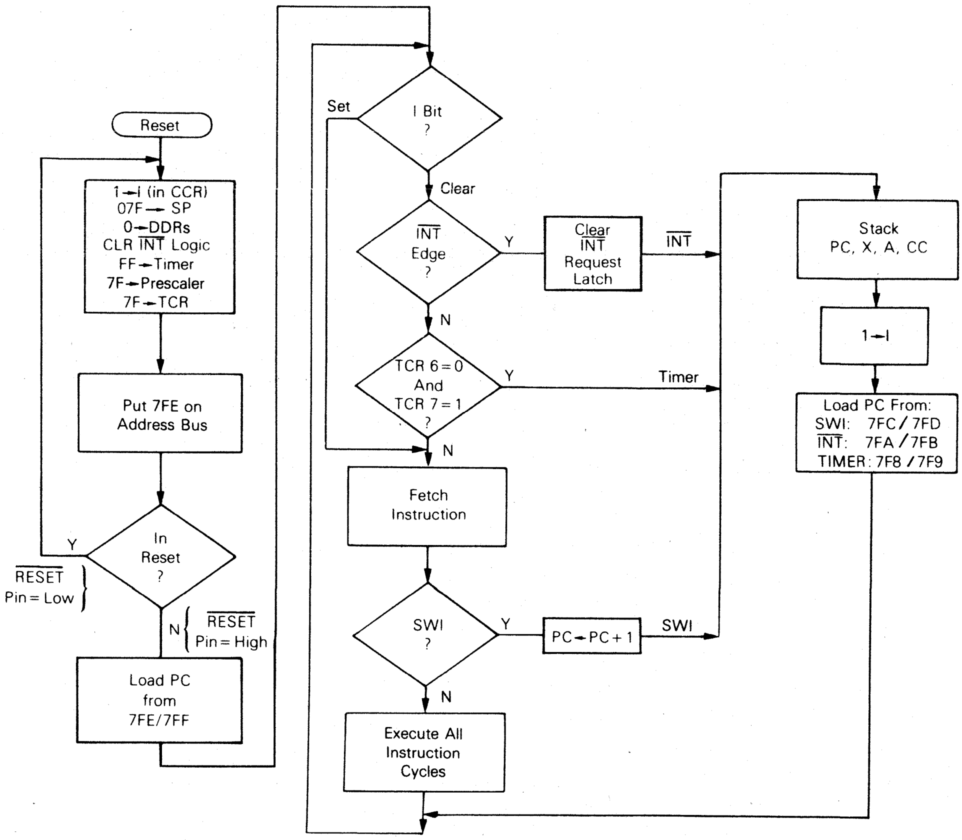
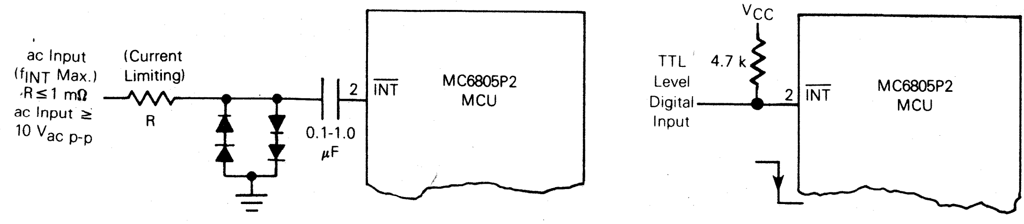


FIGURE 18 — TYPICAL INTERRUPT CIRCUITS


(a) Zero-Crossing Interrupt

(b) Digital-Signal Interrupt

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 18(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. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{\text{INT}}$ 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 19. When port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

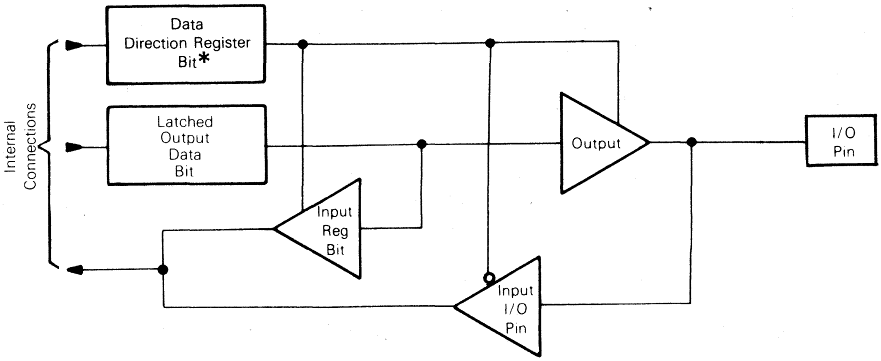
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 6 gives the address of data registers and DDRs. The register configuration is provided in Figure 20 and Figure 21 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 19) 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 19 — 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 20 — MCU REGISTER CONFIGURATION

PORT DATA REGISTER

7	0

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002 (Bits 0→3)

PORT DATA DIRECTION REGISTER (DDR)

7	0

(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 (Bits 0→3)

TIMER CONTROL REGISTER (TCR)

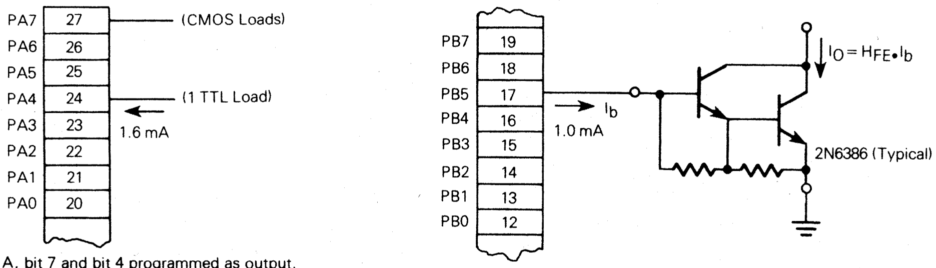
7	6	5	4	3	2	1	0	
		1	1	1	1	1	1	$009

TCR7 — Timer Interrupt Status 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.

TIMER DATA REGISTER (TDR)

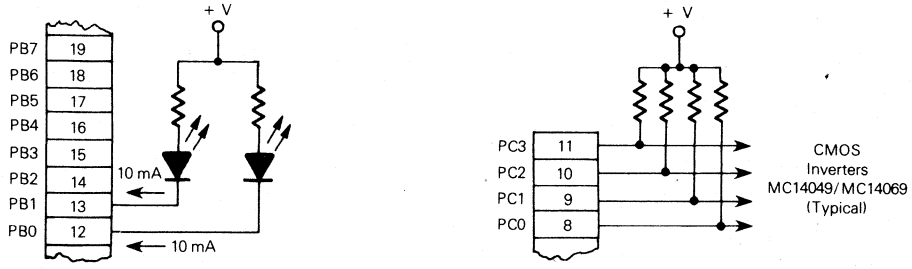
7	0	
MSB	LSB	$008

FIGURE 21(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.



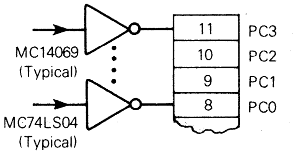
Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

Port C, bits 0-3 programmed as output, driving CMOS loads, using external pullup resistors.

FIGURE 21(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.

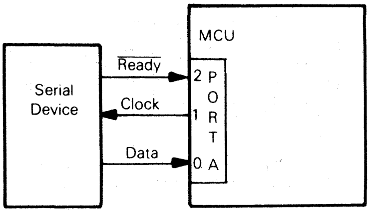
SOFTWARE

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 22 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 22 — BIT MANIPULATION EXAMPLE



```
                 •
                 •
                 •
SELF  BRSET  2, PORTA, SELF
                 •
                 •
                 •
      BSET   1, PORTA
      BRCLR  0, PORTA, CONT
CONT  BCLR   1, PORTA
      ASR    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

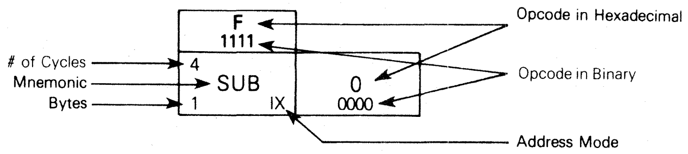
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

	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



ORDERING INFORMATION

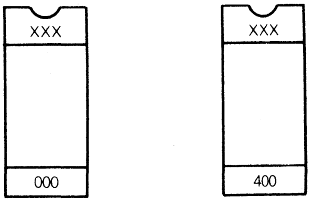
The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

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 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.

MC6805P2 MCU 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
- ☐ 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 Oscillator Input
- ☐ Crystal
- ☐ Resistor

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 ______________________

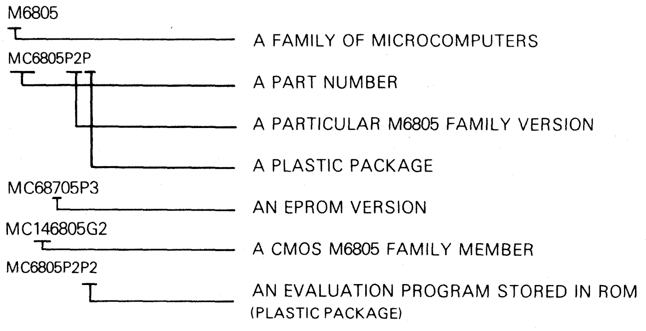
Signature ______________________

M6805 FAMILY

	MC6805P2	MC6805P4	MC6805R2	MC6805T2	MC6805U2
Technology	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	28	40
On-Chip RAM (Bytes)	64	112	64	64	64
On-Chip User ROM (Bytes)	1.1K	1.1K	2K	2.5K	2K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	19	24
Unidirectional I/O Lines	None	None	6 Inputs	None	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, PLL	Timer
External Interrupt Inputs	1	1	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705R3	None	MC68705U3
STOP and WAIT	No	No	No	No	No

	MC68705P3	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2
Technology	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS
Number of Pins	28	40	40	40	28	40
On-Chip RAM (Bytes)	112	112	112	112	64	112
On-Chip User ROM (Bytes)	1.8K EPROM	3.8K EPROM	3.8K EPROM	None	1K	2K
External Bus	None	None	None	Yes	None	None
Bidirectional I/O Lines	20	24	24	16	16	32
Unidirectional I/O Lines	None	6 Inputs	8 Inputs	None	4 Inputs	None
Other I/O Features	Timer	Timer, A/D	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	2	2	1	1	1
EPROM Version	–	–	–	None	None	None
STOP and WAIT	No	No	No	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE



MOTOROLA

MC6805P4

Advance Information

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. A comparison of the key features of several members of the M6805 Family is shown on the last page of this data sheet. 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

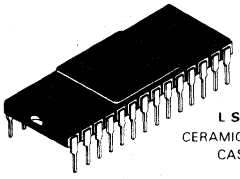
USER SELECTABLE OPTIONS

- 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

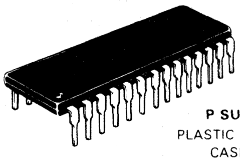
HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

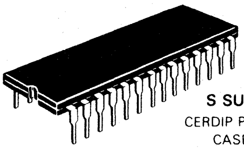
8-BIT
MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719

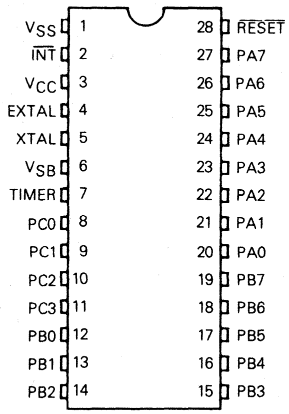


P SUFFIX
PLASTIC PACKAGE
CASE 710



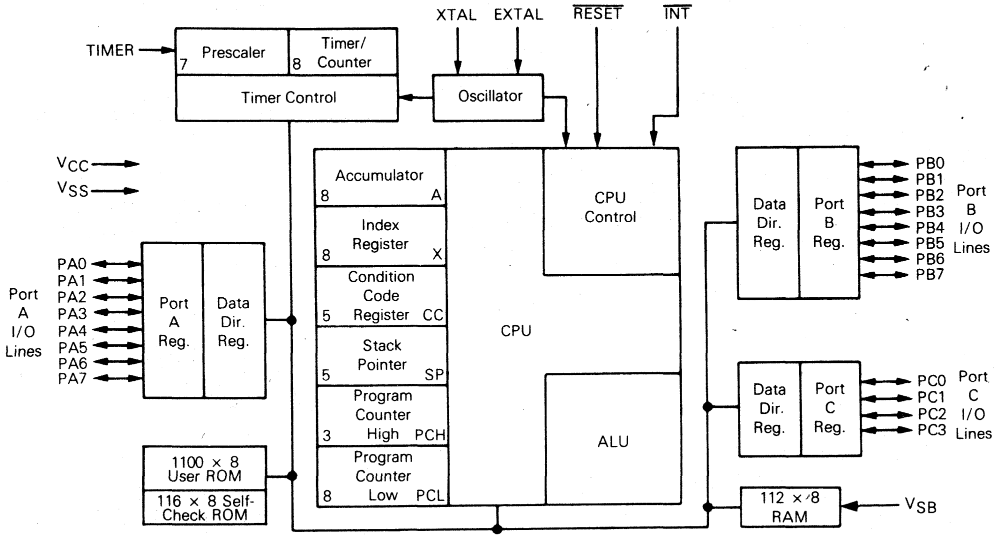
S SUFFIX
CERDIP PACKAGE
CASE 733

FIGURE 1 — PIN ASSIGNMENTS



ADI-865-R1

FIGURE 2 — 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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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}	120 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:

$$P_D = K + (T_J + 273°C) \quad (2)$$

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
$\overline{\text{INT}}$ and TIMER Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	–	100	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (±5° Accuracy)	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	V_{IH}				V
$\overline{\text{RESET}}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	–	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{\text{INT}}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	*	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer	V_{IH}				V
Timer Mode		2.0	–	V_{CC}	
Self-Check Mode		–	9.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{\text{RESET}}$		−0.3	–	0.8	
$\overline{\text{INT}}$		−0.3	*	1.5	
All Other		−0.3	–	0.8	
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figures 11, 12, and 13)					V
"Out of Reset"	V_{IRES+}	2.1	–	4.0	
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{\text{INT}}$ 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	C_{in}				pF
XTAL		–	25	–	
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	–	3.5	–	V
Input Current	I_{in}				μA
TIMER (V_{in} = 0.4 V)		–	–	20	
$\overline{\text{INT}}$ (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	
$\overline{\text{RESET}}$ (V_{in} = 0.8 V) (External Capacitor Charging Current)		−4.0	–	−50	

*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}	3.5	–	–	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}	0.3	–	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}	–0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	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}	–0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

STANDBY RAM CHARACTERISTICS (T_A = 0°C to 70°C)

Characteristic	Symbol	Min	Typ	Max	Unit
Standby Current 16 Bytes 64 Bytes 112 Bytes	I_{SB}	– – –	1.6 3.4 5.2	TBD TBD TBD	mA
RAM Standby Voltage	V_{SB}	3.0	5.25	5.75	V
V_{CC} Turn-off Rate	V_{CCTO}	–	–	1/100	V/μs

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

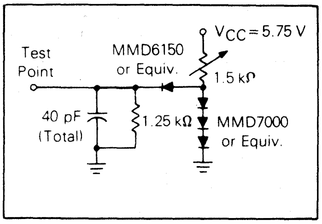


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

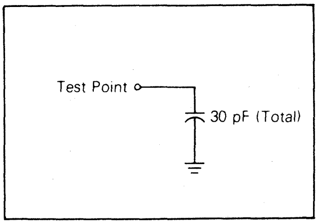
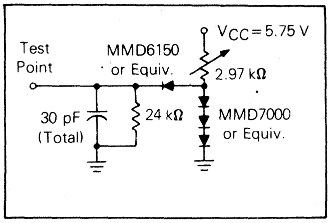


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$

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.

$\overline{\text{RESET}}$

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

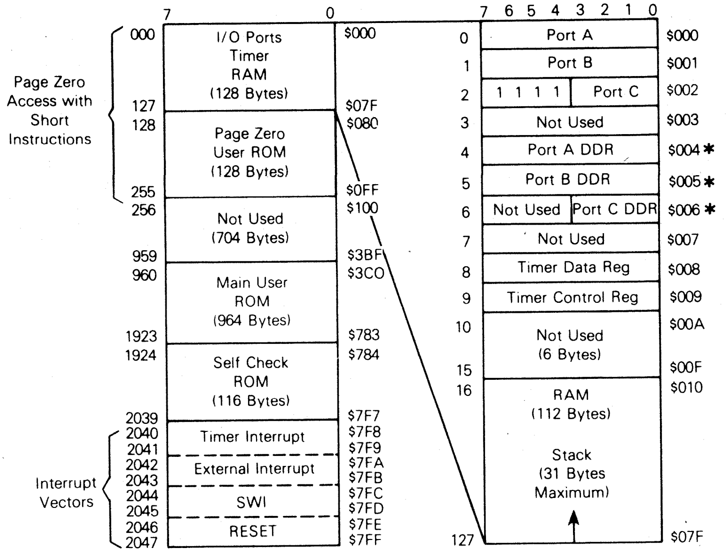
As shown in Figure 6, the MCU is capable of addressing 2048 bytes of memory and I/O registers with its program counter. The MC6805P4 MCU has implemented 1336 of these locations. This consists of: 1100 bytes of user ROM, 116 bytes of self-check ROM, 112 bytes of user RAM, 6 bytes of port I/O, and 2 timer 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 7. 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

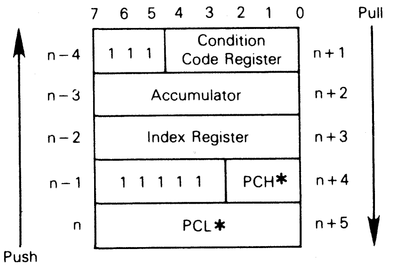
The CPU of the M6805 Family is implemented independently from the I/O or memory configuration. Consequently, it can be treated as an independent central processor communicating with I/O and memory via internal address, data, and control buses.

FIGURE 6 — MC6805P4 MCU ADDRESS MAP



*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 7 — INTERRUPT STACKING ORDER



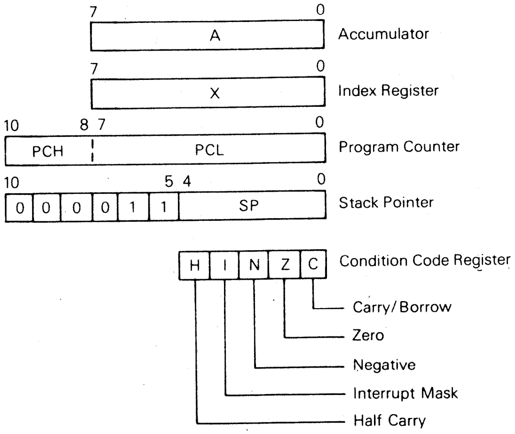
*For subroutine calls, only PCL and PCH are stacked.

REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 8 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.

FIGURE 8 — PROGRAMMING MODEL



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 ($\overline{\text{INT}}$). 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 9. 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 $\overline{\text{INT}}$ 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 9 — TIMER BLOCK DIAGRAM

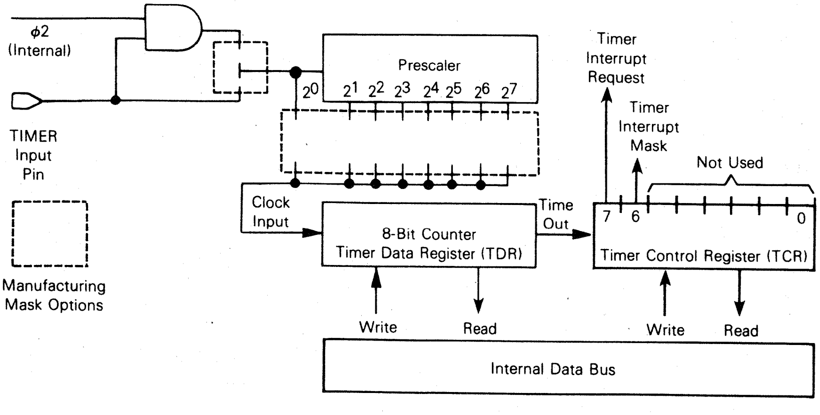
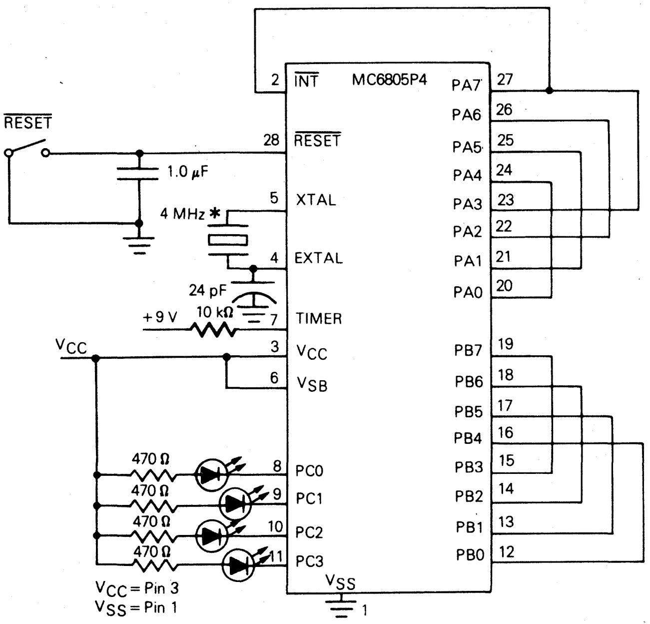


FIGURE 10 — SELF-CHECK CONNECTIONS



*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 $\phi2$ 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 $\phi2$ 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 10 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 ($\overline{\text{RESET}}$), and by an optional internal low voltage detect circuit; see Figure 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 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 12.

During 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 13, typically provides sufficient delay. See Figure 17 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 14. The crystal specifications and suggested PC board layouts 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 11 — POWER AND $\overline{\text{RESET}}$ TIMING

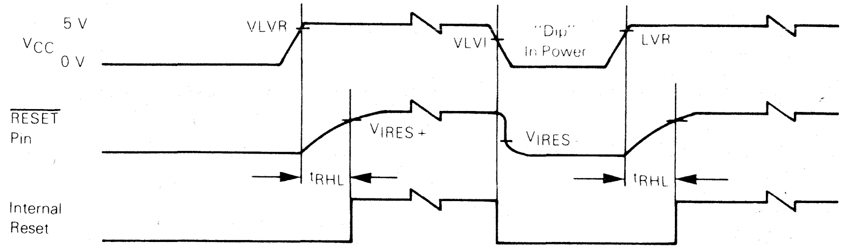


FIGURE 12 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

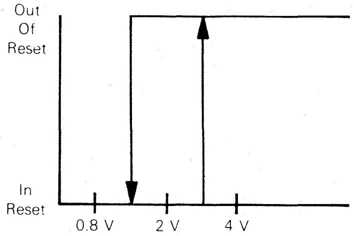


FIGURE 13 — POWER-UP RESET DELAY CIRCUIT

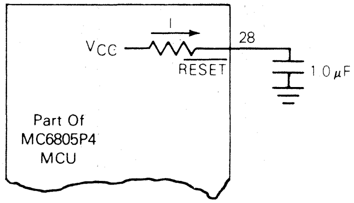
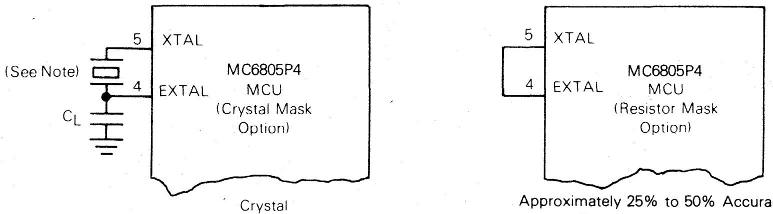
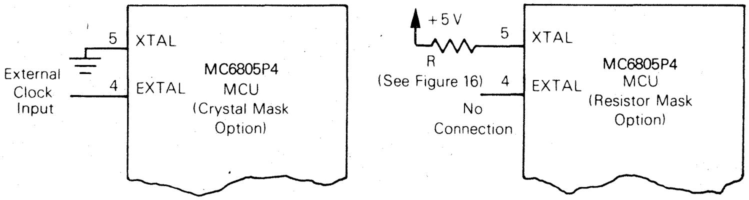


FIGURE 14 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% to 50% Accuracy
Typical $t_{cyc} = 1.25\ \mu s$
External Jumper

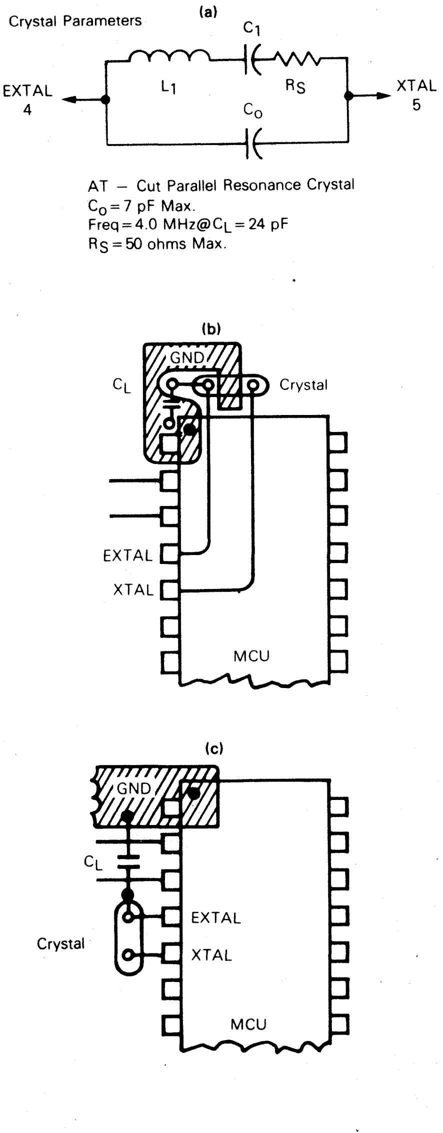


External Clock

Approximately 10% to 25% Accuracy
External Resistor
(Excludes Resistor Tolerance)

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.

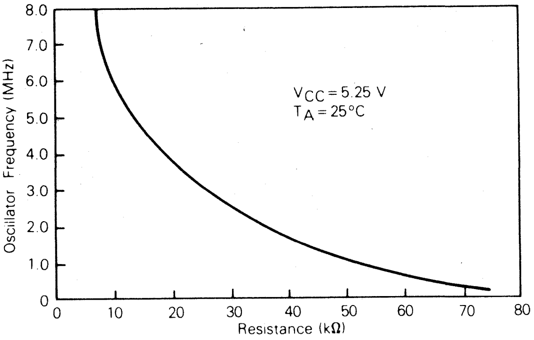
FIGURE 15 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



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 16 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P4 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 require a total of 11 t_{cyc} periods for completion.

A flowchart of the interrupt sequence is shown in Figure 17. 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 $\overline{\text{RESET}}$, 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 (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 18(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. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.

FIGURE 17 — RESET AND INTERRUPT PROCESSING FLOWCHART

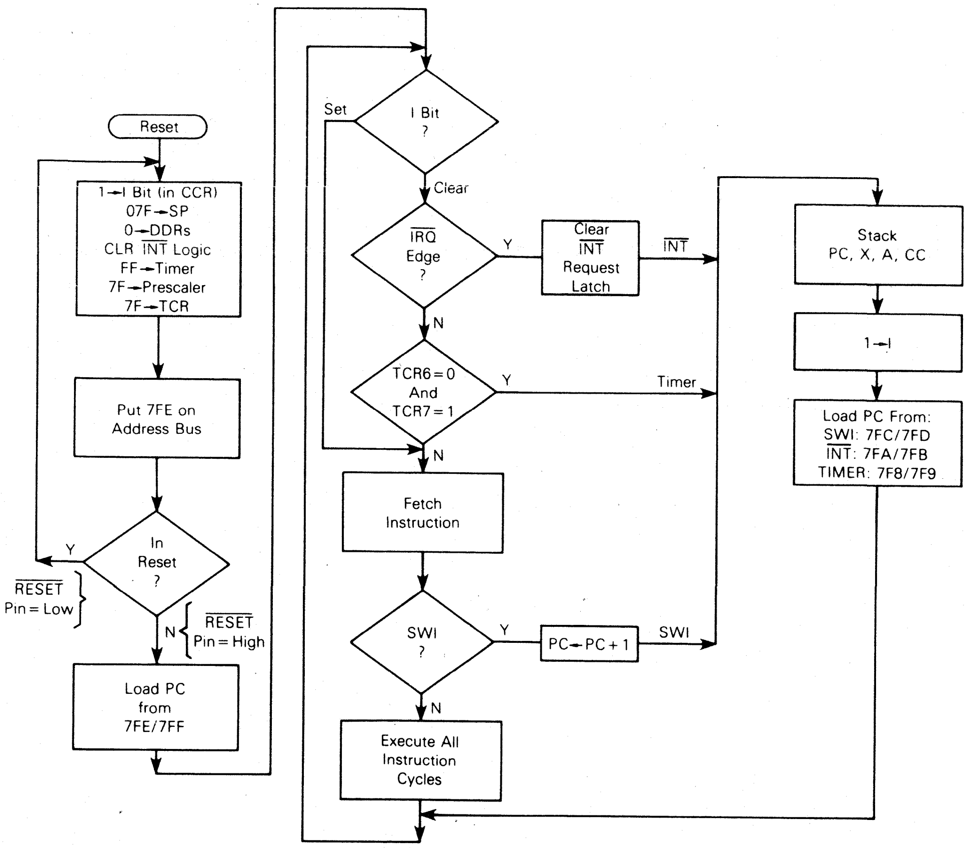
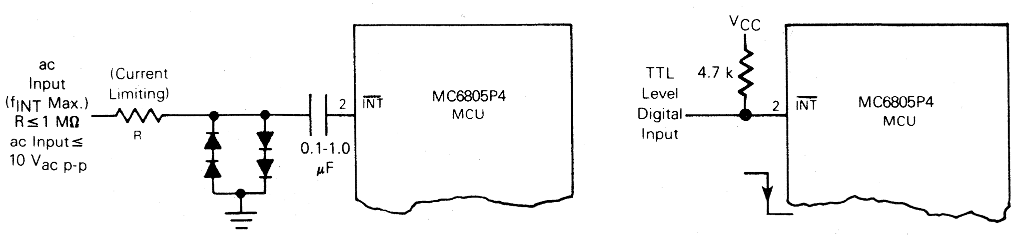


FIGURE 18 — TYPICAL INTERRUPT CIRCUITS



(a) Zero-Crossing Interrupt

(b) Digital-Signal Interrupt

For digital applications, the $\overline{\text{INT}}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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 18(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. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 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 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 19. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

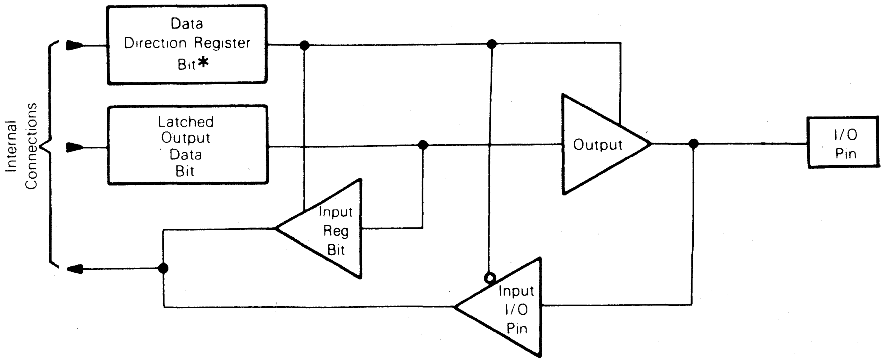
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 6 gives the address of data registers and DDRs. The register configuration is provided in Figure 20 and Figure 21 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 19) 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 19 — TYPICAL PORT I/O CIRCUITRY

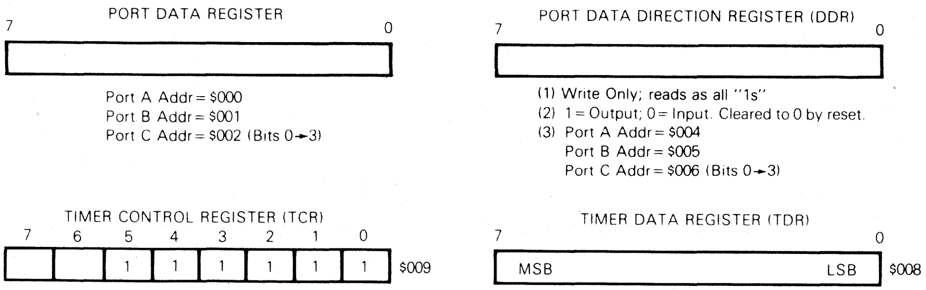


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

* 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 20 — MCU REGISTER CONFIGURATION



Port A Addr = $000
Port B Addr = $001
Port C Addr = $002 (Bits 0→3)

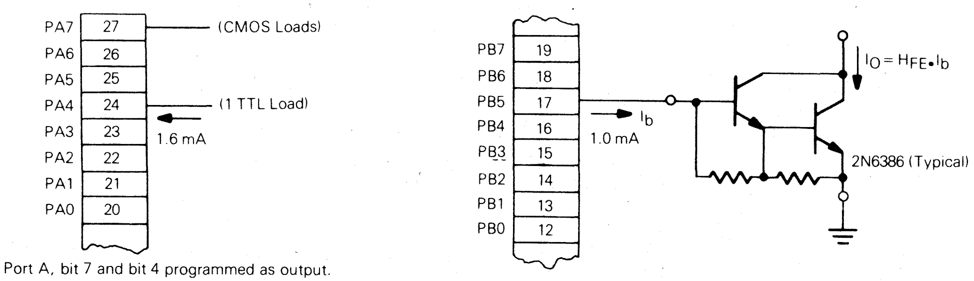
(1) Write Only; reads as all "1s"
(2) 1 = Output; 0 = Input. Cleared to 0 by reset.
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006 (Bits 0→3)

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.

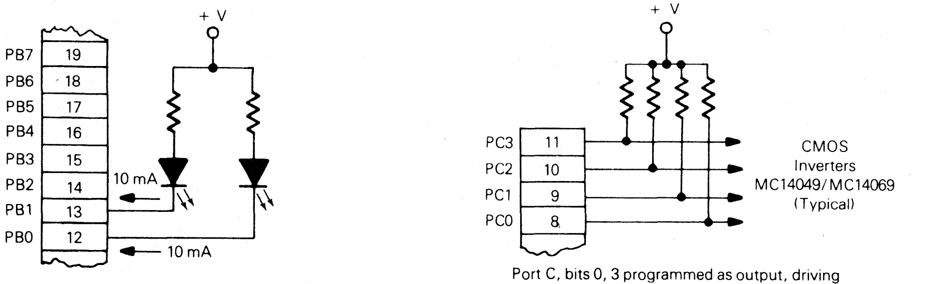
TCR Bits 5, 4, 3, 2, 1, 0 read as "1s" — unused bits.

FIGURE 21(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.

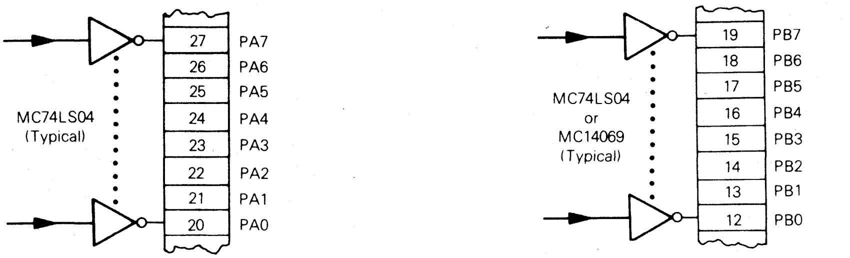
Port B, bit 5 programmed as output, driving Darlington-base directly.



Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.

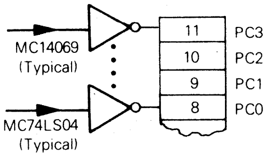
Port C, bits 0, 3 programmed as output, driving CMOS loads, using external pullup resistors.

FIGURE 21(b) — TYPICAL INPUT MODE PORT CONNECTIONS



TTL Driving Port A Directly

CMOS or TTL Driving Port B Directly



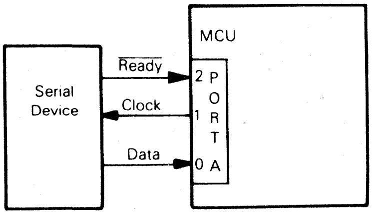
CMOS and TTL Driving Port C Directly

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 22 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 22 — BIT MANIPULATION EXAMPLE



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

		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 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										●	●	●	●	O
CLI	X										●	O	●	●	●
CLR	X		X			X	X				●	●	O	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				●	●	O	∧	∧
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)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero		

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)	C Carry/Borrow
I Interrupt Mask	Λ Test and Set if True, Cleared Otherwise
N Negative (Sign Bit)	● Not Affected
Z Zero	? Load CC Register From Stack

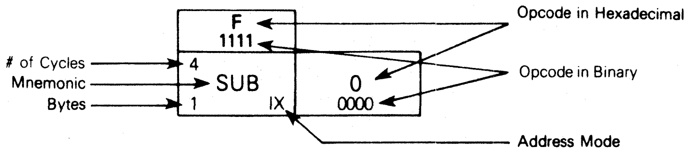
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

	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



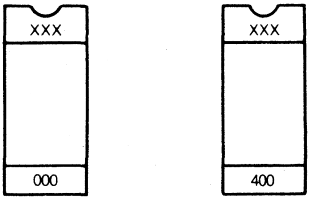
ORDERING INFORMATION

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.

MC6805P4 MCU 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 $\phi2$ 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)

Internal Oscillator Input
- ☐ Crystal
- ☐ Resistor

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Standby RAM
- ☐ 16 Bytes
- ☐ 64 Bytes
- ☐ 112 Bytes
- ☐ Other

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 ____________________

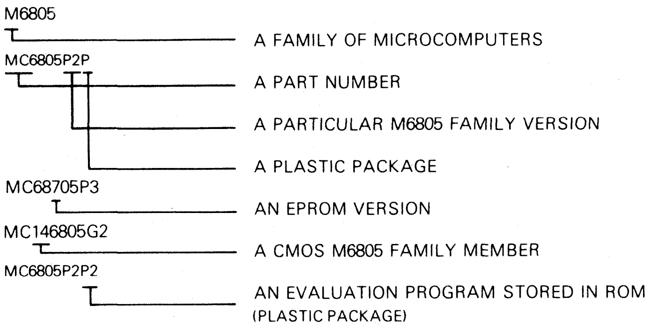
Signature ____________________

M6805 FAMILY

	MC6805P2	MC6805P4	MC6805R2	MC6805T2	MC6805U2
Technology	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	28	40
On-Chip RAM (Bytes)	64	112	64	64	64
On-Chip User ROM (Bytes)	1.1K	1.1K	2K	2.5K	2K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	19	24
Unidirectional I/O Lines	None	None	6 Inputs	None	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, PLL	Timer
External Interrupt Inputs	1	1	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705R3	None	MC68705U3
STOP and WAIT	No	No	No	No	No

	MC68705P3	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2
Technology	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS
Number of Pins	28	40	40	40	28	40
On-Chip RAM (Bytes)	112	112	112	112	64	112
On-Chip User ROM (Bytes)	1.8K EPROM	3.8K EPROM	3.8K EPROM	None	1K	2K
External Bus	None	None	None	Yes	None	None
Bidirectional I/O Lines	20	24	24	16	16	32
Unidirectional I/O Lines	None	6 Inputs	8 Inputs	None	4 Inputs	None
Other I/O Features	Timer	Timer, A/D	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	2	2	1	1	1
EPROM Version	–	–	–	None	None	None
STOP and WAIT	No	No	No	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE



MOTOROLA

MC6805R2

Advance Information

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT
MICROCOMPUTER
WITH A/D

8-BIT MICROCOMPUTER UNIT WITH A/D

The MC6805R2 Microcomputer Unit (MCU) is a member of the M6805 Family of low-cost single-chip Microcomputers. The 8-bit microcomputer contains a CPU, on-chip CLOCK, ROM, RAM, I/O, 4-channel 8-bit A/D, and TIMER. It is designed for the user who needs an economical microcomputer with the proven capabilities of the M6800-based instruction set. A comparison of the key features of several members of the M6805 Family of microcomputers is shown on the last page of this data sheet. The following are some of the hardware and software highlights of the MC6805R2 MCU.

HARDWARE FEATURES:

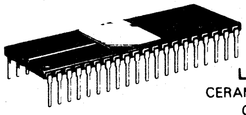
- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 2048 Bytes of User ROM
- 24 TTL/CMOS Compatible Bidirectional I/O Lines (8 lines are LED Compatible)
- A/D converter - up to 5 Input Lines
- A/D Converter
  - 8-Bit Conversion, Monotonic
  - 1 to 4 Multiplexed Analog Inputs
  - ± 1/2 LSB Quantizing Error
  - ± 1/2 LSB All Other Errors
  - ± 1 LSB Total Error (max)
  - Ratiometric Conversion
- Zero-Crossing Detection
- On-Chip Clock Generator
- Self-Check Mode
- Master Reset
- Complete Development System Support On EXORciser®
- 5 V Single Supply

SOFTWARE FEATURES:

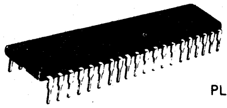
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Register/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes
- User Callable Self-Check Subroutines

USER SELECTABLE OPTIONS:

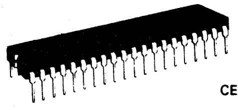
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits 2^N)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- 4 Vectored Interrupts; Timer, Software, and 2 External



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

FIGURE 1 — PIN ASSIGNMENTS

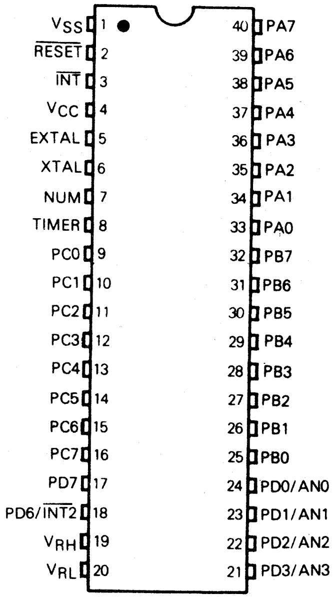
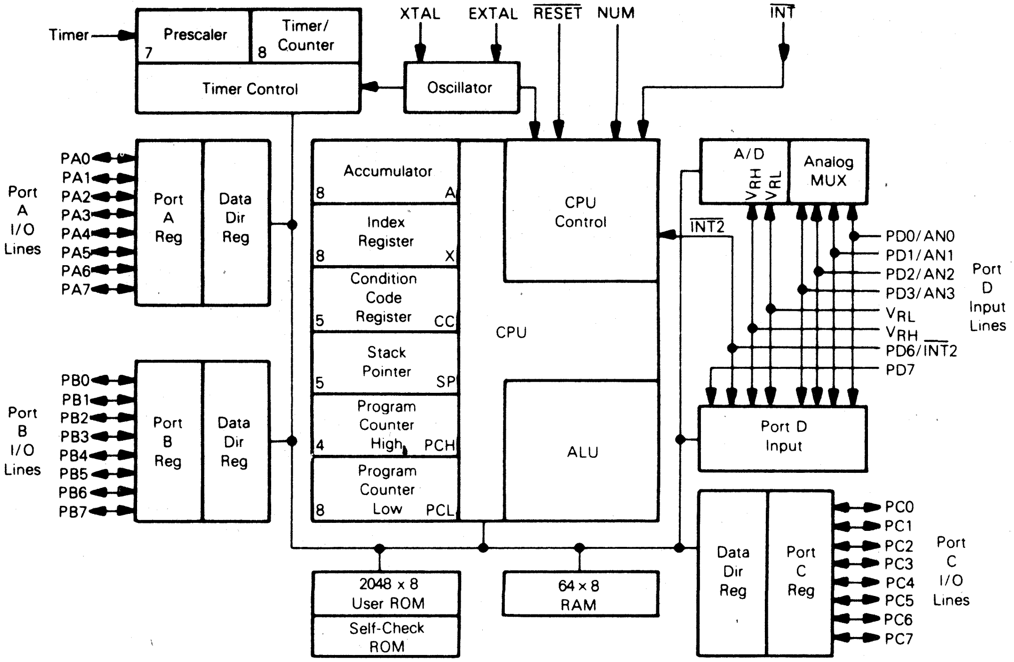


FIGURE 2 — MC6805R2 HMOS MICROCOMPUTER BLOCK DIAGRAM



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	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Junction Temperature	T_J		°C
Plastic Package		150	
Ceramic Package		175	
Cerdip		175	

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		100	
Ceramic		50	
Cerdip		60	

Stresses above those listed under "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. 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 = $P_{INT} + P_{PORT}$
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.

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	V_{IH}				V
$\overline{RESET}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	–	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{INT}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	*	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer	V_{IH}				V
Timer Mode		2.0	–	V_{CC}	
Self-Check Mode		–	9.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{RESET}$		−0.3	–	0.8	
$\overline{INT}$		−0.3	*	1.5	
All Other (Except A/D Inputs)		−0.3	–	0.8	
$\overline{RESET}$ Hysteresis Voltages (See Figures 11, 12, and 13)					V
"Out of Reset"	V_{IRES+}	2.1	–	4.0	
"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	–	600	–	mW
Input Capacitance	C_{in}				pF
EXTAL		–	25	–	
All Other Except Analog Inputs		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVR}	–	3.5	–	V
Input Current	I_{in}				μA
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)		–	–	10	
(V_{in} = 0.4 V Crystal Option)		–	–	−1600	
$\overline{RESET}$ (V_{in} = 0.8 V) (External Capacitor Charging Current)		−4.0	–	−50	

NOTE: Port D Analog Inputs, when selected, C_{in} = 25 pF for the first 5 out or 30 cycles.
*Due to internal biasing this input (when unused) floats to approximately 2.0 V.

A/D CONVERTER CHARACTERISTICS (V_{CC} = +5.25 Vdc ±0.5 Vdc, V_{SS} = GND, 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
Quantizing Error	–	–	±1/2	LSB	For V_{RH} = 4.0 to 5.0 V and V_{RL} = 0 V
Conversion Range	V_{RL}	–	V_{RH}	V	
V_{RH}	–	–	5.0	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 5.0 V
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	FF	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 V_{RL} are not allowed at any time during conversion

SWITCHING 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
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time (4/f_{osc})	t_{cyc}	0.95	–	10	μs
$\overline{INT}$, $\overline{INT}2$, 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 (for ±5° Accuracy)	f_{INT}	0.03	–	1	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}	3.5	–	–	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}	−0.3	–	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}	−0.3	–	0.8	V
Hi-Z-State Input Current	I_{TSI}	–	2	20	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}	−0.3	–	0.8	V
Hi-Z-State Input Current	I_{TSI}	–	2	20	μA
Port D (Digital Inputs Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input Current*	I_{in}	–	–	20	μA

*V_{RL}/P_D4 – V_{RH}/P_D5: The A/D conversion resistor (11.6 k nominal) is connected internally between these two lines, impacting their use as digital inputs in some applications.

FIGURE 3 – TTL EQUIVALENT TEST LOAD (PORT B)

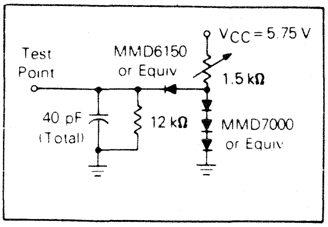


FIGURE 4 – CMOS EQUIVALENT TEST LOAD (PORT A)

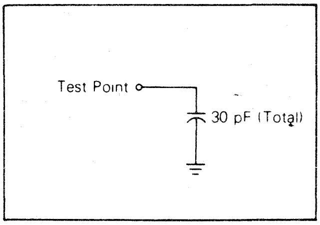
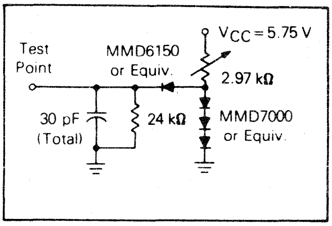


FIGURE 5 – TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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 additonal information.

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 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 INTERNAL CLOCK GENERATOR OPTIONS for recommendations about these inputs.

TIMER – The pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER 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. The MCU can be reset by pulling $\overline{RESET}$ low. Refer to RESETS for additional information.

NUM (Non-User Mode) – This pin is not for user application and must be connected to V_{SS}.

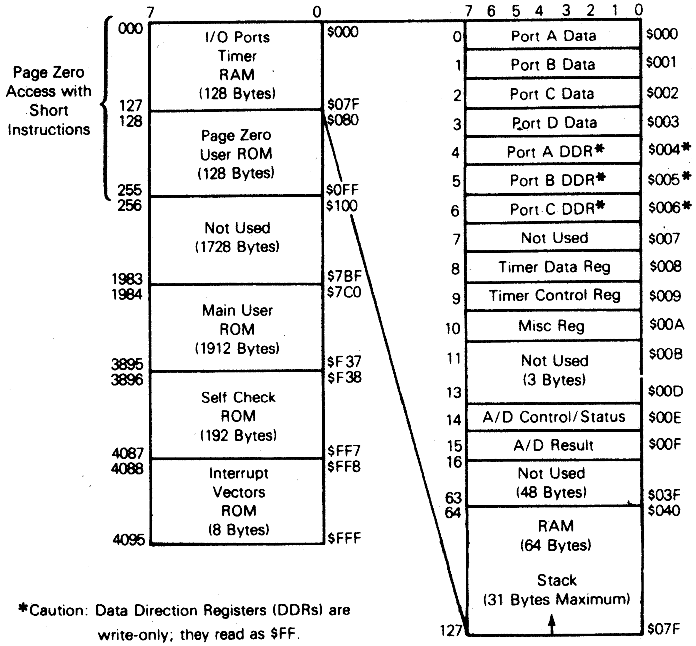
INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) – These 30 lines are arranged into three 8-bit ports (A, B, and C) plus port D with 6 inputs. Ports A, B, and C are programmable as either inputs or outputs, under software control of the data direction registers. Port D has from one to four analog inputs, an $\overline{INT2}$ input, and from one to five digital inputs. All port D lines can also be directly read and used as binary inputs. The voltage reference pins (V_{RH} and V_{RL}) for the A/D converter are also read as a part of port D. Refer to INPUT/OUTPUT, A/D CONVERTER, and INTERRUPTS for additional information.

MEMORY – The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The MC6805R2 MCU has implemented 2316 of these bytes. This consists of: 2048 user ROM bytes, 192 self-check ROM bytes, 64 user RAM bytes, 7 port I/O bytes, 2 timer registers, 2 A/D registers and a miscellaneous register; see Figure 6 for the Address Map. 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 \$7C0 to \$F37. The last 8 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 12 have been implemented. These locations are used for the ports, the port DDRs, the timer, the $\overline{INT2}$ miscellaneous register, and the A/D. Of the 64 RAM bytes, 31 bytes are shared with the stack area. The stack must be used with care when data shares the stack area.

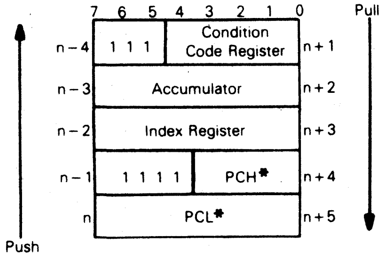
The shared stack area is used during the processing of an interrupt or subroutine calls to save the contents of the CPU state. The register contents are pushed onto the stack in order shown in Figure 7. Since the Stack pointer decrements during pushes, the low order byte (PCL) of the Program Counter 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 bing pushed onto the stack; the remaining CPU registers are not pushed.

FIGURE 6 — MC6805R2 MCU ADDRESS MAP



*Caution: Data Direction Registers (DDRs) are write-only; they read as $FF.

FIGURE 7 — 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.

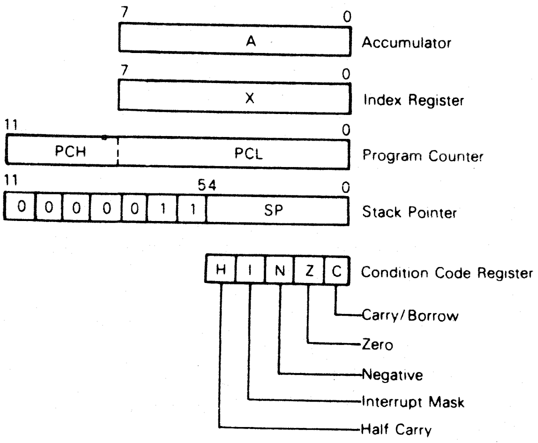
REGISTERS

The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 8 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.

FIGURE 8 – PROGRAMMING MODEL



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. 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 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).

CONDITION CODE REGISTERS (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.

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 interrupts ($\overline{\text{INT}}$ and $\overline{\text{INT}}$2) 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 one).

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 MC6805R2 MCU timer circuitry is shown in Figure 9. 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 in the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine (see the INTERRUPT section). THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE. The timer and $\overline{\text{INT2}}$ share the same interrupt vector. THE INTERRUPT ROUTINE MUST CHECK THE REQUEST BITS TO DETERMINE THE SOURCE OF THE INTERRUPT.

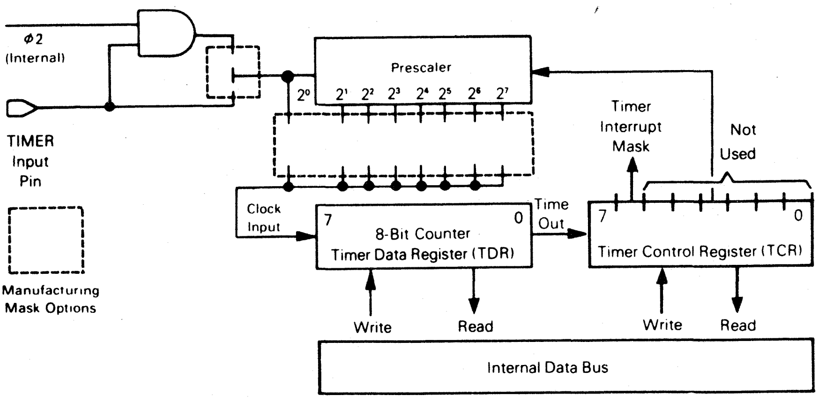
The clock input to the timer can be from an external source (decrementing of Timer Counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal $\phi 2$ signal. When the $\phi 2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE: For ungated $\phi 2$ clock input 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 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 Timer Control Register is written to a logic 1 (this bit always reads as a logic 0).

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 request mask bit (bit 6) is set.

FIGURE 9 — TIMER BLOCK DIAGRAM



SELF-CHECK

The self-check capability of the MC6805R2 MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 10 and monitor the output of Port C bit 3 for an oscillation of approximaty 7 Hz. A 9 volt level on the TIMER input, pin 8, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, timer, A/D, interrupts, and I/O ports.

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 $\phi 2$ clock.

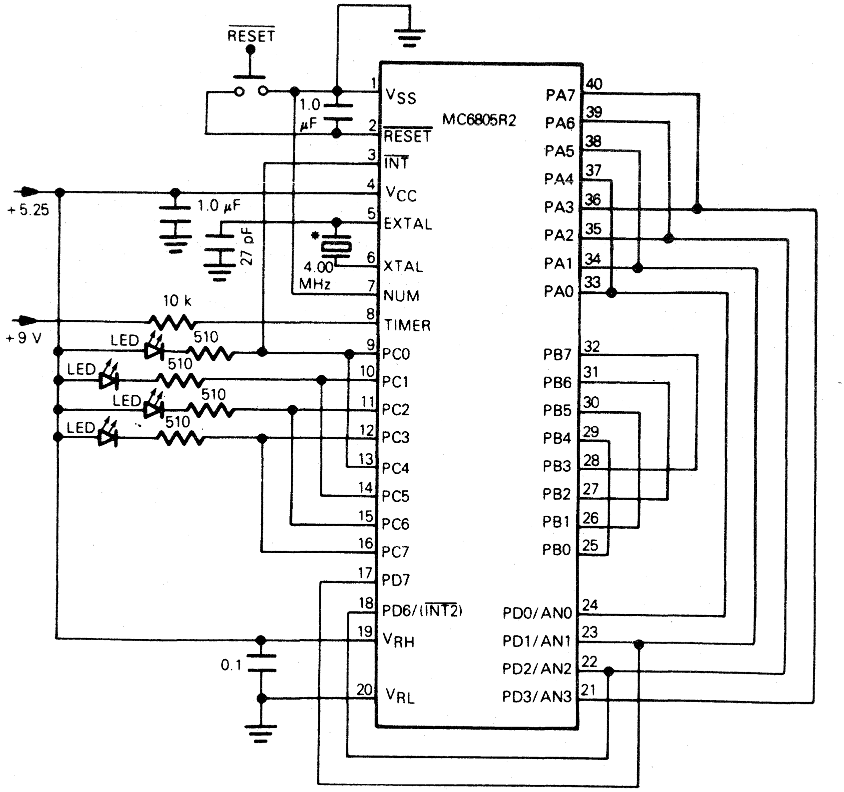
RAM SELF-CHECK SUBROUTINE — The RAM self-check is called at location $F6F and returns with the Z-bit clear if any error is detected; otherwise the Z-bit is set. The walking diagnostic pattern method is used.

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.

ROM CHECKSUM SUBROUTINE — The ROM self-check is called at location $F8A and returns with the 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 $040-$043 are overwritten.

FIGURE 10 — SELF-CHECK CONNECTIONS



*This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected.

LED Meanings

C0	C1	C2	C3	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 Part

Anything else bad Part, Bad Port 3, Bad ISP, etc.

A-TO-D CONVERTER SELF-CHECK — The A/D self-check is called at location $FA4 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 4 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.

TIMER SELF-CHECK SUBROUTINE — The timer self-check is called at location $FCF 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 $\phi 2$ clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask 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.

RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{RESET}$), and by an t_{cyc} internal low voltage detect circuit; see Figure 11. 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 logical 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 12..

Upon power-up, a delay of T_{RHL} milliseconds 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 13 typically provides sufficient delay.

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 14. The Crystal specifications are given in Figure 15. A resistor selection graph is shown in Figure 16.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially Rs) oscillator load capacitances, IC parameters, ambient temperatures, supply voltage, and supply voltage turn-on time. To ensure rapid oscillator start-up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

FIGURE 11 — POWER AND RESET TIMING

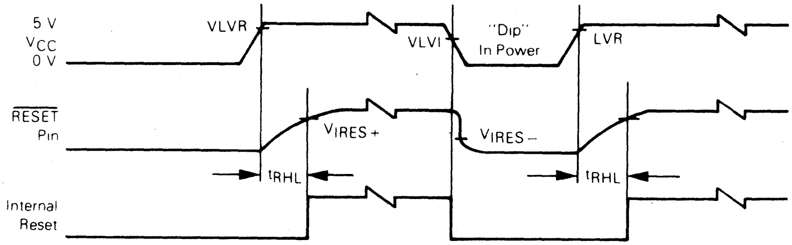


FIGURE 12 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

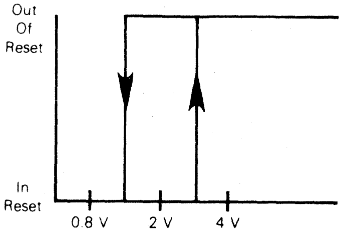


FIGURE 13 — POWER UP RESET DELAY CIRCUIT

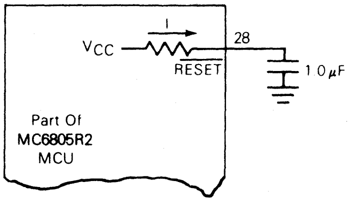
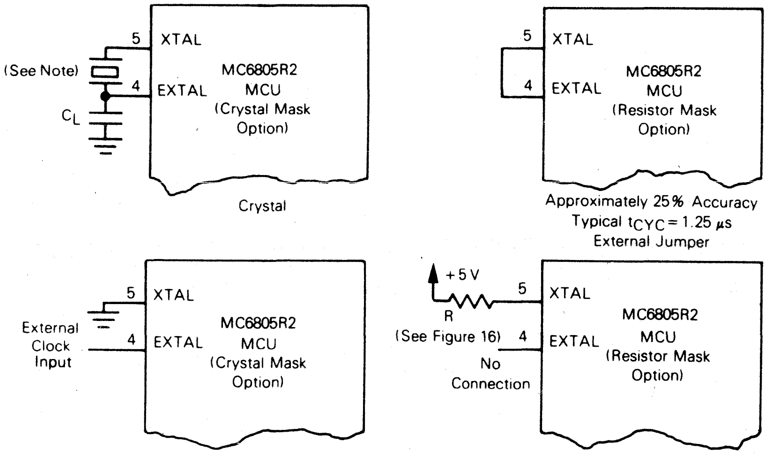


FIGURE 14 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% Accuracy
Typical $t_{CYC} = 1.25\ \mu s$
External Jumper

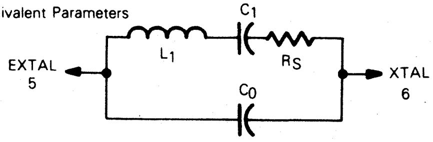
External Clock

Approximately 10% Accuracy
External Resistor
(Excludes Resistor Tolerance)

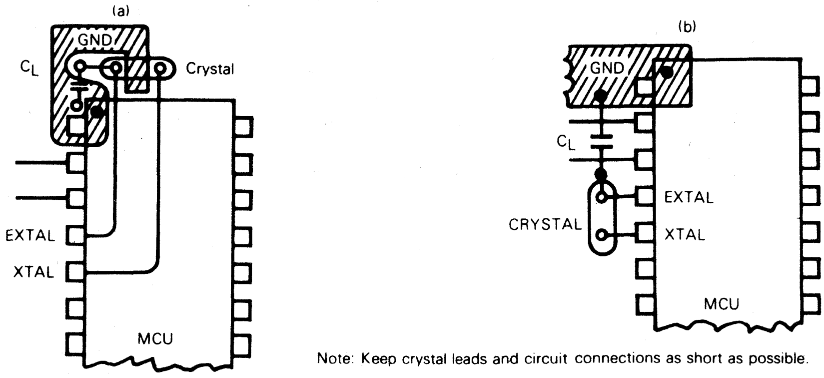
NOTE: The recommended C_L value with a 4.0 MHz crytal is 27 pF, maximum, inlcuding system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the 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 15 — CRYSTAL MOTIONAL ARM PARAMETERS
AND SUGGESTED PC BOARD LAYOUT

Crystal Motional Arm
Equivalent Parameters

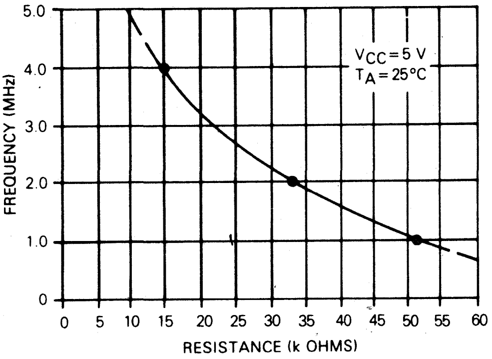


AT — Cut Parallel Resonance Crystal
$C_0 = 7$ pF Max
FREQ = 4.0 MHz @ $C_L = 24$ pF
$R_S = 50$ ohms Max.



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

FIGURE 16 – TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION

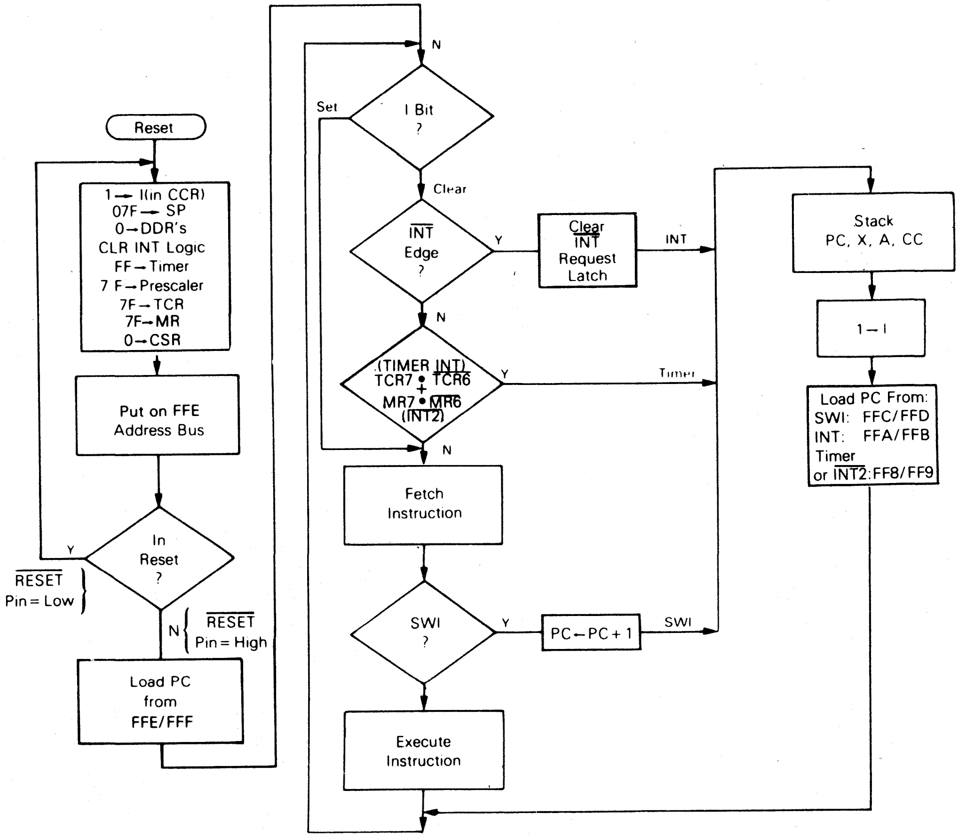


INTERRUPTS

The MC6805R2 MCU 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, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU 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.

Refer to Figure 17 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 1 provides a listing of the interrupts, their priority, and the address of the vector which

FIGURE 17 – $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART



contains the starting address of the appropriate interrupt service routine. The interrupt priority applies to those pending when the CPU is ready to accept a new interrupt. $\overline{\text{RESET}}$ is listed in Table 1 because it is processed similar to an interrupt. However, it is not normally used as an interrupt. When the interrupt mask bit in the Condition Code Register is set the interrupt is latched for later interrupt execution.

NOTE

> The timer and $\overline{\text{INT2}}$ share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR7 and MR7). Both TCR7 and MR7 can only be written to 0 by software.

The external interrupts, $\overline{\text{INT}}$ and $\overline{\text{INT2}}$, are set on the falling edge of the input signal. The $\overline{\text{INT2}}$ has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the Miscellaneous Register (MR); refer to Figure 18. The $\overline{\text{INT2}}$ interrupt is inhabited when the mask bit is set. The $\overline{\text{INT2}}$ is always readable as a digital input of Port D. The $\overline{\text{INT2}}$ and timer interrupt request bits, if set, causes the MCU to process an interrupt when the condition code I-bit is clear.

TABLE 1 — INTERRUPT PRIORITIES

Interrupt	Priority	Vector Address
$\overline{\text{RESET}}$	1	$FFE and $FFF
SWI	2*	$FFC and $FFD
$\overline{\text{INT}}$	3	$FFA and $FFB
TIMER/$\overline{\text{INT2}}$	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 4, like any other instruction, the priority of $\overline{\text{INT}}$ thus becomes 2 and the timer becomes 3.

FIGURE 18 — MCU REGISTER CONFIGURATION

PORT DATA DIRECTION REGISTER (DDR)

7	0

(1) Write Only; reads as all 1's
(2) 1 = Output, 0 = Input. Cleared to 0 by Reset.
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

PORT DATA REGISTER

7	0

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0	
		1	1		1	1	1	$009

TCR7 — Timer Interrupt Request Status Bit: Set when TDR goes to zero; must be cleared by software. Cleared to 0 by Reset.
TCR6 — Timer Interrupt Mask Bit: 1 = timer interrupt masked (disabled). Set to 1 by Reset.
TCR3 — Clear prescaler: always reads as a 0; clears prescaler when written to a logic 1.
TCR Bits 5, 4, 2, 1, 0 reads 1's — unused bits.

TIMER DATA REGISTER (TDR)

7	0	
MSB	LSB	$008

MISCELLANEOUS REGISTER (MR)

7	6						0	
		1	1	1	1	1	1	$00A

MR7 Bit 7 — INT2 Interrupt Request Bit: Set when falling edge detected on INT2 pin, must be cleared by software. Cleared to 0 by Reset.
MR6 Bit 6 — INT2 Interrupt Mask Bit: 1 = INT2 Interrupt masked (disabled). Set to 1 by Reset.
MR Bits 5, 4, 3, 2, 1, 0 — Read as 1's — unused bits.

A/D CONTROL REGISTER (ACR)

7	6	5	4	3	2	1	0	
	1	1	1	1				$00E

Bit 7 — Conversion Complete Status Flag: Set when conversion is complete; Cleared only on a write to ACR.

Readable, not writable.

Bits 2, 1, 0 — A/D input Mux Selection (See Table 2).
Bits 6, 5, 4, 3 read as 1's — unused bits.

A/D RESULT REGISTER (ARR)

7	0	
MSB	LSB	$00F

A sinusoidal input signal (f_{INT} maximum) can be used to operate an external interrupt ($\overline{INT}$), as shown in Figure 19, 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. For digital applications, the $\overline{INT}$ can be driven directly by a digital signal at a maximum period of t_{IWL}. 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 provide a 2f clock.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWI's are usually used as breakpoints for debugging or as system calls.

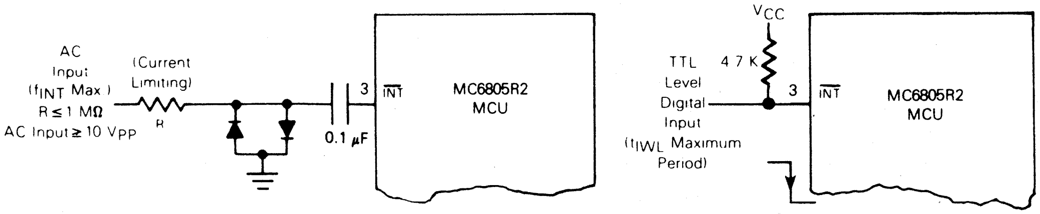
INPUT/OUTPUT

There are 30 input or input/output pins. The $\overline{INT}$ 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 "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 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 20.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs using a mask option. 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 1.0 milliampere on each pin.

Port D provides the multiplexed analog inputs, reference voltages, and $\overline{INT2}$. 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. The V_{RL} to V_{RH} lines (PD4 and PD5) are internally connected by the A/D resistor. Analog inputs may be pre-scaled to attain the V_{RL} to V_{RH} recommended input voltage range.

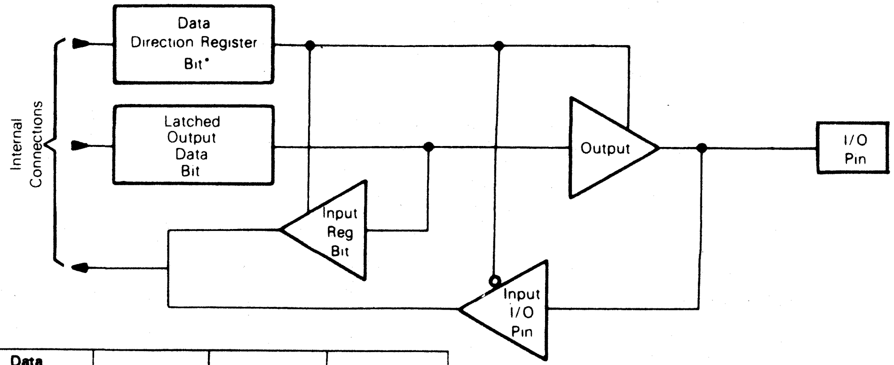
FIGURE 19 — TYPICAL INTERRUPT CIRCUITS



a — Zero Crossing Interrupt

b — Digital Signal Interrupt

FIGURE 20 — 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 1's.

**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.

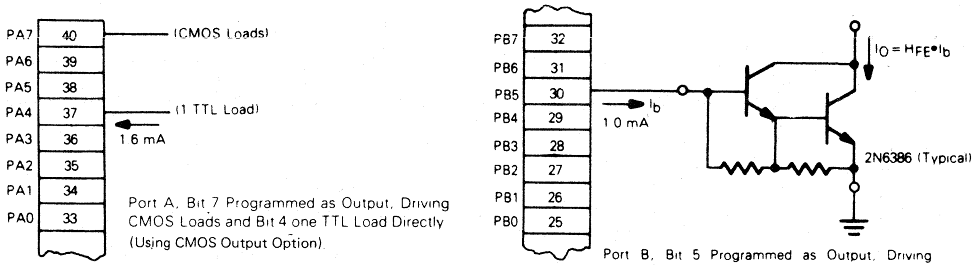
Figure 21 provides some examples of port connections. The Address Map in Figure 6 gives the addresses of data registers and DDRs. The Register Configuration is provided in Figure 18.

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 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.

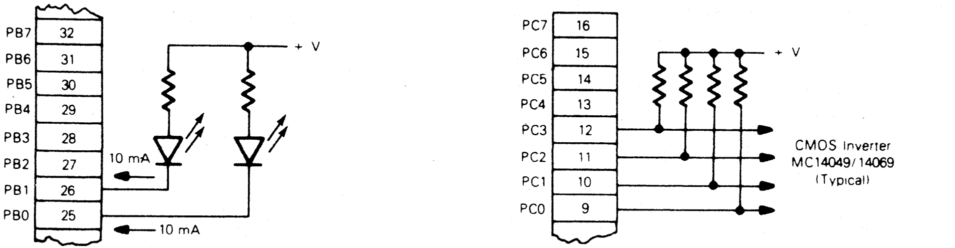
FIGURE 21 — TYPICAL PORT CONNECTIONS

a. Output Modes



Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 one TTL Load Directly (Using CMOS Output Option)

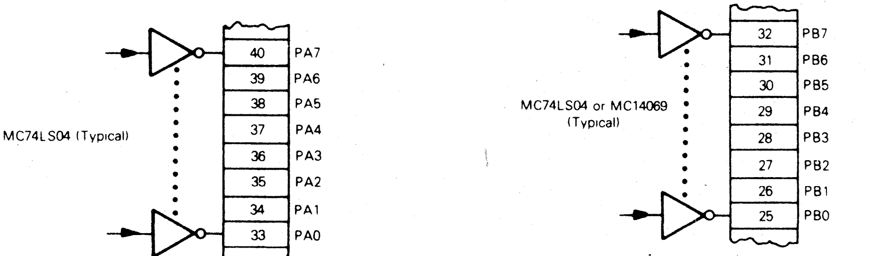
Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly



Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly

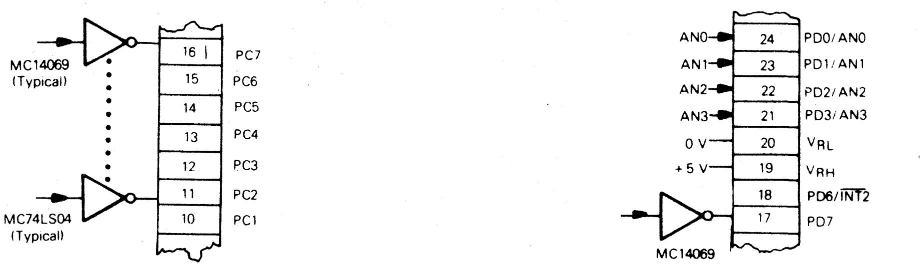
Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors

b. Input Modes



TTL Driving Port A Directly

CMOS or TTL Driving Port B Directly



CMOS and TTL Driving Port C Directly

Port D used as 4-Channel A/D Input with Bit 7 used as CMOS Digital Input.

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).

ANALOG-TO-DIGITAL CONVERTER (A/D) — The MC6805R2 MCU has an 8-bit A/D converter implemented on the chip using a successive approximation technique, as shown in Figure 22. Up to four external analog inputs, via port D, are connected to the A/D through a multiplexer. Four internal analog signals may be selected for calibration purposes (V_{RH}, $V_{RH}/2$, $V_{RH}/4$, V_{RL}).

The multiplexer selection is controlled by the A/D Control Register (ACR) bits 0, 1, and 2, see Table 2. This register is cleared during any Reset condition. Refer to Figure 18 for Register Configuration.

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 and held internally.

The converter operates continously using 30 machine cycles to complete a conversion of the sampled analog input. When conversion is complete, the digitized sample or 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 voltage (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 $00. An input voltage greater than V_{RH} converts to FF_{16} and no overflow indication is provided. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

CAUTION

This device contains circuitry to protect the inputs against damage due to high static voltages or electric field; however, the design of the input circuitry for the analog reference voltage pins (19 and 20) does not offer the SAME level of protection. Precautions should be taken to avoid applications of any voltage higher than maximum-rated voltage or handled in any environment producing high-static voltages.

FIGURE 22 — A/D BLOCK DIAGRAM

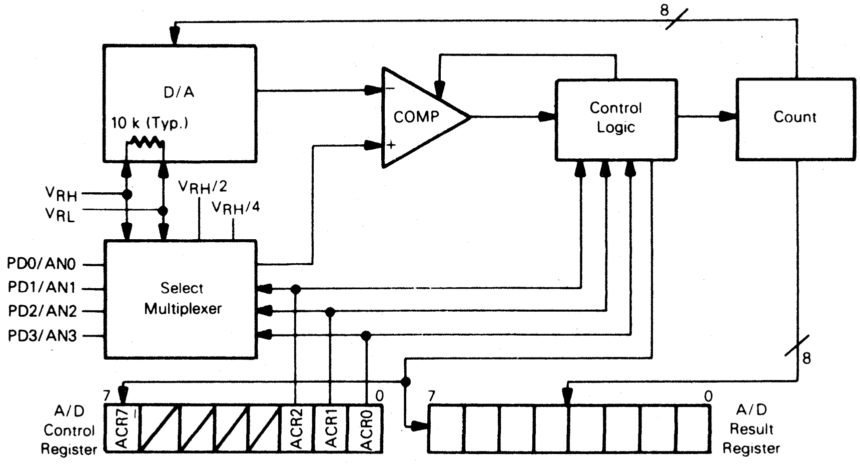


TABLE 2 — A/D INPUT MUX SELECTION

A/D Control Register			Input Selected
CR2	CR1	CR0	
0	0	0	AN0
0	0	1	AN1
0	1	0	AN2
0	1	1	AN3
1	0	0	V_{RH}*
1	0	1	V_{RL}*
1	1	0	$V_{RH}/4$*
1	1	1	$V_{RH}/2$*

*Internal (Calibration) levels

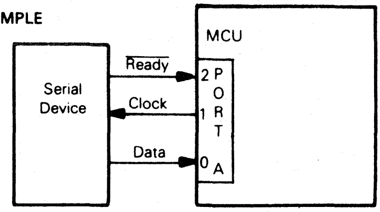
BIT MANIPULATION

The MC6805R2 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 paragraph) 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 work 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 23 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, LBS 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 23 — BIT MANIPULATION EXAMPLE

```
        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
        ASR     RAMLOC              MOVE C-BIT INTO RAM
```



ADDRESSING MODES

The MC6805R2 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 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 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 with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-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.

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 +129 to −126 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.

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 511 ($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 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 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 condition (set or clear) 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 clear in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 location of memory. The span of branching is from +130 to −125 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 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 instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC6805R2 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 paragraph. 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 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 — The 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 test and branch operations. Refer to Table 6.

CONTROL INSTRUCTION — 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 — Table 9 is an opcode map for the instruction used on the MCU.

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			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											●	●	●	●	●
BMS											●	●	●	●	●
BNE											●	●	●	●	●
BPL											●	●	●	●	●
BRA											●	●	●	●	●
BRN											●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLL	X										●	●	●	●	O
CLI	X										●	O	●	●	●
CLR	X		X			X	X				●	●	O	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				●	●	O	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										●	●	●	●	●

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

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

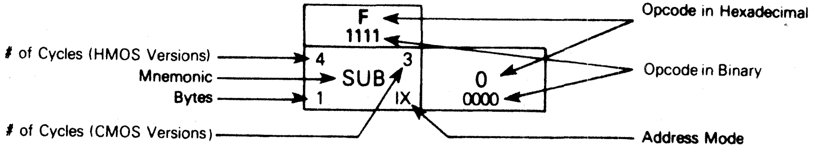
TABLE 8 — M6805 FAMILY OPCODE MAP

Hi / Low	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						Hi / Low
	BTB	BSC	REL	DIR	INH(A)	INH(X)	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	10 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEGA 1 INH	4 3 NEGX 1 INH	7 6 NEG 2 IX1	6 5 NEG 1 IX	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
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL						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 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL								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
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COMA 1 INH	4 3 COMX 1 INH	7 6 COM 2 IX1	6 5 COM 1 IX	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
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSRA 1 INH	4 3 LSRX 1 INH	7 6 LSR 2 IX1	6 5 LSR 1 IX			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
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL								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
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 RORA 1 INH	4 3 RORX 1 INH	7 6 ROR 2 IX1	6 5 ROR 1 IX			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
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASRA 1 INH	4 3 ASRX 1 INH	7 6 ASR 2 IX1	6 5 ASR 1 IX		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
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSLA 1 INH	4 3 LSLX 1 INH	7 6 LSL 2 IX1	6 5 LSL 1 IX		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
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROLA 1 INH	4 3 ROLX 1 INH	7 6 ROL 2 IX1	6 5 ROL 1 IX		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
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DECA 1 INH	4 3 DECX 1 INH	7 6 DEC 2 IX1	6 5 DEC 1 IX		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
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL							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
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INCA 1 INH	4 3 INCX 1 INH	7 6 INC 2 IX1	6 5 INC 1 IX		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
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TSTA 1 INH	4 3 TSTX 1 INH	7 5 TST 2 IX1	6 4 TST 1 IX		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
E 1110	10 5 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL						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
F 1111	10 5 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLRA 1 INH	4 3 CLRX 1 INH	7 6 CLR 2 IX1	6 5 CLR 1 IX	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

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



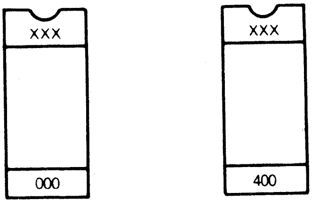
*CMOS Versions only

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 you 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 styroforam.

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 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 .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.

MC6805R2 MCU 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 $\phi2$ 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 Oscillator Input
- ☐ Crystal
- ☐ Resistor

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

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* ☐ −40° to +125°C*

*Requires prior factory approval

Marking Information (12 Characters Maximum)

Title ____________________

Signature ____________________

MOTOROLA

MC6805R3

Product Preview

8-BIT MICROCOMPUTER UNIT WITH A/D

The MC6805R3 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, A/D converter, and TIMER.

Since the MC6805R3 is supported by a pin compatible EPROM MCU, the MC68705R3, this offers the user an economical means of designing the MC6805R3 into his system. Prototype evaluation and pilot production can be supported by the fully compatible MC68705R3 EPROM MCU before committing high volume production to the MC6805R3 mask ROM MCU.

HARDWARE FEATURES

- A/D Converter
  - 8-Bit Conversion, Monotonic
  - Four Multiplexed Analog Inputs
  - ± ½ LSB Quantizing Error
  - ± 2 LSB All Other Errors
  - ± 2½ LSB Total Error (Maximum)
  - Ratiometric Conversion
- 32 TTL/CMOS Compatible I/O Lines
  - 24 Bidirectional (Eight Lines are LED Compatible)
  - Eight Input Only
- 3776 Bytes of User ROM
- 112 Bytes of RAM
- Self-Check Mode
- Zero-Crossing Detect/Interrupt
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler and Clock Source
- 5 Volt Single Supply

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

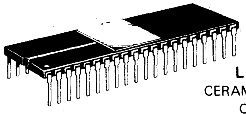
USER SELECTABLE OPTIONS

- Eight Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- Eight Bidirectional I/O Lines with TTL or Open-Drain Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vector Interrupts: Timer, Software, and External

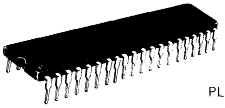
EXORciser is a registered trademark of Motorola Inc.

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

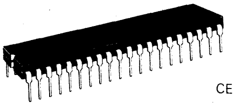
**8-BIT
MICROCOMPUTER
WITH A/D**



L SUFFIX
CERAMIC PACKAGE
CASE 715

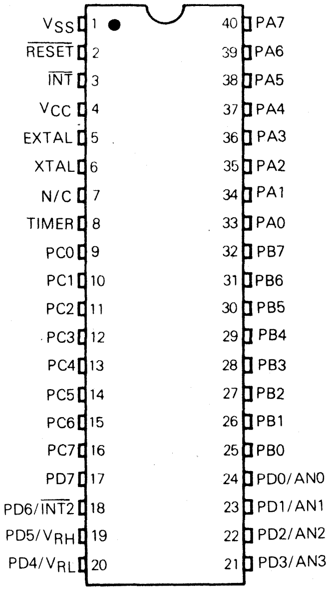


P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

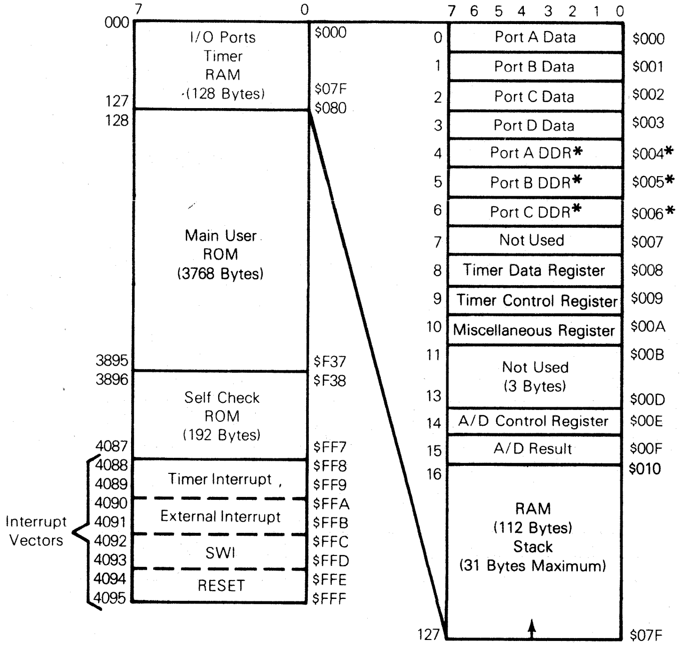
PIN ASSIGNMENTS



V_{SS}	1	40	PA7
$\overline{\text{RESET}}$	2	39	PA6
$\overline{\text{INT}}$	3	38	PA5
V_{CC}	4	37	PA4
EXTAL	5	36	PA3
XTAL	6	35	PA2
N/C	7	34	PA1
TIMER	8	33	PA0
PC0	9	32	PB7
PC1	10	31	PB6
PC2	11	30	PB5
PC3	12	29	PB4
PC4	13	28	PB3
PC5	14	27	PB2
PC6	15	26	PB1
PC7	16	25	PB0
PD7	17	24	PD0/AN0
PD6/$\overline{\text{INT2}}$	18	23	PD1/AN1
PD5/V_{RH}	19	22	PD2/AN2
PD4/V_{RL}	20	21	PD3/AN3

NP-362

MC6805R3 ADDRESS MAP



*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

ANALOG-TO-DIGITAL CONVERTER

The MC6805R3 microcomputer has an 8-bit analog-to-digital (A/D) converter implemented on the chip using a successive approximation technique, as shown in the A/D block diagram. 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}, $V_{RH/2}$, $V_{RH/4}$, and V_{RL}). The accuracy of these internal channels will not necessarily meet the accuracy specifications of the external channels.

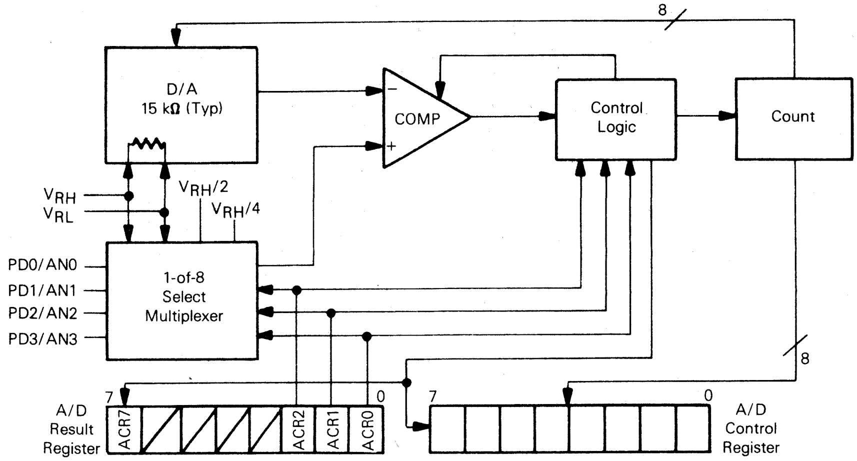
The multiplexer selection is controlled by the A/D control register (ACR) bits 0, 1, and 2; see the A/D input multiplexer selection table. This register is cleared during any reset condition.

Wherever 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 and held internally.

The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When conversion is complete, the digitized sample or 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. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

A/D BLOCK DIAGRAM



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

ADDITIONAL INFORMATION

For additional information, please refer to the following Advance Information Data Sheets:
MC6805R2 (ADI-845-R2)   MC68705R2 (AD-858-R1).

MOTOROLA

MC6805T2

Advance Information

8-BIT MICROCOMPUTER UNIT WITH PLL LOGIC

The MC6805T2 Microcomputer Unit (MCU) with PLL logic 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, 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. A comparison of the key features of several members of the M6805 Family of microcomputers is shown on the last page of this data sheet. The following are some of the hardware and software highlights of the MC6805T2 MCU.

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 (8 Lines are LED Compatible)
- On-Chip Clock 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 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:

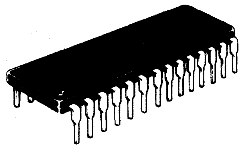
- Internal 8-Bit Timer with Selectable Clock Source (External Timer Input or Internal Machine Clock)
- Timer Prescaler Option (7 Bits 2^N)
- 8 Bidirectional I/O Lines with TTL or TTL/CMOS Interface Option
- 4 Vectored Interrupts; Timer, Software, and External

® EXORciser is a registered trademark of Motorola Inc.

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

8-BIT
MICROCOMPUTER
WITH PLL LOGIC



P SUFFIX
PLASTIC PACKAGE
CASE 710

FIGURE 1 — PIN ASSIGNMENTS

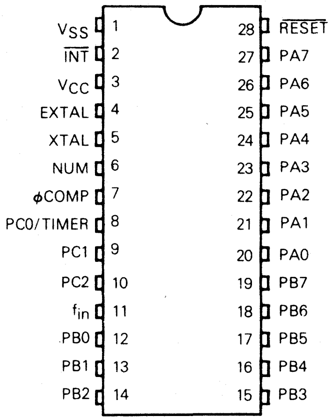
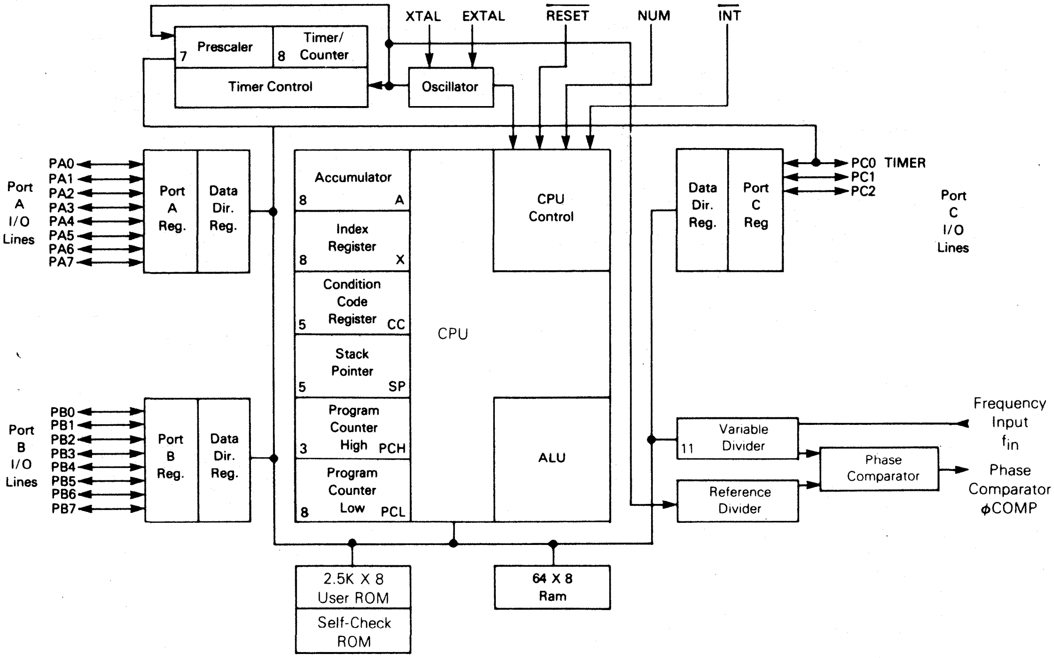


FIGURE 2 — MC6805T2 HMOS MICROCOMPUTER BLOCK DIAGRAM



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	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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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		120	
Ceramic	θ_{JA}	50	°C/W
Cerdip		60	

POWER CONSIDERATIONS

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

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

Where:

$T_A = $ Ambient Temperature, °C
$\theta_{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:

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

Solving equations 1 and 2 for K gives:

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

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} = 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	μA
$\overline{\text{INT}}$ and TIMER Pulse Width	t_{WH}, t_{WL}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Capacitance = 1.0 μF)	t_{RHL}	–	100	–	ms
Input Frequency	f_{in}	1	–	16	MHz
Input Frequency Rise Time at f_{in}	t_{INR}	–	–	20	ns
Input Frequency Fall Time at f_{in}	t_{INF}	–	–	20	ns
Duty Cycle of f_{in} and External Input on EXTAL	–	40	–	60	%
Injection Pulse Active Time	t_{arr}	–	70	–	ns
$\overline{\text{INT}}$ Zero-Crossing Detection Input Frequency (for ±5° Accuracy	f_{INT}	0.03	–	1.0	kHz

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	V_{IH}				V
$\overline{\text{RESET}}$ ($4.75 \le V_{CC} \le 5.75$)		4.0	–	V_{CC}	
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{\text{INT}}$ ($4.75 \le V_{CC} \le 5.75$)		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	V_{IH}				V
Normal Mode		–	–	V_{CC}	
Self-Check Mode		–	9.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{\text{RESET}}$		−0.3	–	0.8	
$\overline{\text{INT}}$		−0.3	*	1.5	
All Other Except f_{in}		−0.3	–	0.8	
$\overline{\text{INT}}$ 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	C_{in}				pF
EXTAL		–	25	–	
All Others		–	10	–	
AC Coupled Input Voltage Swing	V_{FIP}	0.5	1.2	–	$V_{ac\ p\text{-}p}$
Input Current ($V_{IH} = V_{CC}$)	I_{FH}	–	–	40	μA
Ouput Low Current (V_{OL} = 1.0 V)	I_{CML}	–	300	–	μA
Output High Current ($V_{OH} = V_{CC} - 1$ V)	I_{CMH}	–	200	–	μA
Leakage Current ($V_{in} = V_{CC}$)	I_{OFF}	–	2	–	nA
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figures 11 and 12)					V
"Out of Reset"	V_{IRES+}	2.1	–	4.0	
"Into Reset"	V_{IRES-}	0.8	–	2.0	
Input Current	I_{in}				μA
TIMER (V_{in} = 0.4 V)		–	–	20	
$\overline{\text{INT}}$ (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	
$\overline{\text{RESET}}$ (V_{in} = 0.8 V) (External Capacitor Charging Current)		−4.0	–	−50	

*Due to internal biasing, this input (when unused) floats to approximately 2.0 V.

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}	3.5	–	–	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}	0.3	–	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}	0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	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}	0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

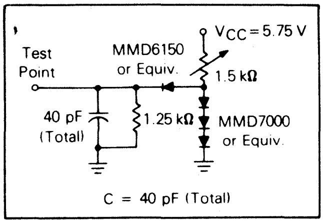


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

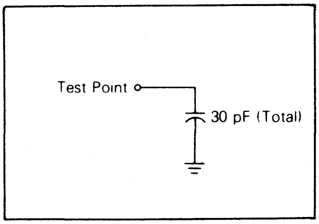
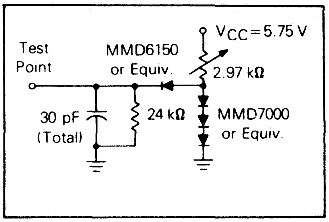


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$ — 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 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} — The high frequency digital input to the variable divider portion of the on-chip frequency synthesizer is on the f_{in} pin. The reference frequency for the phase lock loop is divided down from the crystal oscillator. Refer to the PHASE LOCK LOOP section 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 PLL for details. In Self-Check, ϕCOMP is raised to ≈9 V.

RESET — This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. 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 6. 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, 6 bytes of port I/O, 2 timer registers, and 2 PLL registers. The user ROM is split into four areas. The first area begins at memory location $080, which allows the user to access ROM locations utilizing the direct and table look-up indexed addressing modes. The second user ROM area begins at memory location $100. 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.

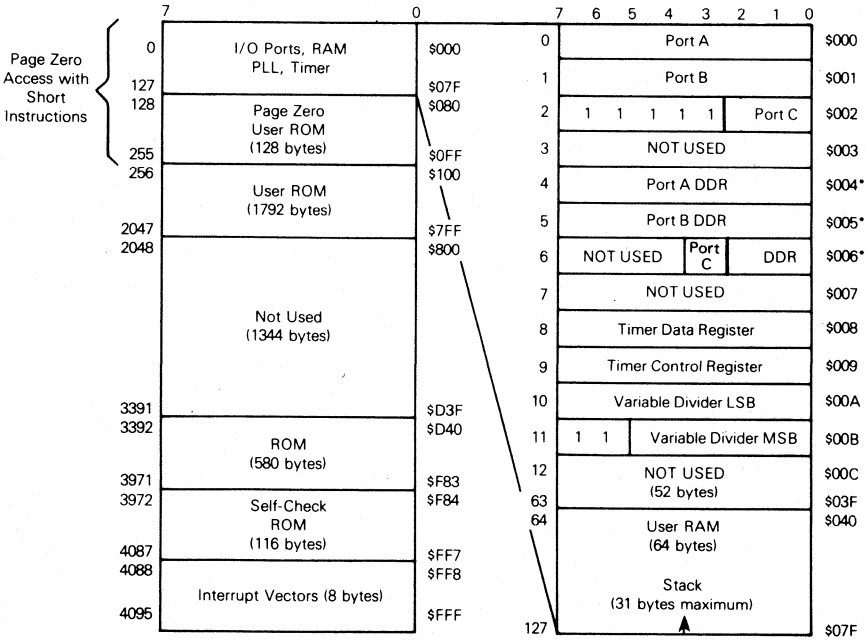
The register contents are pushed onto the stack in the order shown in Figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly 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.

REGISTERS

The MCU has five registers available to the programmer. They are shown in Figure 8 and are explained in the following paragraphs.

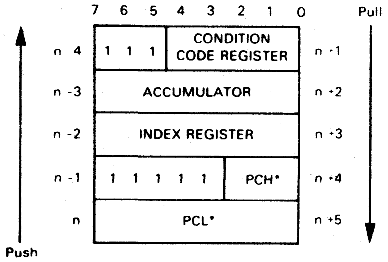
ACCUMULATOR (A) — The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

FIGURE 6 — MC6805T2 MCU ADDRESS MAP



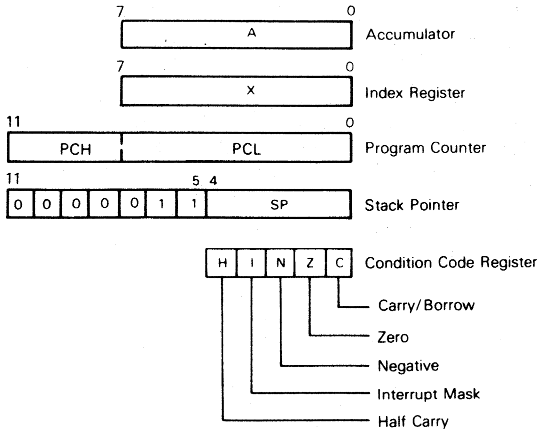
*Caution: Data Direction Register (DDRs) are write-only; they read as $FF.

FIGURE 7 — INTERRUPT STACKING ORDER



* For subroutine calls, only PCH and PCL are stacked

FIGURE 8 — PROGRAMMING MODEL



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 point 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 a 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 ($\overline{\text{INT}}$). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is reset.

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.

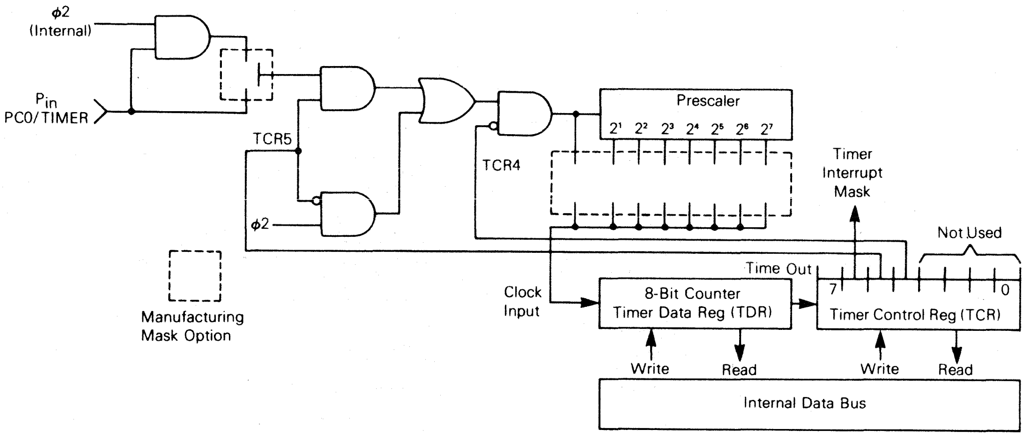
Carry/Borrow (C) — Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the LAST arithmetic operation. This bit is also affected during bit test and branch instructions plus shifts and rotates.

TIMER

The MCU timer circuitry is shown in Figure 9. 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 in the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine; see INTERRUPT section.

The timer clock input is established via bit 5 (TCR5) in the Timer Control Register. When this bit is set high, logical one (external mode), the timer clock source is the PCO/TIMER pin. In this mode a mask option is used to select either: a) the gated $\phi 2$ with PCO or b) the positive transition on PCO/TIMER as timer clock source. This allows easily performed pulse width or pulse count measurements. When TCR5 is low, logical zero, the timer clock source is the internal $\phi 2$.

FIGURE 9 — TIMER BLOCK DIAGRAM



Bit 4 in the Timer Control Register (TCR4) disables the timer clock source when set to logical one.

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 Powerup or Reset, the prescaler and counter are initialized with all logical ones; the timer interrupt request bit (bit 7) is cleared; the timer interrupt mask bit (bit 6) is set; the external timer source bit (bit 5) is cleared and the timer disable bit (bit 4) is cleared.

SELF-CHECK

The self-check capability of the MCU provides an internal check to determine if the port is functional. Connect the MCU as shown in Figure 10 and monitor the output of Port C bit 3 for an oscillation of approximately 3 Hz. 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 can be reset two ways: by the external reset input ($\overline{\text{RESET}}$) and during power up time. See Figure 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 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 12.

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 12, will provide sufficient delay. See Figure 17 for the complete reset sequence.

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 13 and 14. The crystal specifications are given in Figure 15.

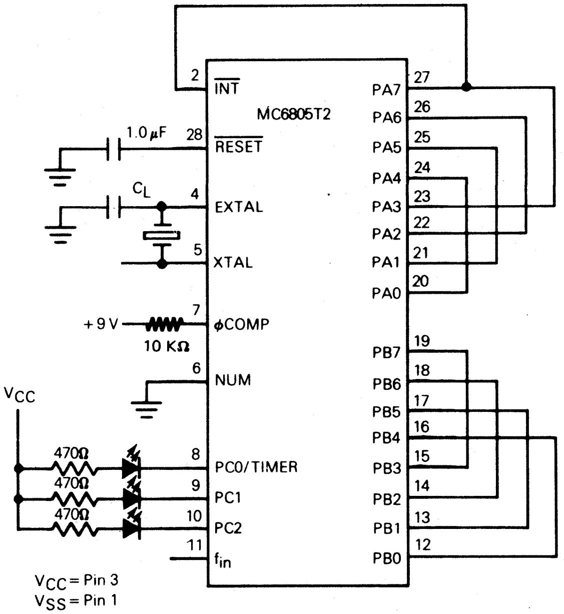
The crystal oscillator startup time is a function of many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply voltage. To ensure rapid oscillator startup, neither the crystal characteristics nor the load capacitance should exceed recommendations.

INTERRUPTS

The 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 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). Table 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. The interrupt priority applies to those pending when the CPU is ready to accept a new interrupt. $\overline{\text{RESET}}$ is listed in Table 1 because it is treated as an interrupt. However, it is not normally used as an interrupt. When the interrupt mask bit in the Condition Code Register is set the interrupt is latched for later interrupt execution.

FIGURE 10 — SELF-CHECK CONNECTIONS



V_{CC} = Pin 3
V_{SS} = Pin 1

FIGURE 11 — POWER AND $\overline{\text{RESET}}$ TIMING

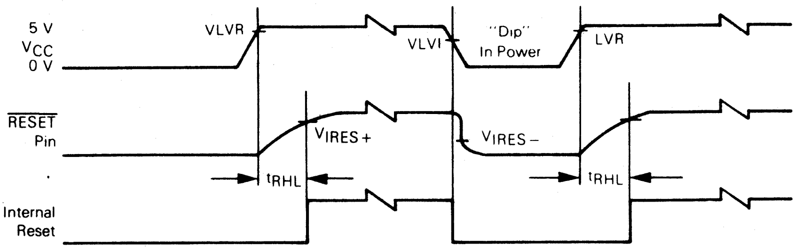


FIGURE 12 — POWERUP RESET DELAY CIRCUIT

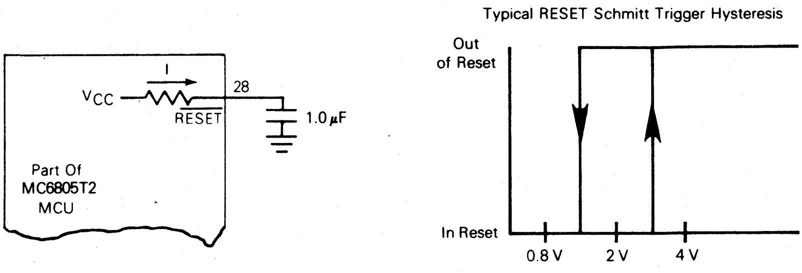
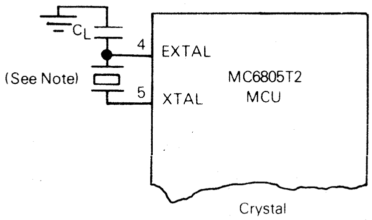
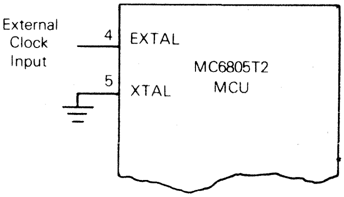


FIGURE 13 — CRYSTAL OSCILLATOR



Crystal

FIGURE 14 — EXTERNAL OSCILLATOR

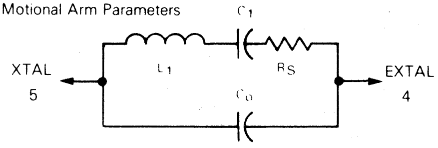


External Clock

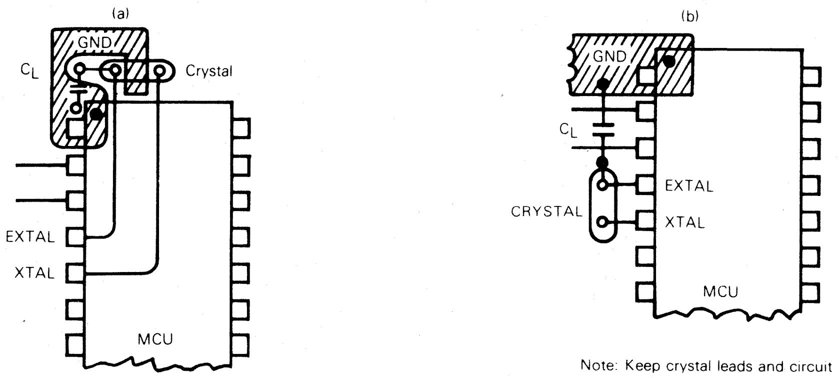
NOTE: The recommended C_L value with a 4.0 MHz crystal is 27 pF, maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the 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 15 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT

Recommended
Crystal Motional Arm Parameters



AT — Cut Parallel Resonance Crystal
$C_0 = 7$ 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 16 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

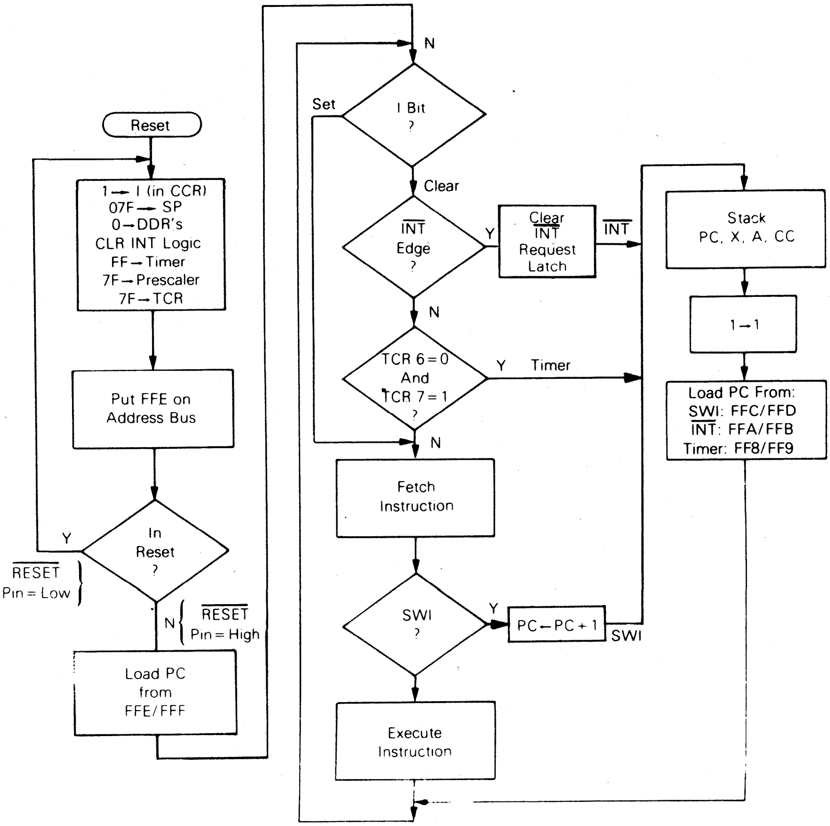


TABLE 1 — INTERRUPT PRIORITIES

Interrupt	Priority	Vector Address
$\overline{\text{RESET}}$	1	\$FFE and \$FFF
SWI	2*	\$FFC and \$FFD
$\overline{\text{INT}}$	3	\$FFA and \$FFB
Timer	4	\$FF8 and \$FF9

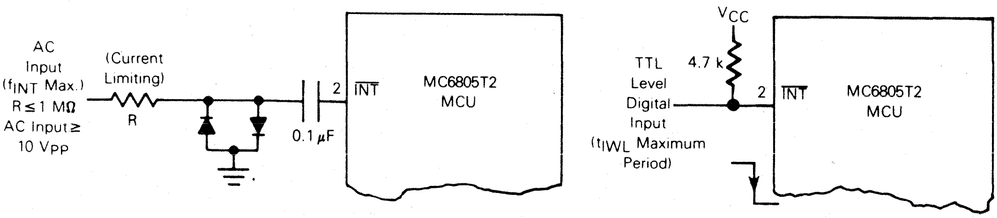
*Priority 2 applies when the I-bit in the Condition Code Register is set. When I = 0, SWI has a priority of 4, like any other instruction, the priority of $\overline{\text{INT}}$ thus becomes 2 and the timer becomes 3.

The external interrupt is internally synchronized and then latched 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 17, for use as a Zero-Crossing Detector. For digital applications the $\overline{\text{INT}}$ can be driven by a digital signal at a maximum period of t_{IWL}.

INPUT/OUTPUT

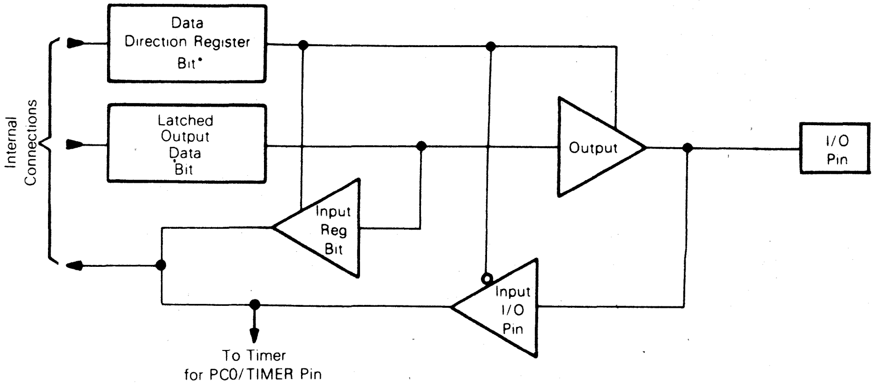
There are 19 input/output pins. (The $\overline{\text{INT}}$ pin may also be polled with branch instructions to provide an additional input pin.) All pins (Ports A, B, and C) are programmable as either inputs or outputs under software control of the corresponding Data Direction Registers (DDR). The port 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 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.

FIGURE 17 — TYPICAL INTERRUPT CIRCUITS


a – Zero Crossing Interrupt

b – Digital Signal Interrupt

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	3-State**	Pin

*DDR is a write-only register and reads as all 1's.

**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 19 provides some examples of port connections. The address map in Figure 6 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 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).

PHASE LOCK LOOP (PLL)

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 20. The equations governing the PLL are given in Figure 21.

TABLE 2 — MCU REGISTER CONFIGURATION

PORT DATA DIRECTION REGISTER (DDR)

7	0

(1) Write Only; reads as all 1's
(2) 1 = Output, 0 = Input. Cleared to 0 by Reset.
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

PORT DATA REGISTER

7	0

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002

TIMER CONTROL REGISTER (TCR)

$009	7	6	5	4	3	2	1	0
					1	1	1	1

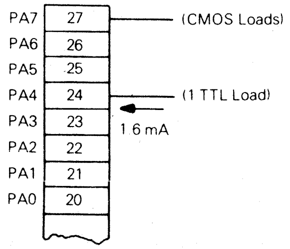
TCR7 – Timer Interrupt Request Status Bit: Set when TDR goes to zero, must be cleared by software. Cleared to 0 by Reset.
TCR6 – Timer Interrupt Mask Bit: 1 = timer interrupt masked (disabled). Set to 1 by Reset.
TCR5 – External Timer Source: 1 = External, 0 = Internal. Cleared to 0 by Reset.
TCR4 – Disable Timer: 1 = Timer Source Disconnected, 0 = Timer Input Enabled. Cleared to 0 by Reset.
TCR Bits 3, 2, 1, and 0 read as 1's (not used).

TIMER DATA REGISTER (TDR)

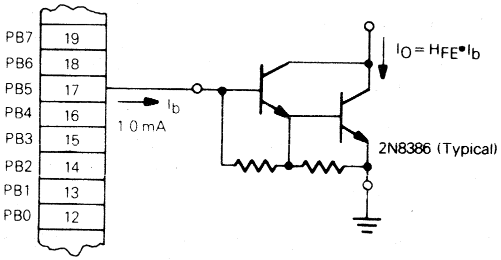
$008	7	0
	MSB	LSB

FIGURE 19 — TYPICAL PORT CONNECTIONS

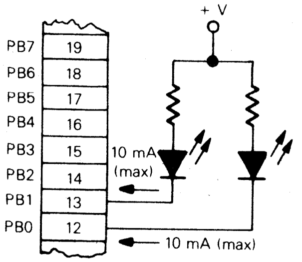
a. Output Modes



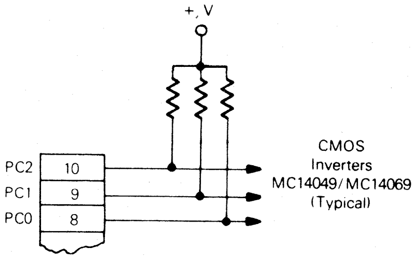
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

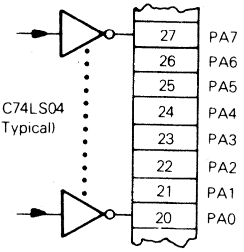


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

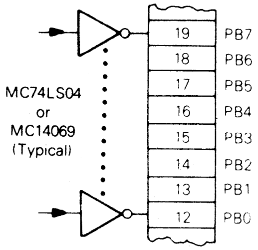


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors.

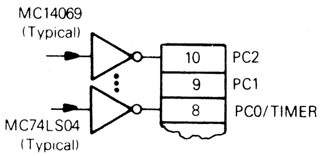
b. Input Modes



TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly.



CMOS and TTL Driving Port C Directly.

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 logical "1's". When the variable divider count has reached zero a preset pulse, f_{VAR}, is generated. This is used to reload the variable divider with the contents of the 14-bit latch as shown in Figure 22.

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. Figure 23 shows a typical error free manipulation of the 14-bit data in the fine tuning operations.

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 logical "1's".

The variable frequency input pin, f_{in}, is self biased requiring an ac coupled signal with a normal swing of 0.5 V. The input frequency range of f_{in} allows the device, together with a suitable prescaler, to cover the entire TV frequency spectrum.

REFERENCE DIVIDER

This 10-stage binary counter generates a reference frequency, f_{REF}, which is compared with the output of the variable divider. The reference divider is mask programmable, thus, allowing the user a choice of reference frequency, see Figure 22.

FIGURE 20 — PHASE LOCK LOOP AN AN RF FREQUENCY SYNTHESIZER

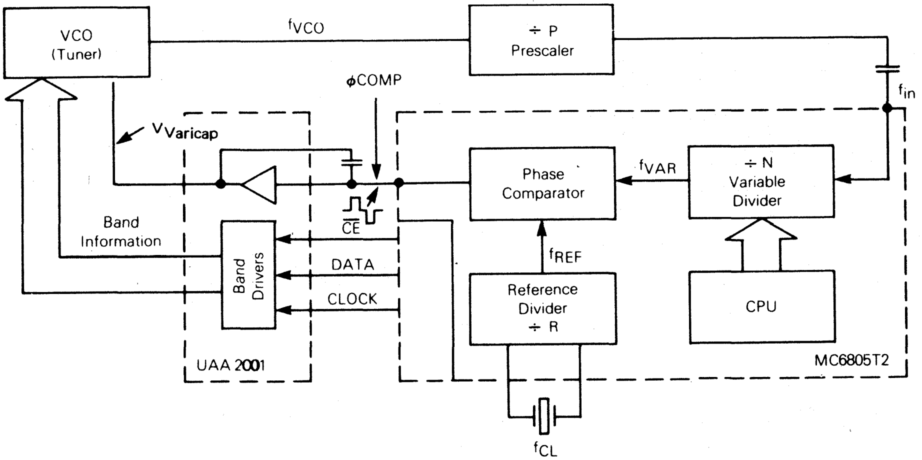


FIGURE 21 — PRINCIPAL PLL EQUATIONS

For a system in lock:

$f_{VAR} = f_{REF}$

since $f_{in} = f_{VAR} \bullet N$

$f_{VCO} = f_{in} \bullet P$

where P = prescaler division ratio

$f_{VCO} = f_{REF} \bullet P \bullet N$

Minimum frequency step =

$$\frac{\Delta f_{VCO}}{\Delta N} = f_{REF} \bullet P$$

e.g.: $f_{CL} = 4.00$ MHz

$R = 2^{10}$

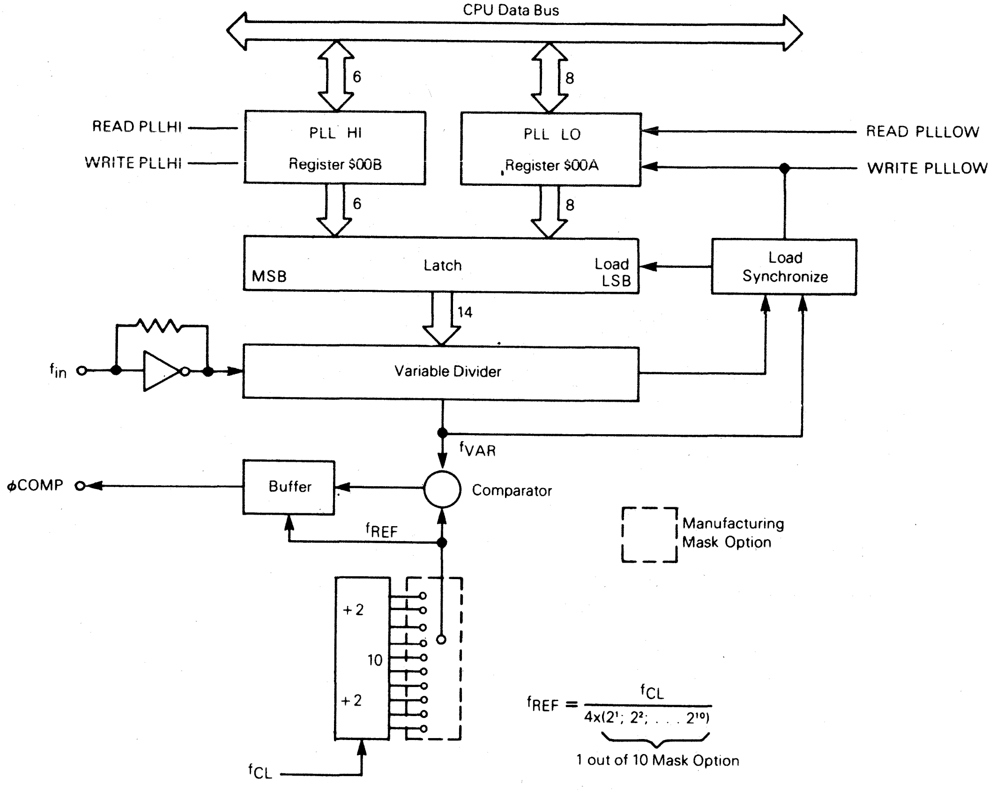
where R = the reference divider ratio

$P = 64$

$$\frac{\Delta f_{VCO}}{\Delta N} = 62.5 \text{ kHz}$$

$f_{REF} = 976.5$ Hz

FIGURE 22 — MC6805T2 PLL BLOCK DIAGRAM



$$f_{REF} = \frac{f_{CL}}{4x\underbrace{(2^1;\ 2^2;\ \ldots\ 2^{10})}}$$

FIGURE 23 — TYPICAL FINE TUNE EXAMPLE

```
FTUP    LDA     PLLLOW
        INCA                     check if LS byte=$FF (Reg $00A)
        BNE     TT1              if not increment only LS byte
        INC     PLLLOW           increment MSB (Reg $00B) before LSB
TT1     INC     PLLLOW
        •
        •
        •
FTDWN   TST     PLLLOW           check if LS byte=$00
        BNE     TT2              if not decrement only LS byte
        DEC     PLLHI            decrement MSB before LSB
TT2     DEC     PLLLOW
        •
        •
        •
```

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, φCOMP, as shown in Figures 24 and 25. The output waveform is then integrated, amplified, and the resultant dc voltage is applied to the voltage controlled oscillator.

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 zone, as shown in Figures 26 and 27.

A typical application, for a TV frequency synthesizer, is illustrated in Figure 28.

FIGURE 24 – PHASE COMPARATOR STATE DIAGRAM

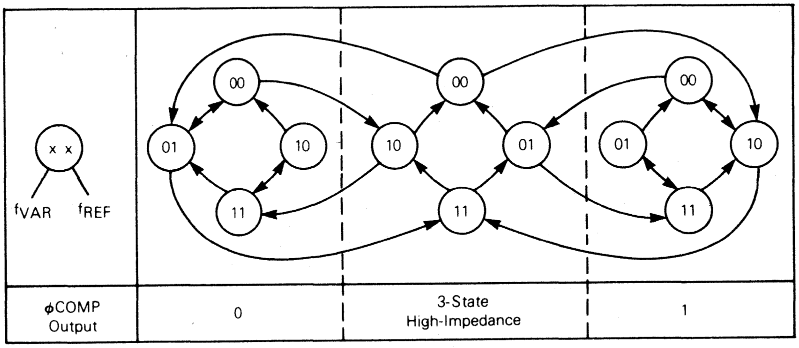


FIGURE 25 – PHASE COMPARATOR OUTPUT WAVEFORM

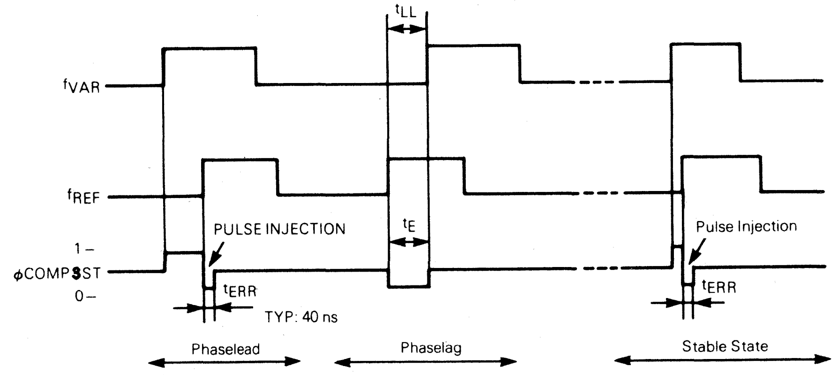


FIGURE 26 — PHASE COMPARATOR CHARACTERISTICS

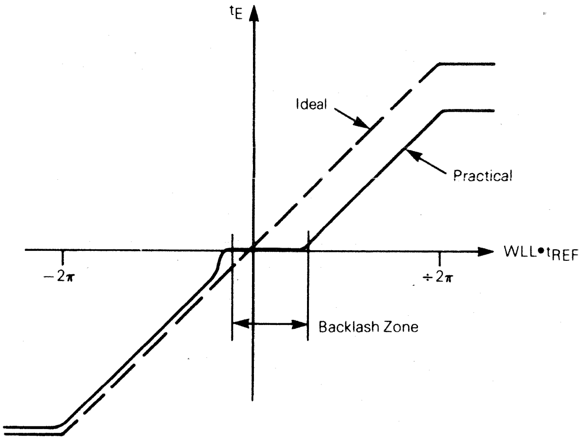


FIGURE 27 — PHASE COMPARATOR WITH PULSE INJECTION

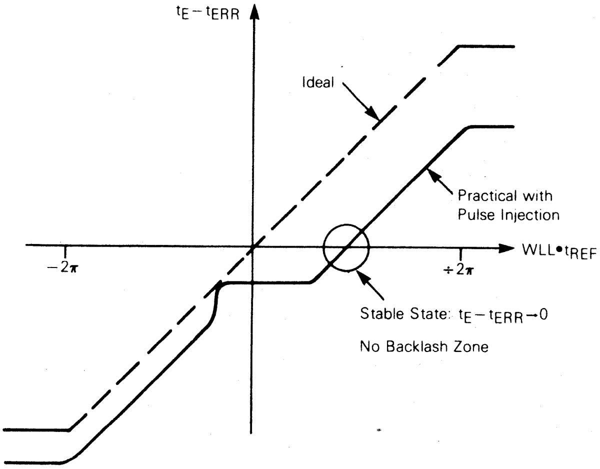
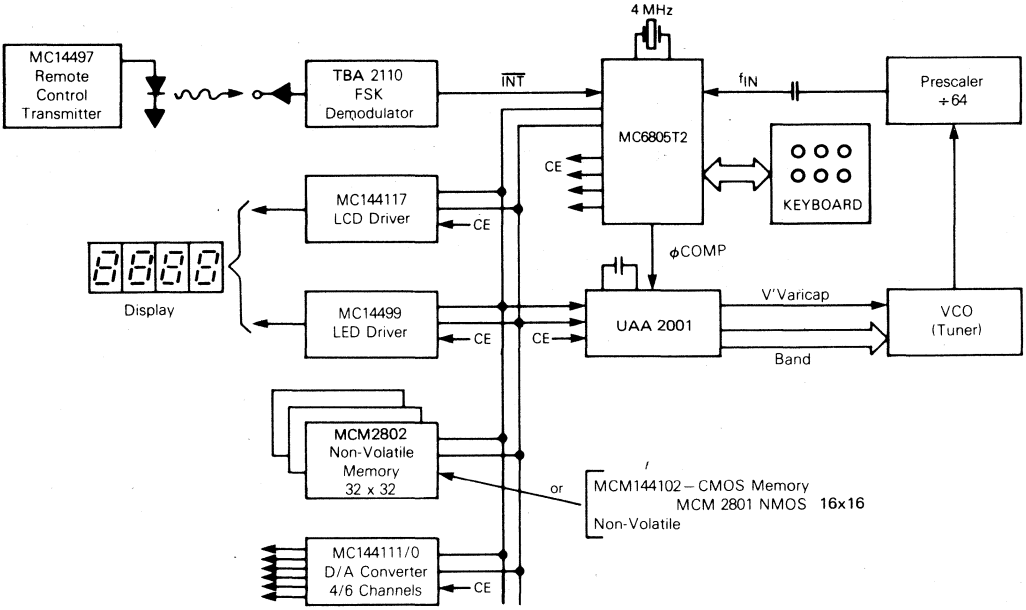


FIGURE 28 — A TYPICAL TV SYNTHESIZER APPLICATION

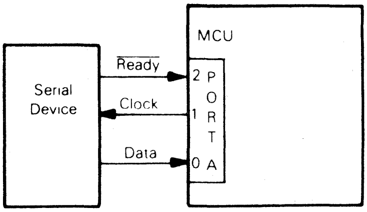


BIT MANIPULATION

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 paragraph), 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 29 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 29 — BIT MANIPULATION EXAMPLE



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

ADDRESSING MODES

The 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 address modes. EA is defined as the address from which the argument for an instruction is fectched 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 (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 511 ($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.

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 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 and condition (set or clear) is included in the opcode, which is to be tested 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 +130 to −125 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 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 Cautions 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 4

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

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 Cautions under INPUT/OUTPUT paragraph). 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.

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											●	●	●	●	●
BMS											●	●	●	●	●
BNE											●	●	●	●	●
BPL											●	●	●	●	●
BRA											●	●	●	●	●
BRN											●	●	●	●	●
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)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero		

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)	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

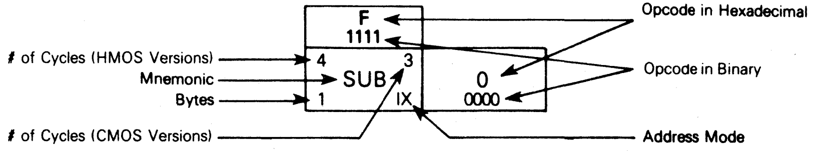
TABLE 9 — M6805 FAMILY INSTRUCTION OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	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 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEGA 1 INH	4 3 NEGX 1 INH	7 6 NEG 2 IX1	6 5 NEG 1 IX	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
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL						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 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL								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
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COMA 1 INH	4 3 COMX 1 INH	7 6 COM 2 IX1	6 5 COM 1 IX	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
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSRA 1 INH	4 3 LSRX 1 INH	7 6 LSR 2 IX1	6 5 LSR 1 IX			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
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL								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
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 RORA 1 INH	4 3 RORX 1 INH	7 6 ROR 2 IX1	6 5 ROR 1 IX			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
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASRA 1 INH	4 3 ASRX 1 INH	7 6 ASR 2 IX1	6 5 ASR 1 IX		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
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSLA 1 INH	4 3 LSLX 1 INH	7 6 LSL 2 IX1	6 5 LSL 1 IX		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
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROLA 1 INH	4 3 ROLX 1 INH	7 6 ROL 2 IX1	6 5 ROL 1 IX		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
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DECA 1 INH	4 3 DECX 1 INH	7 6 DEC 2 IX1	6 5 DEC 1 IX		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
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL							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
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INCA 1 INH	4 3 INCX 1 INH	7 6 INC 2 IX1	6 5 INC 1 IX		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
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TSTA 1 INH	4 3 TSTX 1 INH	7 5 TST 2 IX1	6 4 TST 1 IX		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
E 1110	10 5 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL						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
F 1111	10 5 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLRA 1 INH	4 3 CLRX 1 INH	7 6 CLR 2 IX1	6 5 CLR 1 IX	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

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
*	**CMOS Versions Only**

LEGEND

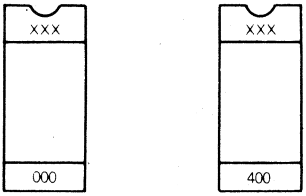


ORDERING INFORMATION

The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or 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 – 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 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, EXORsets, etc.

MC6805T2 MCU ORDERING INFORMATION

Date ______________________ Customer PO Number ______________

Customer Company ______________________________

Motorola Part Numbers

MC ______________

SC ______________

Address ______________________________

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
- ☐ Gated $\phi 2$
- ☐ Input Transition

Timer Prescaler
- ☐ 2^0 (divide by 1)
- ☐ 2^1 (divide by 2)
- ☐ 2^2 (divide by 4)
- ☐ 2^3 (divide by 8)

Reference Divider
- ☐ 2^{10} (divide by 1024)
- ☐ 2^9 (divide by 512)
- ☐ 2^8 (divide by 256)
- ☐ 2^4 (divide by 16)
- ☐ 2^5 (divide by 32)
- ☐ 2^6 (divide by 64)
- ☐ 2^7 (divide by 128)

- ☐ 2^7 (divide by 128)
- ☐ 2^6 (divide by 64)
- ☐ 2^5 (divide by 32)
- ☐ 2^4 (divide by 16)
- ☐ 2^3 (divide by 8)
- ☐ 2^2 (divide by 4)
- ☐ 2^1 (divide by 2)
- ☐ 2^0 (divide by 1)

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

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*

*Requires prior factory approval

Marking Information (12 Characters Maximum)

Title ______________________________

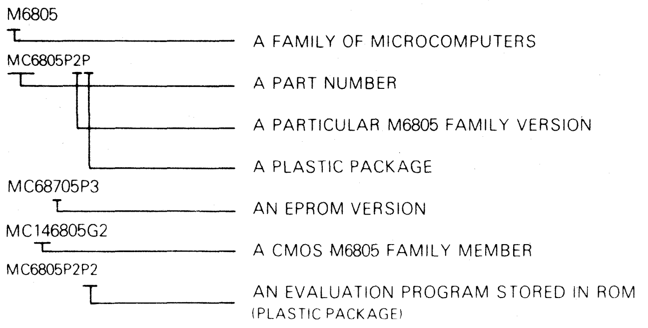
Signature ______________________________

M6805 FAMILY

	Available Now					
	MC6805P2	**MC6805U2**	**MC6805R2**	**MC68705P3**	**MC146805E2**	**MC146805G2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS	CMOS
Number of Pins	28	40	40	28	40	40
On-Chip RAM (Bytes)	64	64	64	112	112	112
On-Chip User ROM (Bytes)	1.1K	2K	2K	1.8K EPROM	None	2K
External Bus	None	None	None	None	Yes	None
Bidirectional I/O Lines	20	24	24	20	16	32
Unidirectional I/O Lines	None	8 Inputs	6 Inputs	None	None	None
Other I/O Features	Timer	Timer	Timer, A/D	Timer	Timer	Timer
EPROM Version	MC68705P3	MC68705U3	MC68705R3	–	None	None
STOP and WAIT	No	No	No	No	Yes	Yes

	Available				
	3Q81	**3Q81**	**3Q81**	**4Q81**	**1Q82**
	MC68705U3	**MC68705R3**	**MC6805T2**	**MC6805P4**	**MC146805F2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS
Number of Pins	40	40	28	28	28
On-Chip RAM (Bytes)	112	112	64	112	64
On-Chip User ROM (Bytes)	3.8K EPROM	3.8K EPROM	2.5K	1.1K	1K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	24	24	19	20	20
Unidirectional I/O Lines	8 Inputs	6 Inputs	None	None	None
Other I/O Lines	Timer	Timer, A/D	Timer, PLL	Timer	Timer
EPROM Version	--	–	None	MC68705P3	None
STOP and WAIT	No	No	No	No	Yes

M6805 FAMILY NOMENCLATURE



MOTOROLA

MC6805U2

HMOS

(HIGH DENSITY
N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

**8-BIT
MICROCOMPUTER**

Advance Information

8-BIT MICROCOMPUTER UNIT

The MC6805U2 Microcomputer Unit (MCU) is a member of the M6805 Family of low-cost single-chip Microcomputers. The 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. A comparison of the key features of several members of the M6805 Family of microcomputers is shown on the last page of this data sheet. The following are some of the hardware and software highlights of the MC6805U2 MCU.

HARDWARE FEATURES:

- 32 TTL/CMOS Compatible I/O Lines
  - 24 Bidirectional (8 Lines are LED Compatible)
  - 8 Input-Only
- 2048 Bytes of User ROM
- 64 Bytes of RAM
- Self-Check Mode
- Zero-Crossing Detect/Interrupt
- Timer with 7-Bit Prescaler
- 5 V Single Supply

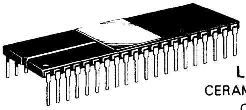
SOFTWARE FEATURES:

- 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

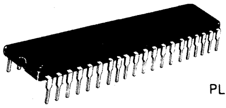
USER SELECTABLE OPTIONS:

- 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
- Vectored Interrupts: Timer, Software, and External

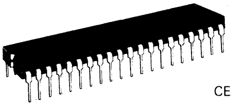
EXORciser is a registered trademark of Motorola Inc.



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

FIGURE 1 — PIN ASSIGNMENTS

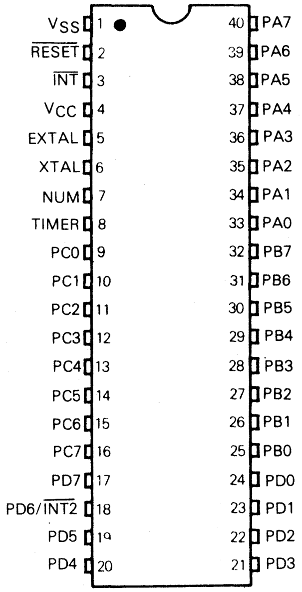
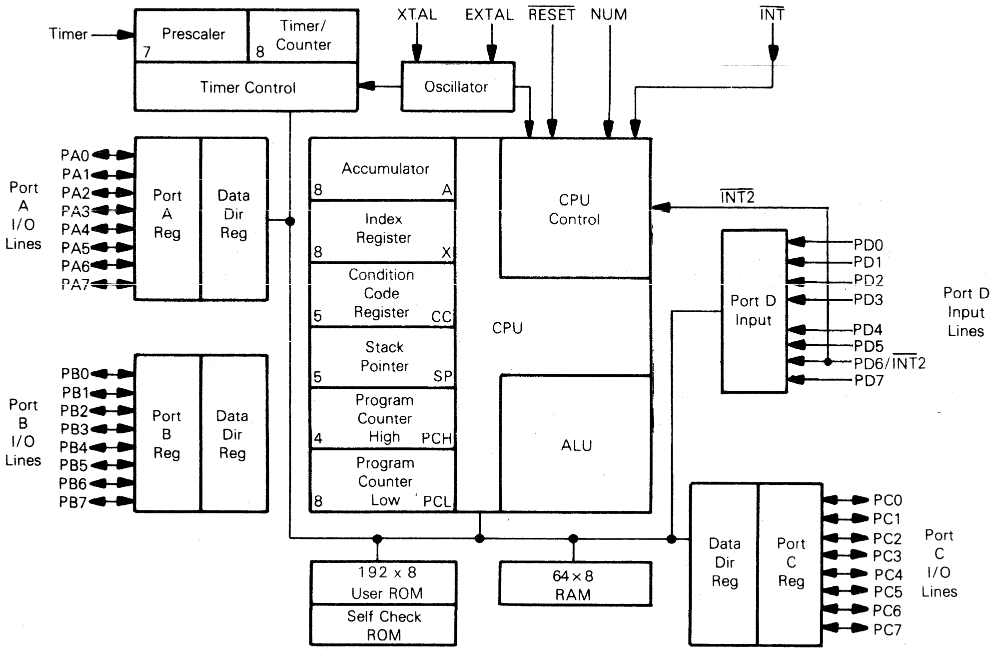


FIGURE 2 — MC6805U2 HMOS MICROCOMPUTER BLOCK DIAGRAM



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 Ceramic Cerdip	T_J	 150 175 175	°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 Plastic Ceramic Cerdip	θ_{JA}	 100 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:

$T_A \equiv$ 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} = GND, $T_A = 0°$ to 70°C Unless Otherwise Noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}				V
$\overline{\text{RESET}}$ $(4.75 \le V_{CC} \le 5.75)$		4.0	–	V_{CC}	
$(V_{CC} < 4.75)$		$V_{CC} - 0.5$	–	V_{CC}	
$\overline{\text{INT}}$ $(4.75 \le V_{CC} \le 5.75)$		4.0	*	V_{CC}	
$(V_{CC} < 4.75)$		$V_{CC} - 0.5$	*	V_{CC}	
All Other		2.0	–	V_{CC}	
Input High Voltage Timer	V_{IH}				V
Timer Mode		2.0	–	V_{CC}	
Self-Check Mode		–	9.0	15.0	
Input Low Voltage	V_{IL}				V
$\overline{\text{RESET}}$		−0.3	–	0.8	
$\overline{\text{INT}}$		−0.3	*	1.5	
All Other		−0.3	–	0.8	
$\overline{\text{RESET}}$ Hysteresis Voltages (See Figures 11, 12, and 13)					V
"Out of Reset"	V_{IRES+}	2.1	–	4.0	
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{\text{INT}}$ 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, $T_A = 0°C$	P_{INT}	–	600	–	mW
Input Capacitance	C_{in}				pF
XTAL		–	25	–	
All Other		–	10	–	
Low Voltage Recover	V_{LVR}	–	–	4.75	V
Low Voltage Inhibit	V_{LVI}	–	3.5	–	V
Input Current	I_{in}				µA
TIMER $(V_{in} = 0.4$ V$)$		–	–	20	
$\overline{\text{INT}}$ $(V_{in} = 2.4$ 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	
$\overline{\text{RESET}}$ $(V_{in} = 0.8$ V$)$ (External Capacitor Charging Current)		−4.0	–	−50	

*Due to internal biasing, this input (when unused) floats to approximately 2.0 V.

SWITCHING 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
Oscillator Frequency	f_{osc}	0.4	–	4.2	MHz
Cycle Time ($4/f_{osc}$)	t_{cyc}	0.95	–	10	μs
$\overline{\text{INT}}$ and TIMER Pulse Width	t_{WL}, t_{WH}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Cap = 1 μF)	t_{RHL}	–	100	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (for ±5° Accuracy)	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}	3.5	–	–	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}	−0.3	–	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA
Port D (Inputs)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input Current	I_{in}	–	–	20	μA

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

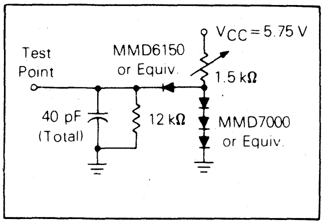


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

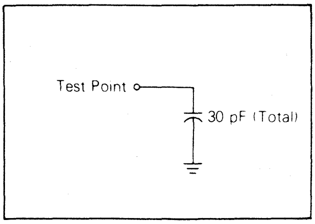
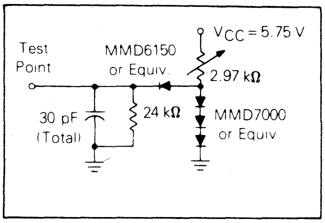


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$ — This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additonal information.

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 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 INTERNAL CLOCK GENERATOR OPTIONS for recommendations about these inputs.

TIMER — This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

$\overline{\text{RESET}}$ — 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 $\overline{\text{RESET}}$ low. Refer to RESETS for additional information.

NUM (Non-User Mode) — This pin is not for user application and must be connected to V_{SS}.

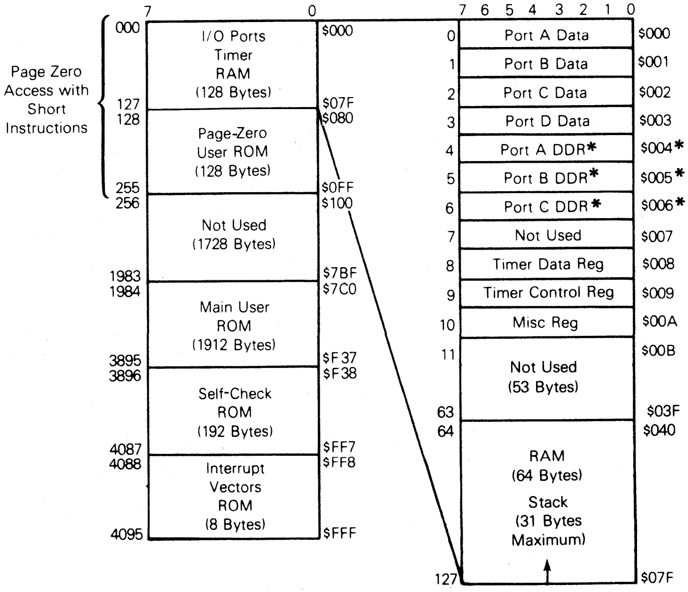
INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) — These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs, under software control of the data direction registers. Port D is for digital input only and bit 6 may be used for a second interrupt $\overline{\text{INT2}}$. Refer to INPUT/OUTPUT and INTERRUPTS for additional information.

MEMORY — The MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The MC6805U2 MCU has implemented 2314 of these bytes. This consists of: 2048 user ROM bytes, 192 self-check ROM bytes, 64 user RAM bytes, 7 port I/O bytes, 2 timer registers, and a miscellaneous register; see Figure 6 for the Address Map. The user ROM has been split into three areas. The first area is memory locations \$080 to \$0FF, and allows the user to access the ROM locations utilizing the direct and table look-up indexed addressing modes. The main user ROM area is from \$7C0 to \$F37. The last 8 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 $\overline{\text{INT2}}$ miscellaneous register. Of the 64 RAM 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.

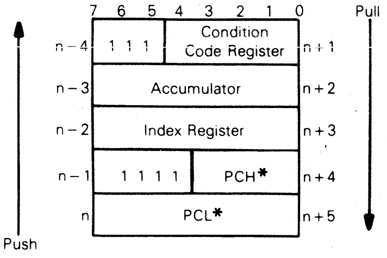
FIGURE 6 — MC6805U2 MCU ADDRESS MAP



*Caution: Data Direction Registers (DDRs) are write-only; they read as \$FF.

The contents of the CPU registers are pushed onto the stack in the order shown in Fig. 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.

FIGURE 7 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCH and PCL are stacked

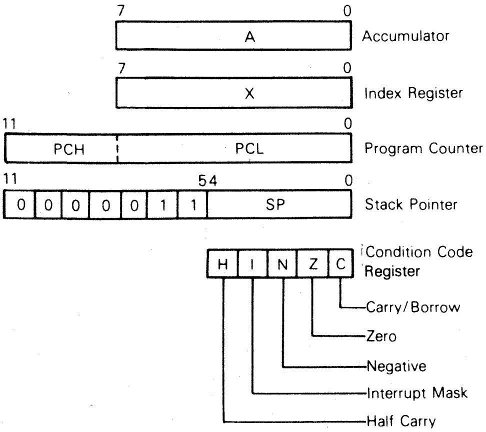
CENTRAL PROCESSING UNIT

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 8 and are explained in the following paragraphs.

FIGURE 8 — 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 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. 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 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).

CONDITION CODE REGISTERS (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.

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 interrupts ($\overline{\text{INT}}$ and $\overline{\text{INT}}$2) 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 one).

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 MC6805U2 MCU timer circuitry is shown in Figure 9. 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 in the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine (see the INTERRUPT section). THE TIMER INTERRUPT REQUEST BIT MUST BE CLEARED BY SOFTWARE. The timer and $\overline{\text{INT2}}$ share the same interrupt vector. THE INTERRUPT ROUTINE MUST CHECK THE REQUEST BITS TO DETERMINE THE SOURCE OF THE INTERRUPT.

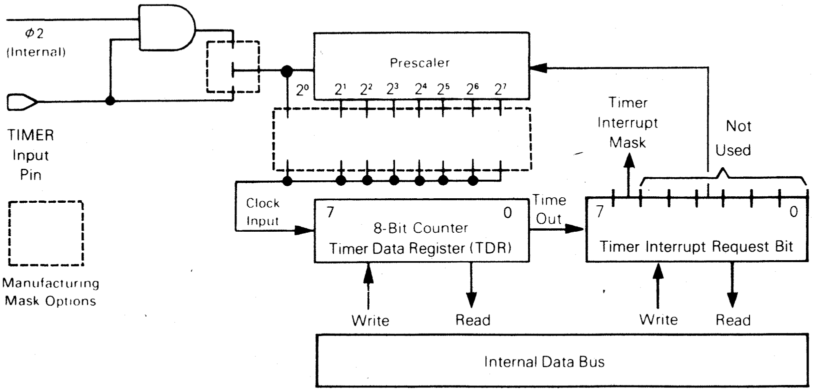
The clock input to the timer can be from an external source (decrementing of Timer Counter occurs on a positive transition of the external source) applied to the TIMER input pin, or it can be the internal $\phi2$ signal. When the $\phi2$ signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. (NOTE: For ungated $\phi2$ clock input 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. To avoid truncation errors, the prescaler is cleared when bit 3 of Timer Control Register is written to a logic 1 (this bit always reads as a logic 0).

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.

FIGURE 9 — TIMER BLOCK DIAGRAM



SELF-CHECK

The self-check capability of the MC6805U2 MCU provides an internal check to determine if the part if functional. Connect the MCU as shown in Figure 10 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 8, energizes the ROM-based self-check feature. The self-check program exercises the RAM, ROM, timer, interrupts, and I/O ports.

Two of the self-check subroutines (the RAM and ROM tests) can be called by a user program with a JSR or BSR instruction. The timer routine may also be called if the timer input is the internal $\phi2$ clock.

RAM SELF-CHECK SUBROUTINE — The RAM self-check is called at location $F6F 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.

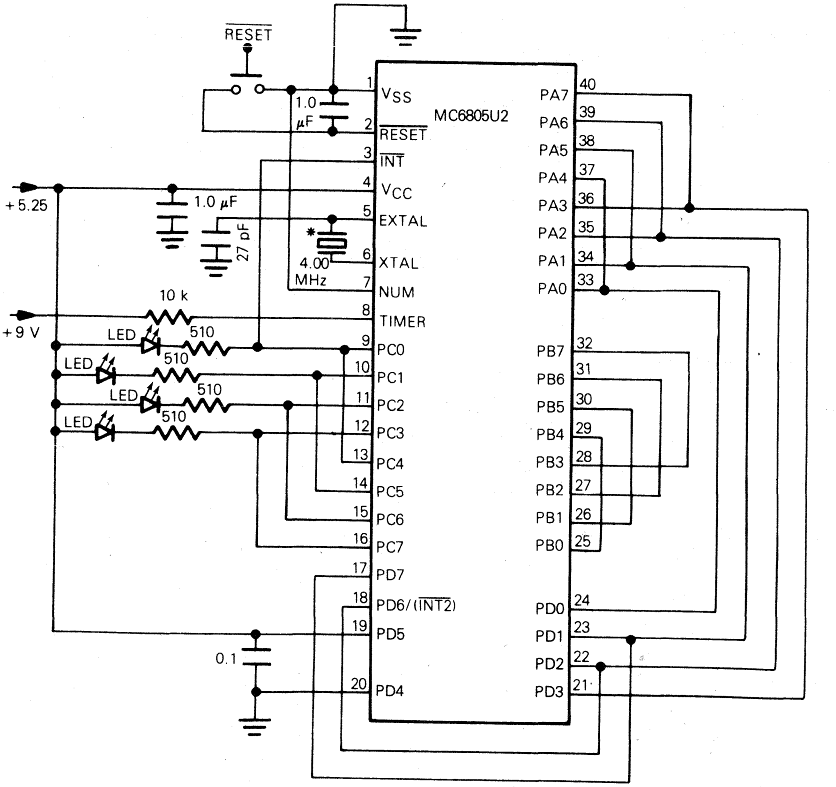
ROM CHECKSUM SUBROUTINE — The ROM self-check is called at location $F8A and returns with the Z-bit cleared if any error was found; otherwise Z = 1. X = 0 on return, and A is zero if the test passed. RAM locations $040-$043 are overwritten. The checksum is the compliment of the exclusive OR of the contents of the user ROM.

TIMER SELF-CHECK SUBROUTINE – The timer self-check is called at location $FCF 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 $\phi 2$ clock must be the clocking source and interrupts must be disabled. Also, on exit, the clock is running and the interrupt mask 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.

FIGURE 10 – SELF-CHECK CONNECTIONS



*This connection depends on clock oscillator user selectable mask option. Use jumper if the RC mask option is selected.

LED Meanings

C0	C1	C2	C3	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 Part

Anything else bad Part, Bad Port 3, Bad ISP, etc.

RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{\text{RESET}}$), and by an optional internal low-voltage detect circuit; see Figure 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 it if senses a 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 12.

Upon power-up, a delay of t_{RHL} milliseconds 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 13 typically provides sufficient delay. See Figure 17 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 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 14. The Crystal specifications are given in Figure 15. A resistor selection graph is shown in Figure 16.

The crystal oscillator start-up time is a function of many variables: crystal parameters (especially Rs) oscillator load capacitances, IC parameters, ambient temperatures, supply voltage, and supply voltage turn-on time. To ensure rapid oscillator start-up, neither the crystal characteristics nor the load capacitances should exceed recommendations.

FIGURE 11 — POWER AND RESET TIMING

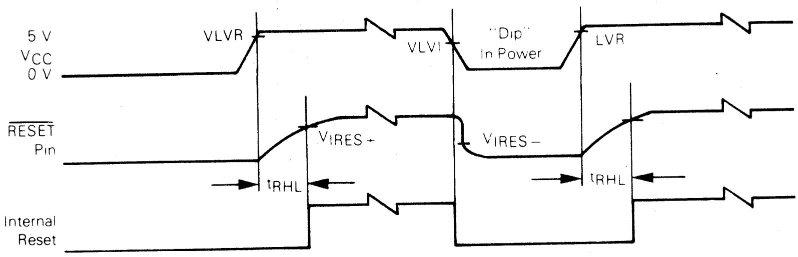


FIGURE 12 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

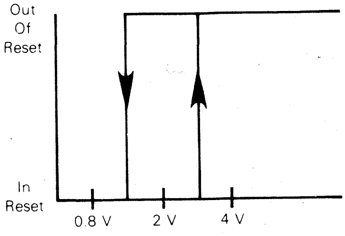


FIGURE 13 — POWER UP RESET DELAY CIRCUIT

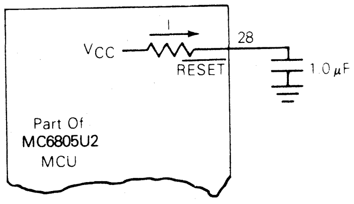
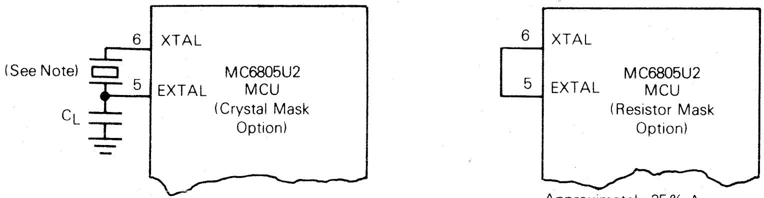
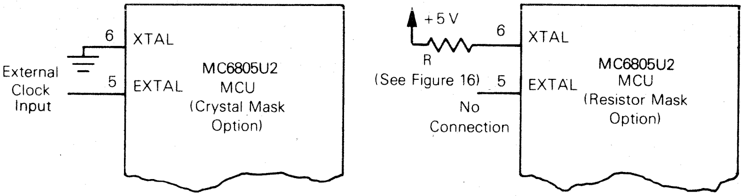


FIGURE 14 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% Accuracy
Typical $t_{CYC} = 1.25\ \mu s$
External Jumper



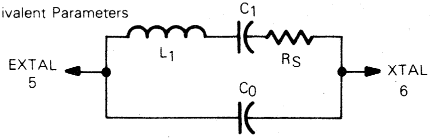
External Clock

Approximately 10% Accuracy
External Resistor
(Excludes Resistor Tolerance)

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 scalled 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.

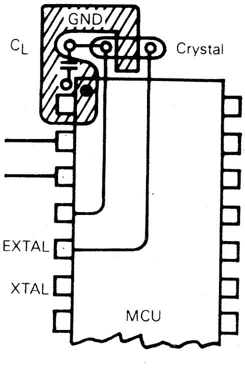
FIGURE 15 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT

Crystal Motional Arm
Equivalent Parameters

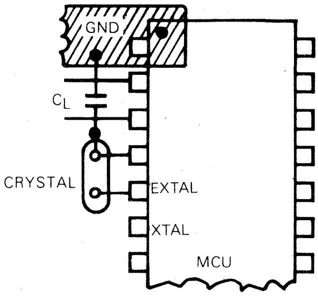


AT — Cut Parallel Resonance Crystal
$C_0 = 7$ pF Max
FREQ = 4.0 MHz @ $C_L = 24$ pF
$R_S = 50$ ohms Max.

(a)

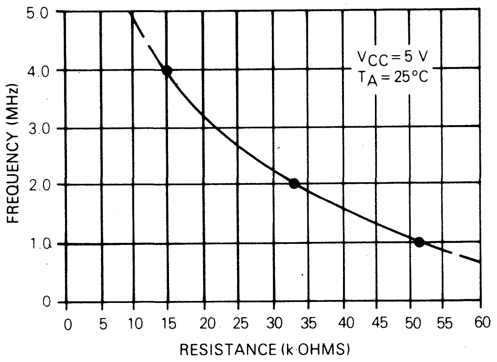


(b)



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

FIGURE 16 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION

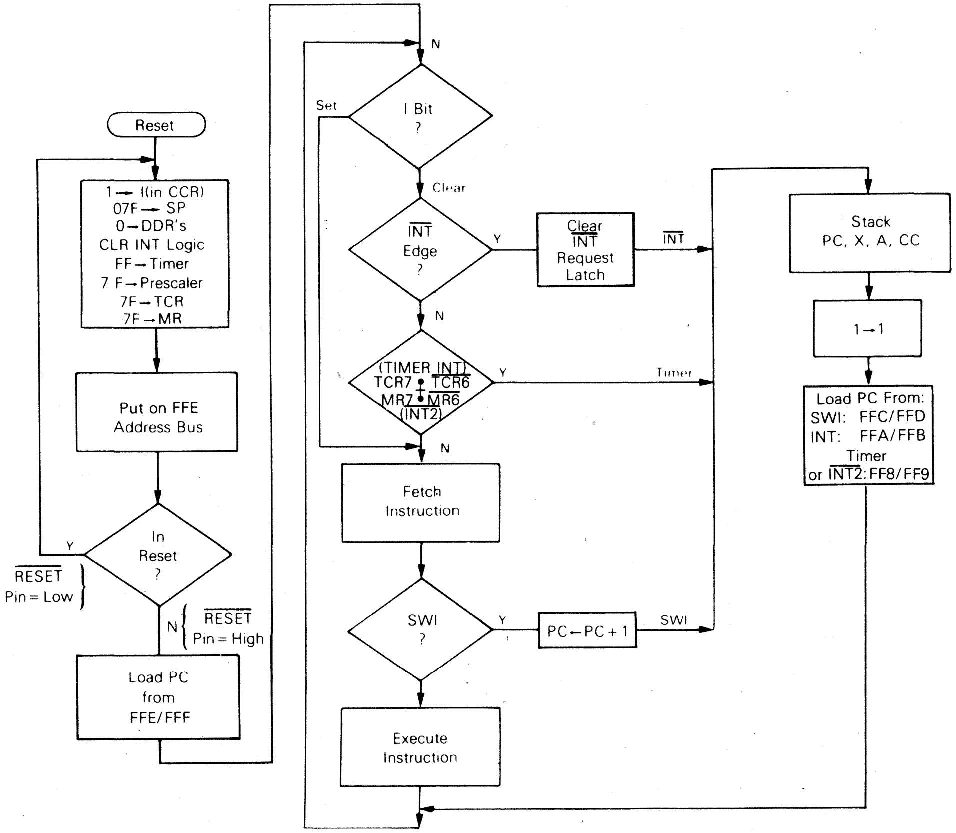


INTERRUPTS

The MC6805U2 MCU can be interrupted four different ways: through the external interrupt ($\overline{\text{INT}}$) input pin, the internal timer interrupt request, the external port D bit 6 ($\overline{\text{INT2}}$) input pin, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU 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.

Refer to Figure 17 for a flowchart. The interrupt service routines must end with a Return from Interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt. Table 1 provides a listing of the interrupts, their priority, and the address of the vector which

FIGURE 17 — RESET AND INTERRUPT PROCESSING FLOWCHART



contains the starting address of the appropriate interrupt service routine. The interrupt priority applies to those pending when the CPU is ready to accept an interrupt or a new interrupt. $\overline{\text{RESET}}$ is listed in Table 1 because it is processed similiar to an interrupt. However, it is not normally used as an interrupt. When the interrupt mask bit in the Condition Code Register is set the interrupt is latched for later interrupt execution.

NOTE

The timer and $\overline{\text{INT2}}$ share the same vector address. The interrupt routine must determine the source by examining the interrupt request bits (TCR7 and MR7). Both TCR7 and MR7 can only be written to 0 by software.

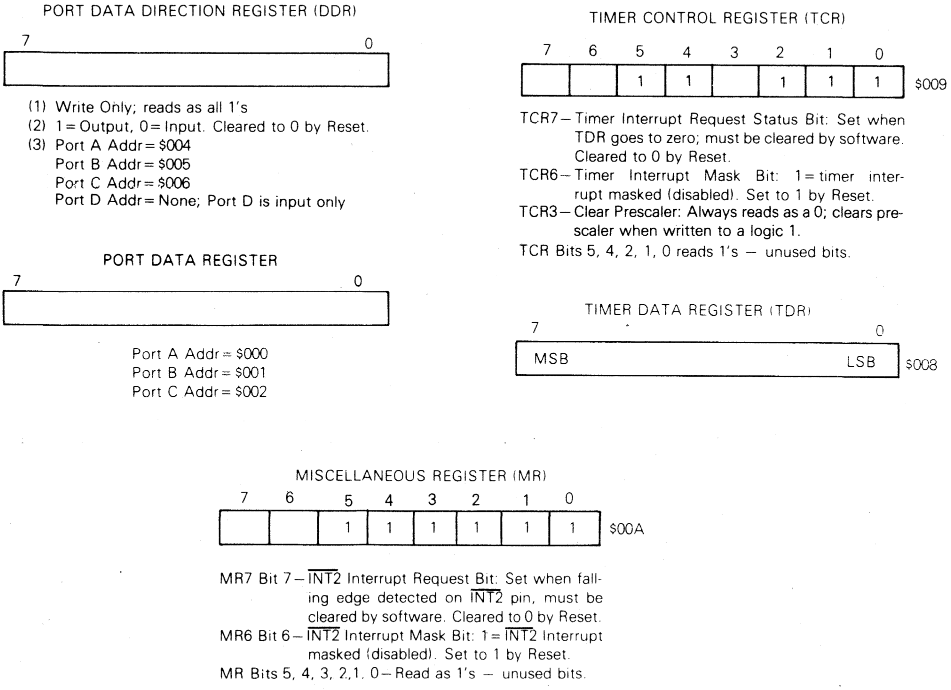
The external interrupts, $\overline{\text{INT}}$ and $\overline{\text{INT2}}$, are set on the falling edge of the input signal. The $\overline{\text{INT2}}$ has an interrupt request bit (bit 7) and a mask bit (bit 6) located in the Miscellaneous Register (MR); refer to Figure 18. The $\overline{\text{INT2}}$ interrupt is inhibited when the mask bit is set. The $\overline{\text{INT2}}$ is always readable as a digital input of Port D. The $\overline{\text{INT2}}$ and timer interrupt request bits, it set, causes the MCU to process an interrupt when the condition code I-bit is clear.

TABLE 1 — INTERRUPT PRIORITIES

Interrupt	Priority	Vector Address
$\overline{\text{RESET}}$	1	\$FFE and \$FFF
SWI	2*	\$FFC and \$FFD
$\overline{\text{INT}}$	3	\$FFA and \$FFB
TIMER/$\overline{\text{INT2}}$	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 4, like any other instruction, the priority of $\overline{\text{INT}}$ thus becomes 2 and the timer becomes 3

FIGURE 18 — MCU REGISTER CONFIGURATION



(1) Write Only; reads as all 1's
(2) 1 = Output, 0 = Input. Cleared to 0 by Reset.
(3) Port A Addr = \$004
Port B Addr = \$005
Port C Addr = \$006
Port D Addr = None; Port D is input only

TCR7 — Timer Interrupt Request Status Bit: Set when TDR goes to zero; must be cleared by software. Cleared to 0 by Reset.
TCR6 — Timer Interrupt Mask Bit: 1 = timer interrupt masked (disabled). Set to 1 by Reset.
TCR3 — Clear Prescaler: Always reads as a 0; clears prescaler when written to a logic 1.
TCR Bits 5, 4, 2, 1, 0 reads 1's — unused bits.

Port A Addr = \$000
Port B Addr = \$001
Port C Addr = \$002

MR7 Bit 7 — $\overline{\text{INT2}}$ Interrupt Request Bit: Set when falling edge detected on $\overline{\text{INT2}}$ pin, must be cleared by software. Cleared to 0 by Reset.
MR6 Bit 6 — $\overline{\text{INT2}}$ Interrupt Mask Bit: 1 = $\overline{\text{INT2}}$ Interrupt masked (disabled). Set to 1 by Reset.
MR Bits 5, 4, 3, 2, 1, 0 — Read as 1's — unused bits.

A sinusoidal input signal (f_{INT} maximum) can be used to operate an external interrupt ($\overline{INT}$), as shown in Figure 19, 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. For digital applications, the $\overline{INT}$ can be driven directly by a digital signal at a maximum period of t_{WL}. 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.

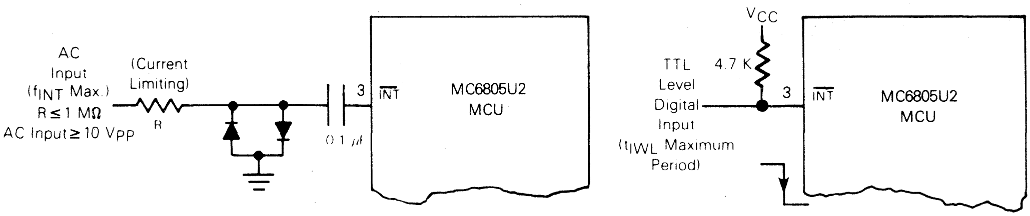
A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWI's are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 32 input or input/output pins. The $\overline{INT}$ 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 "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 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 20.

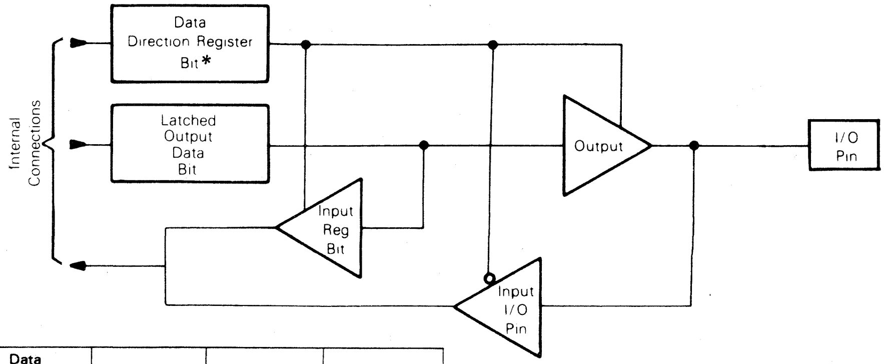
All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs using a mask option. Port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only; thus, there is no corresponding DDR. When programmed as outputs, Port B is capable of sinking 10 milliamperes and sourcing 1.0 milliampere on each pin.

FIGURE 19 — TYPICAL INTERRUPT CIRCUITS


a – Zero Crossing Interrupt

b – Digital Signal Interrupt

FIGURE 20 — 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 1's.

** 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 21 provides some examples of port connections. The Address Map in Figure 6 gives the addresses of data registers and DDRs. The Register Configuration is provided in Figure 18.

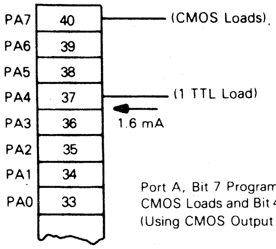
CAUTION

> The corresponding DDRs for Ports A, B, and C are write-only registers (locations $004, $005, and $006). A read operation on these registers 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.

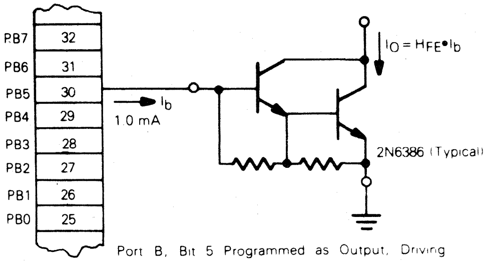
The latched output data bit (see Figure 20) may always be written. Therefore, any write to a port writes all of its data bits even though the port DDR is set to input. This may be used to initialize the data registers and avoid undefined outputs. However, care must be exercised when using read/modify/write 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 21 — TYPICAL PORT CONNECTIONS

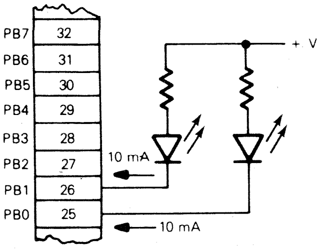
a. Output Modes



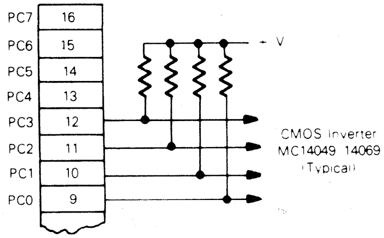
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 one TTL Load Directly. (Using CMOS Output Option)



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly

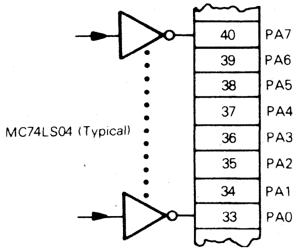


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

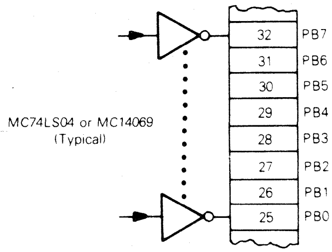


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors

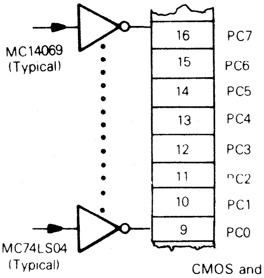
b. Input Modes



TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly



CMOS and TTL Driving Port C Directly.

BIT MANIPULATION

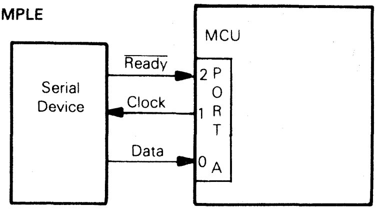
The MC6805U2 MCU has the ability to set or clear any single RAM or input/output bit (except the Data Direction Registers, see Caution under INPUT/OUTPUT paragraph) with a single instruction (BSET, BCLR). Any bit in page zero including ROM, except the DDRs, can be tested using the BRSET and BRCLR instructions and the program branches as a result of its state. The carry bit (C) equals the value of the bit referenced by BRSET and BRCLR. 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 single I/O bits as control lines.

The coding example in Figure 22 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, LBS 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 22 — BIT MANIPULATION EXAMPLE

```
        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
        ASR       RAMLOC               MOVE C-BIT INTO RAM
```



ADDRESSING MODES

The MC6805U2 MCU has 10 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 Family User 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 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 with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-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.

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 +129 to −126 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.

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 (§1 FE 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, 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 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 condition (set or clear) 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 clear in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 location of memory. The span of branching is from +130 to −125 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 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 instructions with no other arguments, are included in this mode. These instructions are one byte long.

INSTRUCTION SET

The MC6805U2 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 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 INPUT/OUTPUT paragraph. 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 3.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

BIT MANIPULATION INSTRUCTIONS — The 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 test and branch operations. Refer to Table 5.

CONTROL INSTRUCTION — The control instructions control the MCU operations during program execution. Refer to Table 6.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 7.

OPCODE MAP — Table 8 is an opcode map for the instruction used on the MCU.

TABLE 2 — 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 3 — 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 4 — 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 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 5 — BIT MANIPULATION INSTRUCTIONS

Function	Mnemonic	Addressing Modes Bit Set/Clear Op Code	 # Bytes	 # Cycles	 Bit Test and Branch 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 6 — 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 7 — 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)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero		

TABLE 7 — 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)	C	Carry/Borrow
I	Interrupt Mask	Λ	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

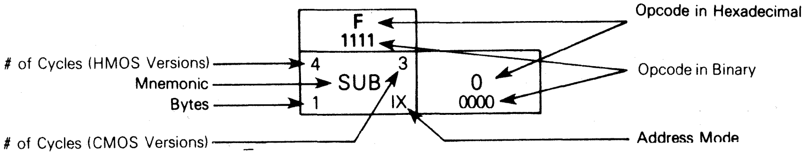
TABLE 8 — M6805 FAMILY OPCODE MAP

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	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 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEGA 1 INH	4 3 NEGX 1 INH	7 6 NEG 2 IX1	6 5 NEG 1 IX	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
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL						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 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL								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
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COMA 1 INH	4 3 COMX 1 INH	7 6 COM 2 IX1	6 5 COM 1 IX	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
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSRA 1 INH	4 3 LSRX 1 INH	7 6 LSR 2 IX1	6 5 LSR 1 IX			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
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL								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
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 RORA 1 INH	4 3 RORX 1 INH	7 6 ROR 2 IX1	6 5 ROR 1 IX			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
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASRA 1 INH	4 3 ASRX 1 INH	7 6 ASR 2 IX1	6 5 ASR 1 IX		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
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSLA 1 INH	4 3 LSLX 1 INH	7 6 LSL 2 IX1	6 5 LSL 1 IX		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
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROLA 1 INH	4 3 ROLX 1 INH	7 6 ROL 2 IX1	6 5 ROL 1 IX		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
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DECA 1 INH	4 3 DECX 1 INH	7 6 DEC 2 IX1	6 5 DEC 1 IX		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
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL							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
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INCA 1 INH	4 3 INCX 1 INH	7 6 INC 2 IX1	6 5 INC 1 IX		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
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TSTA 1 INH	4 3 TSTX 1 INH	7 5 TST 2 IX1	6 4 TST 1 IX		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
E 1110	10 5 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL						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
F 1111	10 5 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLRA 1 INH	4 3 CLRX 1 INH	7 6 CLR 2 IX1	6 5 CLR 1 IX	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

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
*	CMOS Versions Only

LEGEND

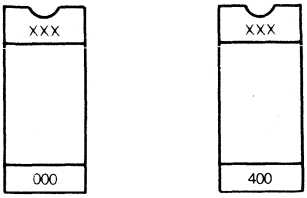


ORDERING INFORMATION

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 you 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 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 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 .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.

The MC68705U3 EPROM MCU may be programmed with the customer program. Note that the MC68705U3 contains 3776 bytes of EPROM memory while the MC6805U2 ROM memory is 2048 bytes.

M6805 FAMILY

	Available Now					
	MC6805P2	**MC6805U2**	**MC6805R2**	**MC68705P3**	**MC146805E2**	**MC146805G2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS	CMOS
Number of Pins	28	40	40	28	40	40
On-Chip RAM (Bytes)	64	64	64	112	112	112
On-Chip User ROM (Bytes)	1.1K	2K	2K	1.8K EPROM	None	2K
External Bus	None	None	None	None	Yes	None
Bidirectional I/O Lines	20	24	24	20	16	32
Unidirectional I/O Lines	None	8 Inputs	6 Inputs	None	None	None
Other I/O Features	Timer	Timer	Timer, A/D	Timer	Timer	Timer
EPROM Version	MC68705P3	MC68705U3	MC68705R3	–	None	None
STOP and WAIT	No	No	No	No	Yes	Yes

	Available				
	3Q81	3Q81	3Q81	4Q81	1Q82
	MC68705U3	**MC68705R3**	**MC6805T2**	**MC6805P4**	**MC146805F2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS
Number of Pins	40	40	28	28	28
On-Chip RAM (Bytes)	112	112	64	112	64
On-Chip User ROM (Bytes)	3.8K EPROM	3.8K EPROM	2.5K	1.1K	1K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	24	24	19	20	20
Unidirectional I/O Lines	8 Inputs	6 Inputs	None	None	None
Other I/O Features	Timer	Timer, A/D	Timer, PLL	Timer	Timer
EPROM Version	–	–	None	MC68705P3	None
STOP and WAIT	No	No	No	No	Yes

M6805 FAMILY NOMENCLATURE

M6805 — A FAMILY OF MICROCOMPUTERS

MC6805P2P — A PART NUMBER

— A PARTICULAR M6805 FAMILY VERSION

— A PLASTIC PACKAGE

MC68705P3 — AN EPROM VERSION

MC146805G2 — A CMOS M6805 FAMILY MEMBER

MC6805P2P2 — AN EVALUATION PROGRAM STORED IN ROM (PLASTIC PACKAGE)

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 Oscillator Input
- ☐ Crystal
- ☐ Resistor

Port A Output Drive
- ☐ CMOS and TTL
- ☐ TTL Only

Low Voltage Inhibit
- ☐ Disable
- ☐ Enable

Pattern Media (All other media requires prior factory approval.)
- ☐ EPROMs (MCM2716 or MCM2532
- ☐ EPROM MCU (MC68705U3)
- ☐ Floppy Disk
- ☐ Other ______________

MOTOROLA

MC6805U3

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

8-BIT
MICROCOMPUTER

Product Preview

8-BIT MICROCOMPUTER UNIT

The MC6805U3 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.

Since the MC6805U3 is supported by a pin compatible EPROM MCU, the MC68705U3, this offers the user an economical means of designing the MC6805U3 into his system. Prototype evaluation and pilot production can be supported by the fully compatible MC68705U3 EPROM MCU before committing high volume production to the MC6805U3 mask ROM MCU.

HARDWARE FEATURES

- 32 TTL/CMOS Compatible I/O Lines
  - 24 Bidirectional (Eight Lines are LED Compatible)
  - Eight Input Only
- 3776 Bytes of User ROM
- 112 Bytes of RAM
- Self-Check Mode
- Zero-Crossing Detect/Interrupt
- Internal 8-Bit Timer with Software Programmable 7-Bit Prescaler and Clock Source
- 5 Volt Single Supply

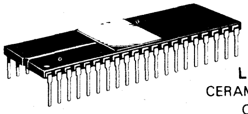
SOFTWARE FEATURES

- 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

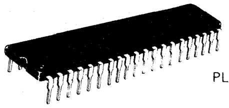
USER SELECTABLE OPTIONS

- Eight Bidirectional I/O Lines with TTL or Open-Drain Interface Option
- Crystal or Low-Cost Resistor Oscillator Option
- Low Voltage Inhibit Option
- Vector Interrupts: Timer, Software, and External

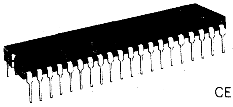
EXORciser is a registered trademark of Motorola Inc.



L SUFFIX
CERAMIC PACKAGE
CASE 715

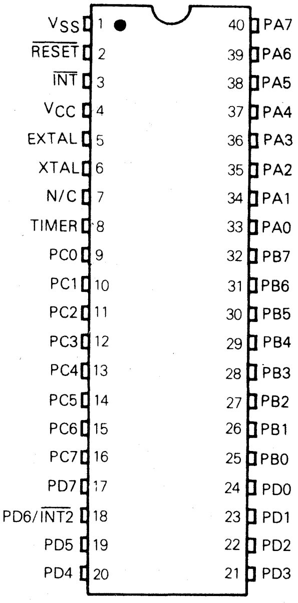


P SUFFIX
PLASTIC PACKAGE
CASE 711

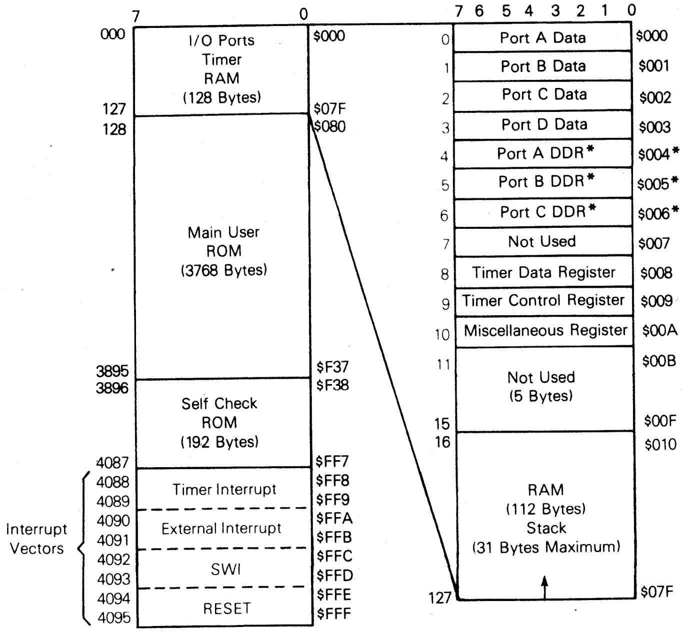


S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENTS

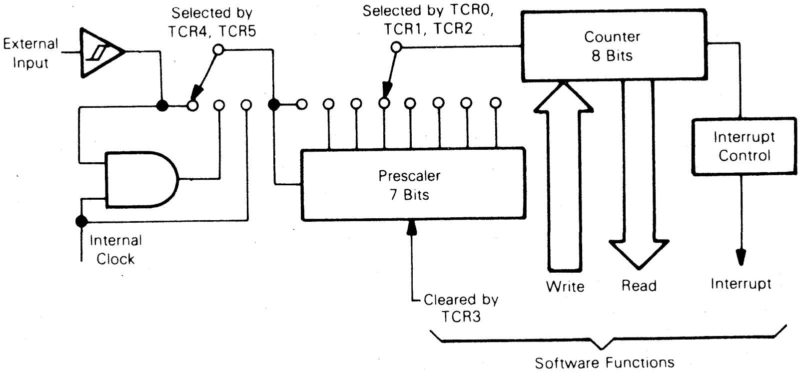


MC6803U3 ADDRESS MAP



*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

MC6805U3 TIMER BLOCK DIAGRAM



NOTES:

1. Prescaler and 8-bit counter are clocked on the falling edge of the internal clock (AS) or external input.
2. Counter is written to during data strobe (DS) and counts down continuously.

ADDITIONAL INFORMATION

For additional information please refer to the following Advance Information Data Sheets:

MC6805U2 (ADI-853-R2)	MC68705U3 (ADI-859-R1)

MOTOROLA

MC6809
(1.0 MHz)
MC68A09
(1.5 MHz)
MC68B09
(2.0 MHz)

HMOS

(HIGH DENSITY N-CHANNEL, SILICON-GATE)

8-BIT MICROPROCESSING UNIT

8-BIT MICROPROCESSING UNIT

The MC6809 is a revolutionary high-performance 8-bit microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the M6800 family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any 8-bit microprocessor today.

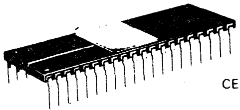
The MC6809 has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications.

MC6800 COMPATIBLE
- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES
- Two 16-bit Index Registers
- Two 16-bit Indexable Stack Pointers
- Two 8-bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

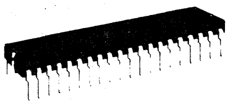
HARDWARE FEATURES
- On-Chip Oscillator (Crystal Frequency = 4XE)
- $\overline{\text{DMA/BREQ}}$ Allows DMA Operation on Memory Refresh
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- MRDY Input Extends Data Access Times for Use with Slow Memory
- Interrupt Acknowledge Output Allows Vectoring By Devices
- SYNC Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use With Slower Memories
- Early Write-Data for Dynamic Memories

SOFTWARE FEATURES
- 10 Addressing Modes
  - 6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing:
    - 0-, 5-, 8-, or 16-bit Constant Offsets
    - 8-, or 16-bit Accumulator Offsets
    - Auto-Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instructions with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address



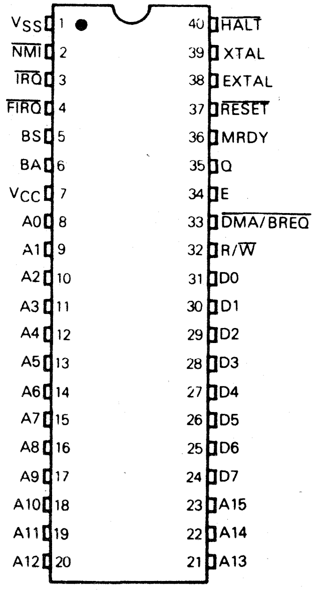
L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

FIGURE 1 — PIN ASSIGNMENT



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 MC6809, MC68A09, MC68B09 MC6809C, MC68A09C, MC68B09C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage levels (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Cerdip Plastic	θ_{JA}	 50 60 100	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

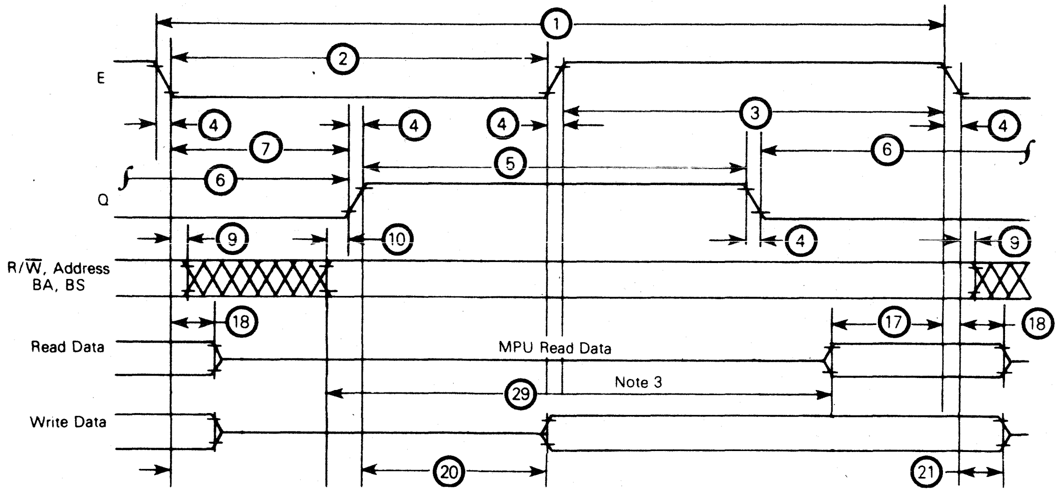
Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL $\overline{RESET}$	V_{IH} V_{IHR}	V_{SS}+2.0 V_{SS}+4.0	– –	V_{CC} V_{CC}	V
Input Low Voltage	Logic, EXTAL, $\overline{RESET}$	V_{IL}	V_{SS}−0.3	–	V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic	I_{in}	–	–	2.5	μA
DC Output High Voltage (I_{Load} = −205 μA, V_{CC} = min) (I_{Load} = −145 μA, V_{CC} = min) (I_{Load} = −100 μA, V_{CC} = min)	 D0-D7 A0-A15 R/$\overline{W}$, Q, E BA, BS	V_{OH}	 V_{SS}+2.4 V_{SS}+2.4 V_{SS}+2.4	 – – –	 – – –	V
DC Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS}+0.5	V
Internal Power Dissipation (measured at T_A = 0°C in steady state operation)		P_{INT}	–	–	1.0	W
Capacitance # (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, $\overline{RESET}$ Logic Inputs, EXTAL, XTAL	C_{in}	– –	10 10	15 15	pF
	A0-A15, R/W, BA, BS	C_{out}	–	–	15	pF
Frequency of Operation (Crystal or External Input)	MC6809 MC68A09 MC68B09	f_{XTAL}	0.4 0.4 0.4	– – –	4 6 8	MHz
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = max)	D0-D7 A0-A15, R/W	I_{TSI}	– –	2.0 –	10 100	μA

capacitances are periodically tested rather than 100% tested.

FIGURE 2 — BUS TIMING



BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

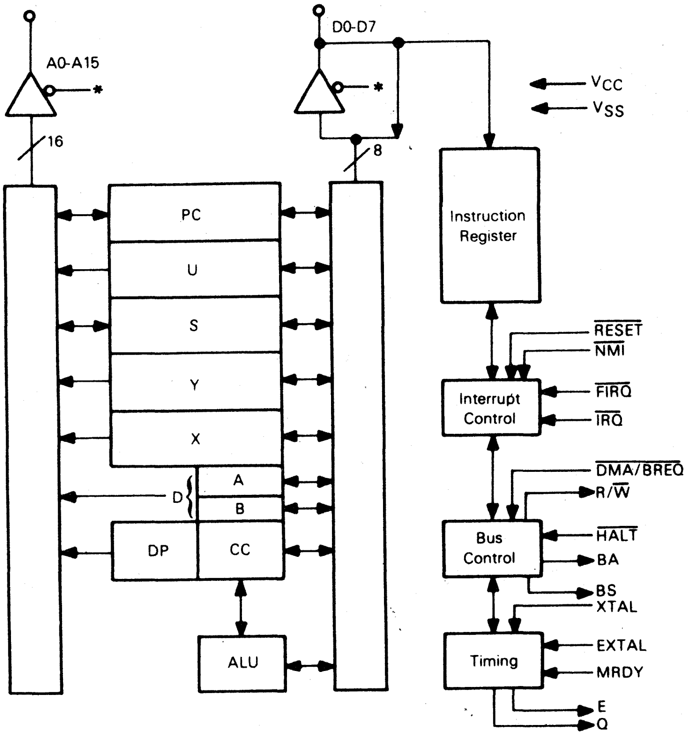
Ident. Number	Characteristics	Symbol	MC6809		MC68A09		MC68B09		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time (See Note 5)	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	5000	280	5000	210	5000	ns
3	Pulse Width, E High	PW_{EH}	450	15500	280	15700	220	15700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	430	5000	280	5000	210	5000	ns
6	Pulse Width, Q Low	PW_{QL}	450	15500	280	15700	220	15700	ns
7	Delay Time, E to Q Rise	t_{AVS}	200	250	130	165	80	125	ns
9	Address Hold Time* (See Note 4)	t_{AH}	20	–	20	–	20	–	ns
10	BA, BS, $R/\overline{W}$, and Address Valid Time to Q Rise	t_{AQ}	50	–	25	–	15	–	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time*	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time*	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time (See Note 3)	t_{ACC}	695	–	440	–	330	–	ns
	Processor Control Setup Time (MRDY, Interrupts, $\overline{DMA/BREQ}$, $\overline{HALT}$, $\overline{RESET}$) (Figures 7, 9, 10, 11, 13, and 14)	t_{PCS}	200	–	140	–	110	–	ns
	Crystal Oscillator Start Time (Figures 7 and 8)	t_{RC}	–	100	–	100	–	100	ms
	Processor Control Rise and Fall Time (Figures 7 and 9)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

*Address and data hold times are periodically tested rather than 100% tested.

NOTES:

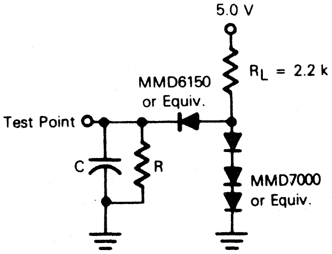
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 1 – 4 – 7 max + 10 – 17.
4. Hold time (⑨) for BA and BS is not specified.
5. Maximum t_{cyc} during MRDY or $\overline{DMA/BREQ}$ is 16 μs.

FIGURE 3 — MC6809 EXPANDED BLOCK DIAGRAM



* Internal Three-State Control

FIGURE 4 — BUS TIMING TEST LOAD



C = 30 pF for BA, BS
130 pF for D0-D7, E, Q
90 pF for A0-A15, R/$\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, E, Q, R/$\overline{W}$
24 kΩ for BA, BS

PROGRAMMING MODEL

As shown in Figure 5, the MC6809 adds three registers to the set available in the MC6800. The added registers include a Direct Page Register, the User Stack pointer and a second Index Register.

ACCUMULATORS (A, B, D)

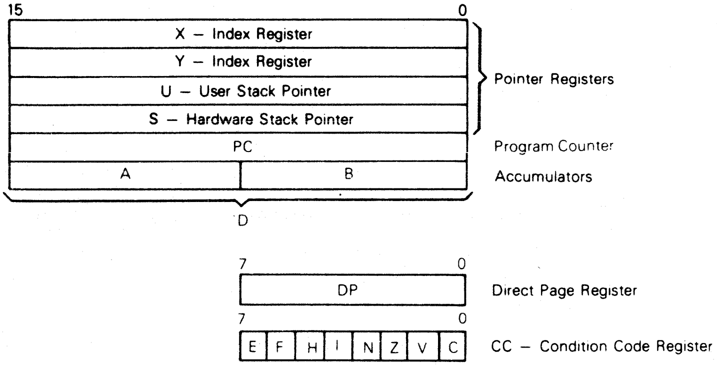
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D Register, and is formed with the A Register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The Direct Page Register of the MC6809 serves to enhance the Direct Addressing Mode. The content of this register appears at the higher address outputs (A8-A15) during direct Addressing Instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure 6800 compatibility, all bits of this register are cleared during Processor Reset.

FIGURE 5 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT



INDEX REGISTERS (X, Y)

The Index Registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculation of effective addresses. This address may be used to point to data directly or may be modifed by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented or decremented to point to the next item of tabular type data. All four pointer registers (X, Y, U, S) may be used as index registers.

STACK POINTER (U, S)

The Hardware Stack Pointer (S) is used automatically by the processor during subroutine calls and interrupts. The stack pointers of the MC6809 point to the top of the stack, in contrast to the MC6800 stack pointer, which pointed to the next free location on the stack. The User Stack Pointer (U) is controlled exclusively by the programmer thus allowing arguments to be passed to and from subroutines with ease. Both Stack Pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support **Push** and **Pull** instructions. This allows the MC6809 to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

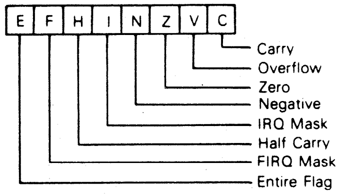
PROGRAM COUNTER

The Program Counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative Addressing is provided allowing the Program Counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The Condition Code Register defines the State of the Processor at any given time. See Figure 6.

FIGURE 6 — CONDITION CODE REGISTER FORMAT



CONDITION CODE REGISTER DESCRIPTION

BIT 0 (C)

Bit 0 is the carry flag, and is usually the carry from the binary ALU. C is also used to represent a 'borrow' from subtract like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

BIT 1 (V)

Bit 1 is the overflow flag, and is set to a one by an operation which causes a signed two's complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

BIT 2 (Z)

Bit 2 is the zero flag, and is set to a one if the result of the previous operation was identically zero.

BIT 3 (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation. Thus, a negative two's-complement result will leave N set to a one.

BIT 4 (I)

Bit 4 is the $\overline{IRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{IRQ}$ line if this bit is set to a one. $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, $\overline{RESET}$, and SWI are set I to a one; SWI2 and SWI3 do not affect I.

BIT 5 (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD). This bit is used by the DAA instruction to perform a BCD decimal add adjust operation. The state of this flag is undefined in all subtract-like instructions.

BIT 6 (F)

Bit 6 is the $\overline{FIRQ}$ mask bit. The processor will not recognize interrupts from the $\overline{FIRQ}$ line if this bit is a one. $\overline{NMI}$, $\overline{FIRQ}$, SWI, and $\overline{RESET}$ all set F to a one. $\overline{IRQ}$, SWI2 and SWI3 do not affect F.

BIT 7 (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the Condition Code Register represents past action.

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part: V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the Address Bus. When the processor does not require the bus for a data transfer, it will output address $FFFF_{16}$, R/$\overline{W}$ = 1, and BS = 0; this is a "dummy access" or $\overline{VMA}$ cycle. Addresses are valid on the rising edge of Q (see Figure 2). All address bus drivers are made high-impedance when output Bus Available (BA) is high. Each pin will drive one Schottky TTL load or four LS TTL loads, and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bi-directional data bus. Each pin will drive one Schottky TTL load or four LS TTL loads, and 130 pF.

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

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. R/$\overline{W}$ is made high impedance when BA is high. R/$\overline{W}$ is valid on the rising edge of Q.

$\overline{RESET}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 7. The Reset vectors are fetched from locations $FFFE_{16}$ and $FFFF_{16}$ (Table 1) when Interrupt Acknowledge is true, ($\overline{BA}$ • BS = 1). During initial power-on, the $\overline{RESET}$ line should be held low until the clock oscillator is fully operational. See Figure 8.

Because the MC6809 $\overline{RESET}$ pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage ensures that all peripherals are out of the reset state before the Processor.

$\overline{HALT}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the Halt or Bus Grant state. While halted, the MPU will not respond to external real-time requests ($\overline{FIRQ}$, $\overline{IRQ}$) although $\overline{DMA/BREQ}$ will always be accepted, and $\overline{NMI}$ or $\overline{RESET}$ will be latched for later response. During the Halt state Q and E continue to run normally. If the MPU is not running ($\overline{RESET}$, $\overline{DMA/BREQ}$), a halted state (BA•BS = 1) can be achieved by pulling $\overline{HALT}$ low while $\overline{RESET}$ is still low. If $\overline{DMA/BREQ}$ and $\overline{HALT}$ are both pulled low, the processor will reach the last cycle of the instruction (by reverse cycle stealing) where the machine will then become halted. See Figure 9.

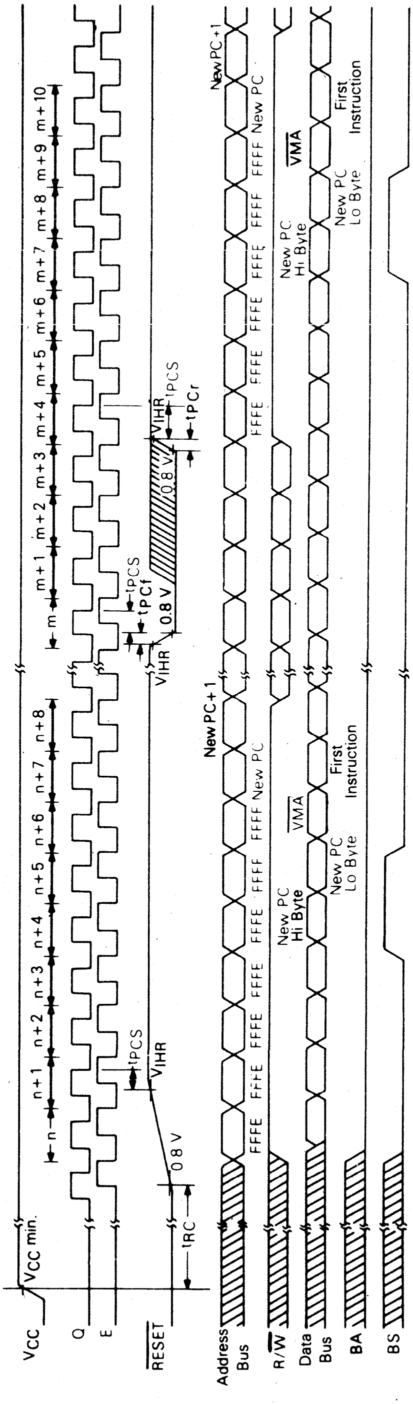
BUS AVAILABLE, BUS STATUS (BA, BS)

The Bus Available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. This signal does not imply that the bus will be available for more than one cycle. When BA goes low, a dead cycle will elapse before the MPU acquires the bus.

The Bus Status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q).

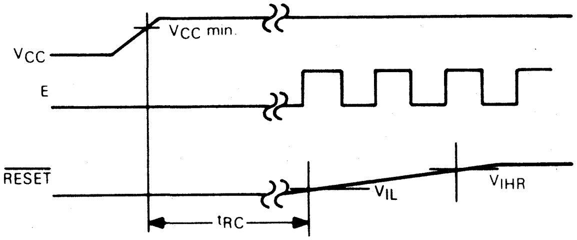
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt or Bus Grant Acknowledge

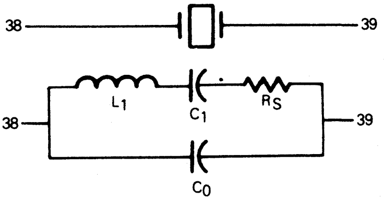
FIGURE 7 — $\overline{\text{RESET}}$ TIMING

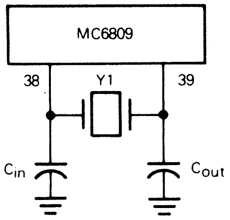


*NOTE: Parts with date codes prefixed by 7F or 5A will come out of $\overline{\text{RESET}}$ one cycle sooner than shown.

FIGURE 8 — CRYSTAL CONNECTIONS AND OSCILLATOR START UP







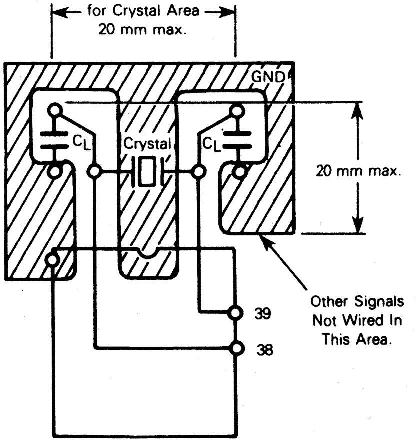
Y1	C_{in}	C_{out}
8 MHz	18 pF	18 pF
6 MHz	20 pF	20 pF
4 MHz	24 pF	24 pF

Nominal Crystal Parameters*

	3.58 MHz	4.00 MHz	6.0 MHz	8.0 MHz
RS	60 Ω	50 Ω	30-50 Ω	20-40 Ω
C_0	3.5 pF	6.5 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
Q	>40K	>30 K	>20 K	>20 K

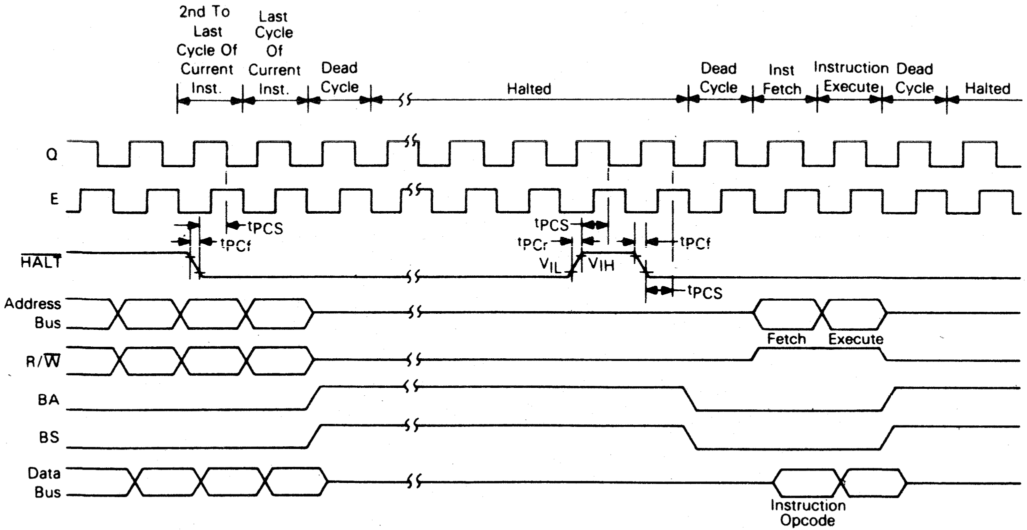
All parameters are 10%

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



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

FIGURE 9 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

Interrupt Acknowledge is indicated during both cycles of a hardware-vector-fetch ($\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, SWI, SWI2, SWI3). This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device. See Table 1.

Sync Acknowledge is indicated while the MPU is waiting for external synchronization on an interrupt line.

Halt/Bus Grant is true when the MC6809 is in a Halt or Bus Grant condition.

TABLE 1: MEMORY MAP FOR INTERRUPT VECTORS

Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{\text{IRQ}}$
FFF6	FFF7	$\overline{\text{FIRQ}}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

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

A negative transition on this input requests that a non-maskable interrupt sequence be generated. A non-maskable interrupt cannot be inhibited by the program, and also has a higher priority than $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$ or software interrupts. During recognition of an $\overline{\text{NMI}}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{\text{NMI}}$ will not be recognized until the first program load of the Hardware Stack Pointer (S). The pulse width of $\overline{\text{NMI}}$ low must be at least one E cycle. If the $\overline{\text{NMI}}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 10.

FAST-INTERRUPT REQUEST ($\overline{\text{FIRQ}}$)*

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard Interrupt Request ($\overline{\text{IRQ}}$), and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 11.

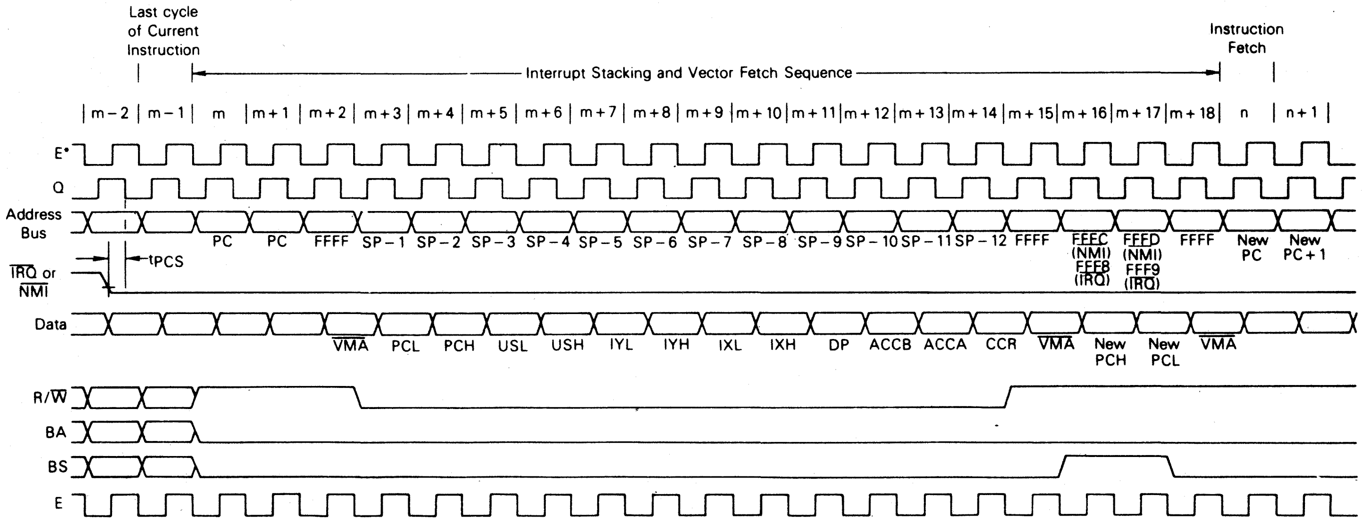
INTERRUPT REQUEST ($\overline{\text{IRQ}}$)*

A low level input on this pin will initiate an Interrupt Request sequence provided the mask bit (I) in the CC is clear. Since $\overline{\text{IRQ}}$ stacks the entire machine state it provides a slower response to interrupts than $\overline{\text{FIRQ}}$. $\overline{\text{IRQ}}$ also has a lower priority than $\overline{\text{FIRQ}}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 10.

---

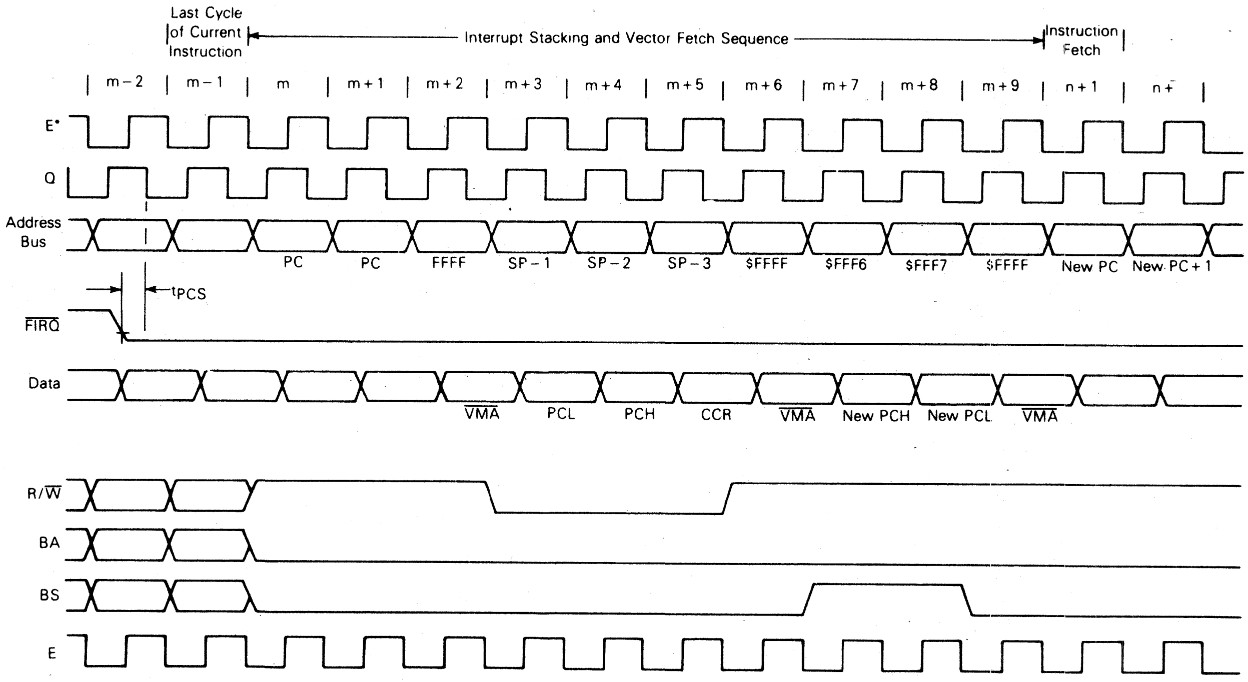
*$\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, and $\overline{\text{IRQ}}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or CWAI condition is present. If $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ do not remain low until completion of the current instruction they may not be recognized. However, NMI is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{\text{RESET}}$ and the rising edge of BS indicating $\overline{\text{RESET}}$ acknowledge.

FIGURE 10 — $\overline{\text{IRQ}}$ AND $\overline{\text{NMI}}$ INTERRUPT TIMING



NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2.0 V and logic low = 0.8 V unless otherwise specified.
E clock shown for reference only.

FIGURE 11 — FIRQ INTERRUPT TIMING



NOTE: Waveform measurements for all inputs and outputs are specified at logic high = 2.0 V and logic low = 0.8 V unless otherwise specified.
E clock shown for reference only.

XTAL, EXTAL

These inputs are used to connect the on-chip oscillator to an external parallel-resonant crystal. Alternately, the pin EXTAL may be used as a TTL level input for external timing by grounding XTAL. The crystal or external frequency is four times the bus frequency. See Figure 8. Proper RF layout techniques should be observed in the layout of printed circuit boards.

E, Q

E is similar to the MC6800 bus timing signal $\phi 2$; Q is a quadrature clock signal which leads E. Q has no parallel on the MC6800. Addresses from the MPU will be valid with the leading edge of Q. Data is latched on the falling edge of E. Timing for E and Q is shown in Figure 12.

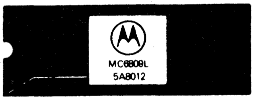
MRDY*

This input control signal allows stretching of E and Q to extend data-access time. E and Q operate normally while MRDY is high. When MRDY is low, E and Q may be stretched in integral multiples of quarter (¼) bus cycles, thus allowing interface to slow memories, as shown in Figure 13(A). During non-valid memory access (VMA cycles) MRDY has no effect on stretching E and Q; this inhibits slowing the processor during "don't care" bus accesses. MRDY may also be used to stretch clocks (for slow memory) when bus control has been transferred to an external device (through the use of HALT and DMA/BREQ).

NOTE: Four of the early production mask sets (G7F, T5A, P6F, T6M) require synchronization of the MRDY input with the 4f clock. The synchronization necessitates an external oscillator as shown in Figure 13(B). The negative transition of the MRDY signal, normally derived from the chip select decoding, must meet the t_{PCS} timing. With these four mask sets, MRDY's positive transition must occur with the rising edge of 4f.

In addition, on these same mask sets, MRDY will not stretch the E and Q signals if the machine is executing either a TFR or EXG instruction during the HALT high-to-low transition. If the MPU executes a CWAI instruction, the machine pushes the internal registers onto the stack and then awaits an interrupt. During this waiting period, it is possible to place the MPU into a Halt mode to three-state the machine, but MRDY will not stretch the clocks.

The mask set for a particular part may be determined by examining the markings on top of the part. Below the part number is a string of characters. The first two characters are the last two characters of the mask set code. If there are only four digits the part is the G7F mask set. The last four digits, the date code, show when the part was manufactured. These four digits represent year and week. For example a ceramic part marked:



is a T5A mask set made the twelveth week of 1980.

DMA/BREQ*

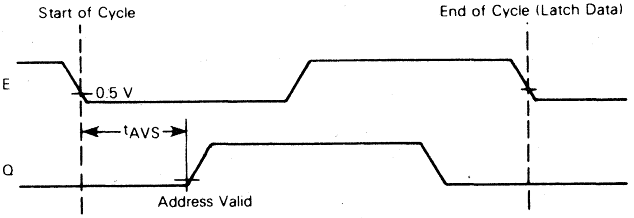
The DMA/BREQ input provides a method of suspending execution and acquiring the MPU bus for another use, as shown in Figure 14. Typical uses include DMA and dynamic memory refresh.

Transitions of DMA/BREQ should occur during Q. A low level on this pin will stop instruction execution at the end of the current cycle unless pre-empted by self-refresh. The MPU will acknowledge DMA/BREQ by setting BA and BS to a one. The requesting device will now have up to 15 bus cycles before the MPU retrieves the bus for self-refresh. Self-refresh requires one bus cycle with a leading and trailing dead cycle. See Figure 15. The self-refresh counter is only cleared if DMA/BREQ is inactive for two or more MPU cycles.

Typically, the DMA controller will request to use the bus by asserting DMA/BREQ pin low on the leading edge of E. When the MPU replies by setting BA and BS to a one, that cycle will be a dead cycle used to transfer bus mastership to the DMA controller.

False memory accesses may be prevented during any dead cycles by developing a system DMAVMA signal which is LOW in any cycle when BA has changed.

FIGURE 12 — E/Q RELATIONSHIP



NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

---

*The on-board clock generator furnishes E and Q to both the system and the MPU. When MRDY is pulled low, both the system clocks and the internal MPU clocks are stretched. Assertion of DMA/BREQ input stops the internal MPU clocks while allowing the external system clocks to RUN (i.e., release the bus to a DMA controller). The internal MPU clocks resume operation after DMA/BREQ is released or after 16 bus cycles (14 DMA, 2 dead), whichever occurs first. While DMA/BREQ is asserted it is sometimes necessary to pull MRDY low to allow DMA to/from slow memory/peripherals. As both MRDY and DMA/BREQ control the internal MPU clocks, care must be exercised not to violate the maximum t_{cyc} specification for MRDY or DMA/BREQ. (See Note 5 in Bus Timing.)

When BA goes low (either as a result of $\overline{\text{DMA/BREQ}}$ = HIGH or MPU self-refresh), the DMA device should be taken off the bus. Another dead cycle will elapse before the MPU accesses memory, to allow transfer of bus mastership without contention.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{\text{RESET}}$ and is repeated indefinitely unless altered by a special instruction or hardware occurrence. software instructions that alter normal MPU operation are: SWI, SWI2, SWI3, CWAI, RTI and SYNC. An interrupt, $\overline{\text{HALT}}$, or $\overline{\text{DMA/BREQ}}$ can also alter the normal execution of instructions. Figure 16 illustrates the flowchart for the MC6809.

FIGURE 13(A) — MRDY TIMING

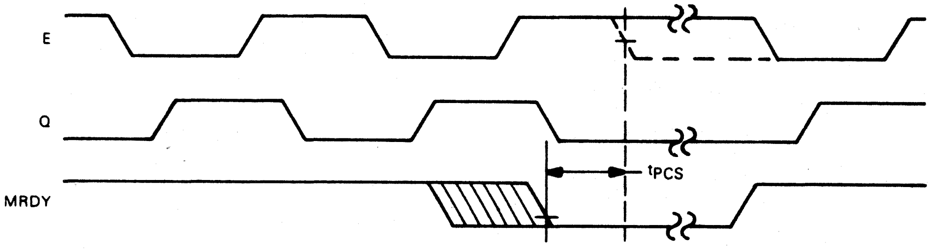


FIGURE 13(B) — MRDY SYNCHRONIZATION

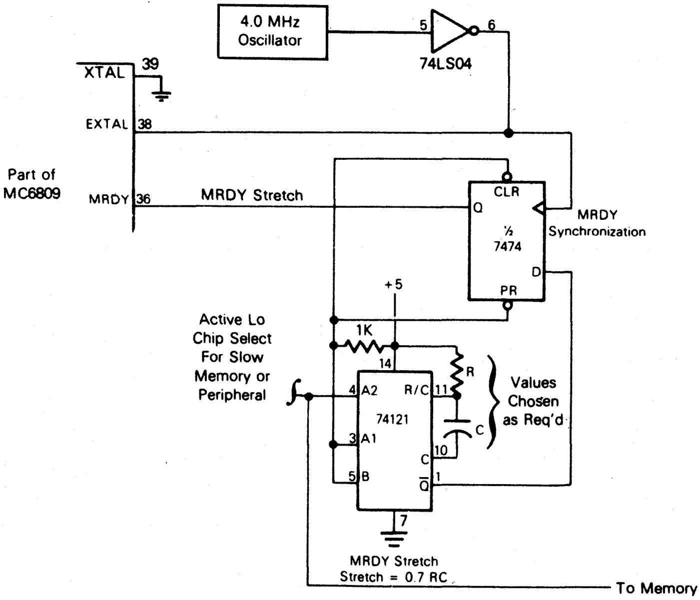


FIGURE 14 — TYPICAL DMA TIMING (< 14 CYCLES)

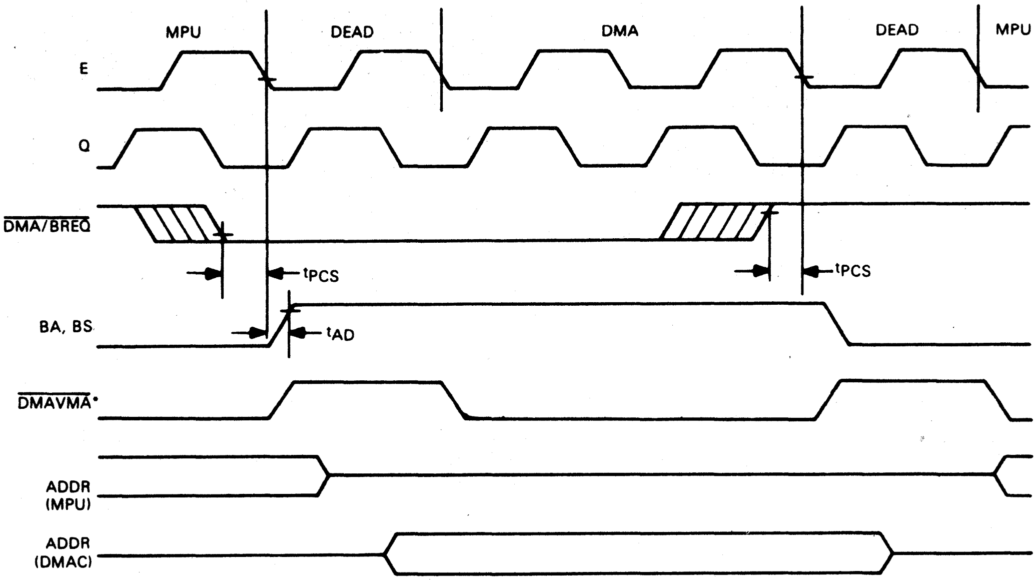
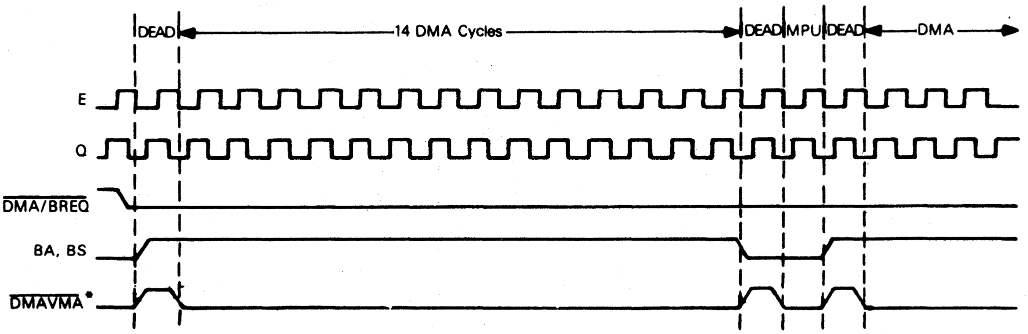


FIGURE 15 — AUTO-REFRESH DMA TIMING (> 14 CYCLES) (REVERSE CYCLE STEALING)



*$\overline{\text{DMAVMA}}$ is a signal which is developed externally, but is a system requirement for DMA. Refer to Application Note AN-8.

NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

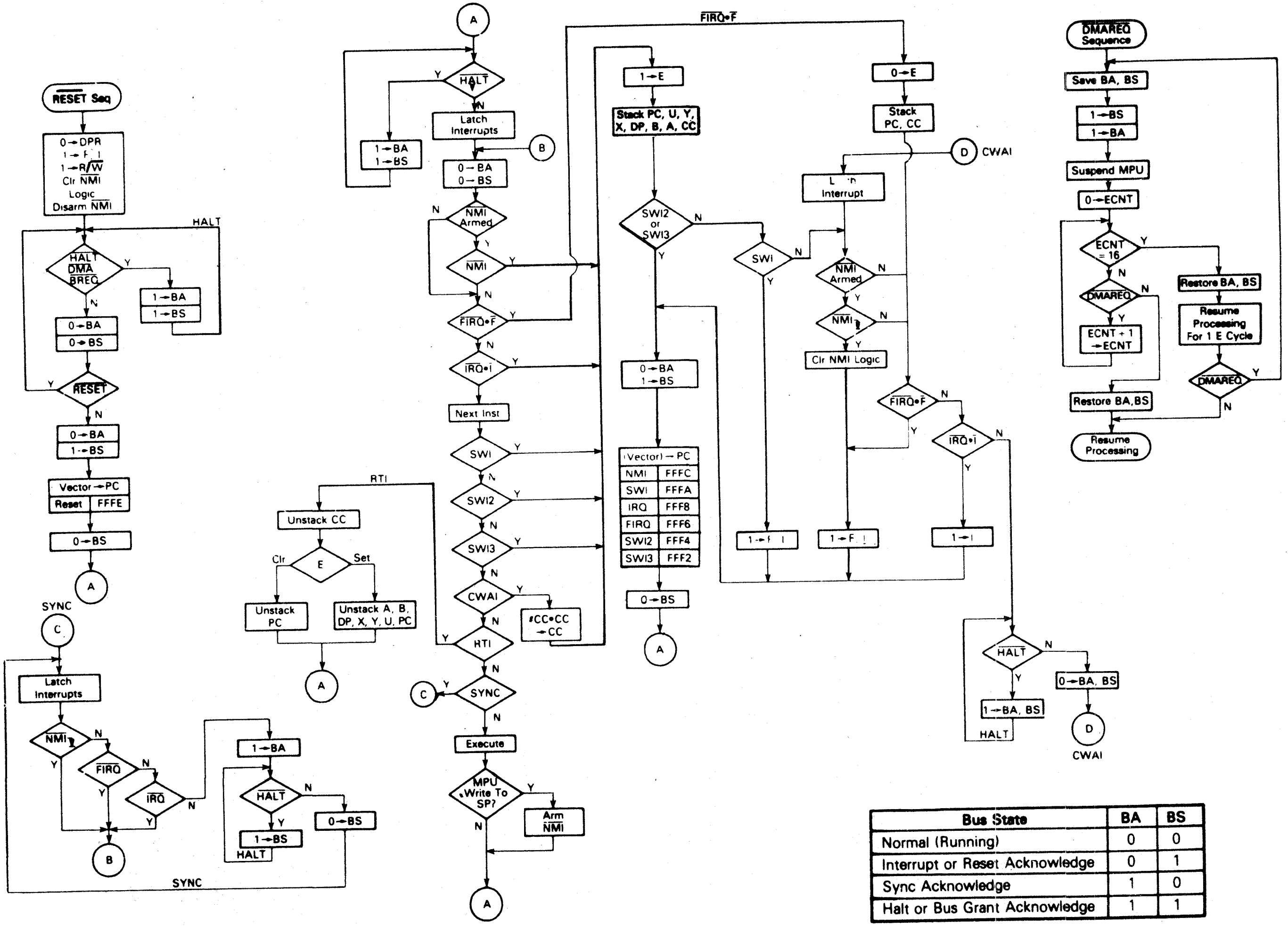


FIGURE 16 — FLOWCHART FOR MC6809 INSTRUCTIONS

Bus State	BA	BS
Normal (Running)	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt or Bus Grant Acknowledge	1	1

NOTE: Asserting $\overline{\text{RESET}}$ will result in entering the reset sequence from any point in the flow chart.

ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809 has the most complete set of addressing modes available on any microcomputer today. For example, the MC6809 has 59 basic instructions; however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the MC6809:

- Inherent (Includes Accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary. Examples of Inherent Addressing are: ABX, DAA, SWI, ASRA, and CLRB.

IMMEDIATE ADDRESSING

In Immediate Addressing, the effective address of the data is the location immediately following the opcode (i.e., the data to be used in the instruction immediately follows the opcode of the instruction). The MC6809 uses both 8 and 16-bit immediate values depending on the size of argument specified by the opcode. Exampies of instructions with Immediate Addressing are:

```
LDA #$20
LDX #$F000
LDY #CAT
```

NOTE: # signifies Immediate addressing, $ signifies hexadecimal value.

EXTENDED ADDRESSING

In Extended Addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent. Examples of Extended Addressing include:

```
LDA CAT
STX MOUSE
LDD $2000
```

EXTENDED INDIRECT

As a special case of indexed addressing (discussed below), one level of indirection may be added to Extended Addressing. In Extended Indirect, the two bytes following the postbyte of an Indexed instruction contain the address of the data.

```
LDA [CAT]
LDX [$FFFE]
STU [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower 8 bits of the address to be used. The upper 8 bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register. Since the DP register is set to $00 on Reset, direct addressing on the MC6809 is compatible with direct addressing on the M6800. Indirection is not allowed in direct addressing. Some examples of direct addressing are:

```
LDA    $30
SETDP $10 (Assembler directive)
LDB    $1030
LDD    < CAT
```

NOTE: < is an assembler directive which forces direct addressing.

REGISTER ADDRESSING

Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction. This is called a postbyte. Some examples of register addressing are:

TFR	X, Y	Transfers X into Y
EXG	A, B	Exchanges A with B
PSHS	A, B, X, Y	Push Y, X, B and A onto S
PULU	X, Y, D	Pull D, X, and Y from U

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode as well as the pointer register to be used. Figure 17 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation.

FIGURE 17 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit								Indexed Addressing Mode
7	6	5	4	3	2	1	0	
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R++
1	R	R	0	0	0	1	0	,−R
1	R	R	i	0	0	1	1	,−−R
1	R	R	i	0	1	0	0	EA = ,R + 0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R + D Offset
1	x	x	i	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]

Addressing Mode Field

Indirect Field
(Sign bit when $b_7 = 0$)

Register Field: RR
00 = X
01 = Y
10 = U
11 = S

x = Don't Care
d = Offset Bit
i = 0 = Not Indirect
1 = Indirect

Zero-Offset Indexed — In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

```
LDD    0,X
LDA    S
```

Constant Offset Indexed — In this mode, a two's-complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offsets are available:

5 -bit (−16 to +15)
8 -bit (−128 to +127)
16-bit (−32768 to +32767)

The two's complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles. The two's complement 8-bit offset is contained in a single byte following the postbyte. The two's complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be concerned with the size of this offset since the assembler will select the optimal size automatically.

Examples of constant-offset indexing are:

```
LDA    23,X
LDX    -2,S
LDY    300,X
LDU    CAT,Y
```

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect Assembler Form	Non Indirect Postbyte OP Code	+ ~	+ #	Indirect Assembler Form	Indirect Postbyte OP Code	+ ~	+ #
Constant Offset From R	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
(2's Complement Offsets)	5 Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8 Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16 Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
(2's Complement Offsets)	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R++	1RR00001	3	0	[,R++]	1RR10001	6	0
	Decrement By 1	,−R	1RR00010	2	0	not allowed			
	Decrement By 2	,−−R	1RR00011	3	0	[,−−R]	1RR10011	6	0
Constant Offset From PC	8 Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
(2's Complement Offsets)	16 Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16 Bit Address	—	—	—	—	[n]	10011111	5	2

R = X, Y, U or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

$\overset{+}{\sim}$ and $\overset{+}{\#}$ indicate the number of additional cycles and bytes for the particular variation.

Accumulator-Offset Indexed – This mode is similar to constant offset indexed except that the two's-complement value in one of the accumulators (A, B or D) and the contents of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are:

```
    LDA     B,Y
    LDX     D,Y
    LEAX    B,X
```

Auto Increment/Decrement Indexed – In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or for the creation of software stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment; but the tables, etc., are scanned from the high to low addresses. The size of the increment/decrement can be either one or two to allow for tables of either 8 or 16-bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allow them to be used to create additional software stacks that behave identically to the U and S stacks.

Some examples of the auto increment/decrement addressing modes are:

```
    LDA     ,X+
    STD     ,Y++
    LDB     ,-Y
    LDX     ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address.

Consider the following instruction:

```
            STX 0,X++ (X initialized to 0)
```

The desired result is to store a 0 in locations $0000 and $0001 then increment X to point to $0002. In reality, the following occurs:

0→temp	calculate the EA; temp is a holding register
X+2→X	perform autoincrement
X→(temp)	do store operation

INDEXED INDIRECT

All of the indexing modes with the exception of auto increment/decrement by one, or a ±4-bit offset may have an additional level of indirection specified. In indirect adddressing, the effective address is contained at the location specified by the contents of the Index register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the Index register and an offset.

```
        Before Execution
        A=XX (don't care)
        X=$F000
$0100   LDA  [$10,X]          EA is now $F010

$F010   $F1                   $F150 is now the
$F011   $50                   new EA

$F150   $AA
        After Execution
        A=$AA Actual Data Loaded
        X=$F000
```

All modes of indexed indirect are included except those which are meaningless (e.g., auto increment/decrement by 1 indirect). Some examples of indexed indirect are:

```
    LDA    [,X]        LDA    [B,Y]
    LDD    [10,S]      LDD    [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter. If the branch condition is true then the calculated address (PC + signed offset) is loaded into the program counter. Program execution continues at the new location as indicated by the PC; short (1 byte offset) and long (2 bytes offset) relative addressing modes are available. All of memory can be reached in long relative addressing as an effective address is interpreted modulo 2^{16}. Some examples of relative addressing are:

```
            BEQ     CAT       (short)
            BGT     DOG       (short)
    CAT     LBEQ    RAT       (long)
    DOG     LBGT    RABBIT    (long)
             •
             •
             •
    RAT     NOP
    RABBIT  NOP
```

PROGRAM COUNTER RELATIVE

The PC can be used as the pointer register with 8 or 16-bit signed offsets. As in relative addressing, the offset is added to the current PC to create the effective address. The effective address is then used as the address of the operand or data. Program Counter Relative Addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the Program Counter. Examples are:

```
    LDA     CAT, PCR
    LEAX    TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available.

```
    LDA     [CAT, PCR]
    LDU     [DOG, PCR]
```

MC6809 INSTRUCTION SET

The instruction set of the MC6809 is similar to that of the MC6800 and is upward compatible at the source code level. The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464.

Some of the new instructions are described in detail below:

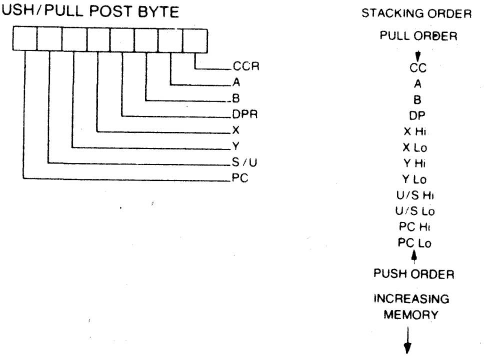
PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register, or set of registers with a single instruction.

PULU/PULS

The pull instructions have the same capability of the push instruction, in reverse order. The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled. The actual PUSH/PULL sequence is fixed; each bit defines a unique register to push or pull, as shown below.

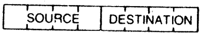
PUSH/PULL POST BYTE



TFR/EXG

Within the MC6809, any register may be transferred to or exchanged with another of like-size; i.e., 8-bit to 8-bit or 16-bit to 16-bit. Bits 4-7 of postbyte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

TRANSFER/EXCHANGE POST BYTE



REGISTER FIELD

0000	D (A:B)	1000	A
0001	X	1001	B
0010	Y	1010	CCR
0011	U	1011	DPR
0100	S		
0101	PC		

NOTE: All other combinations are undefined and INVALID.

LEAX/LEAY/LEAU/LEAS

The LEA (Load Effective Address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register. This makes all the features of the internal addressing hardware available to the programmer. Some of the implications of this instruction are illustrated in Table 3.

The LEA instruction also allows the user to access data and tables in a position independent manner. For example:

```
                LEAX    MSG1, PCR
                LBSR    PDATA (Print message routine)
                •
                •
MSG1            FCC     'MESSAGE'
```

This sample program prints: 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1. This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located, when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the autoincrement and autodecrement addressing modes due to the sequence of internal operations. The LEA internal sequence is outlined as follows:

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U, or S may be substituted for a and b.)

1. b→temp (calculate the EA)
2. b+1→b (modify b, postincrement)
3. temp→a (load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-bit constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-bit constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-bit A accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-bit D accumulator to Y
LEAU − 10, U	U − 10 → U	Subtracts 10 from U
LEAS − 10, S	S − 10 → S	Used to reserve area on stack
LEAS 10, S	S + 10 → S	Used to 'clean up' stack
LEAX 5, S	S + 5 → X	Transfers as well as adds

LEAa , − b

1. b − 1 → temp (calculate EA with predecrement)
2. b − 1 → b (modify b, predecrement)
3. temp → a (load a)

Autoincrement-by-two and autodecrement-by-two instructions work similarly. Note that LEAX ,X + does not change X, however LEAX, − X does decrement LEAX 1, X should be used to increment X by one.

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator. This unsigned multiply also allows multiple-precision multiplications.

Long And Short Relative Branches

The MC6809E has the capability of program counter relative branching throughout the entire memory map. In this mode, if the branch is to be taken, the 8- or 16-bit signed offset is added to the value of the program counter to be used as the effective address. This allows the program to branch anywhere in the 64K memory map. Position-independent code can be easily generated through the use of relative branching. Both short (8-bit) and long (16-bit) branches are available.

SYNC

After encountering a Sync instruction, the MPU enters a Sync state, stops processing instructions and waits for an interrupt. If the pending interrupt is non-maskable ($\overline{NMI}$) or maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) clear, the processor will clear the Sync state and perform the normal interrupt stacking and service routine. Since $\overline{FIRQ}$ and $\overline{IRQ}$ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken. If the pending interrupt is maskable ($\overline{FIRQ}$, $\overline{IRQ}$) with its mask bit (F or I) set, the processor will clear the Sync state and continue processing by executing the next inline instruction. Figure 18 depicts Sync timing.

Software Interrupts

A Software Interrupt is an instruction which will cause an interrupt, and its associated vector fetch. These Software Interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems. Three levels of SWI are available on the MC6809, and are prioritized in the following order: SWI, SWI2, SWI3.

16-Bit Operation

The MC6809 has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes and pulls.

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the MC6809. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flowchart. VMA is an indication of $FFFF_{16}$ on the address bus, $R/\overline{W}$ = 1 and BS = 0. The following examples illustrate the use of the chart; see Figure 19.

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                     •
                     •
                     •
$8000                LBSR     CAT
                     •
                     •
                     •
$A000       CAT      •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	$R/\overline{W}$	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	VMA Cycle
5	FFFF	*	1	VMA Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	VMA Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000       DEC      $A000
            •
            •
            •
$A8000      $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	$R/\overline{W}$	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	VMA Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	VMA Cycle
7	A000	7F	0	Store the Decremented Data

*The data bus has the data at that particular address.

MC6809 INSTRUCTION SET TABLES

The instructions of the MC6809 have been broken down into five different categories. They are as follows:

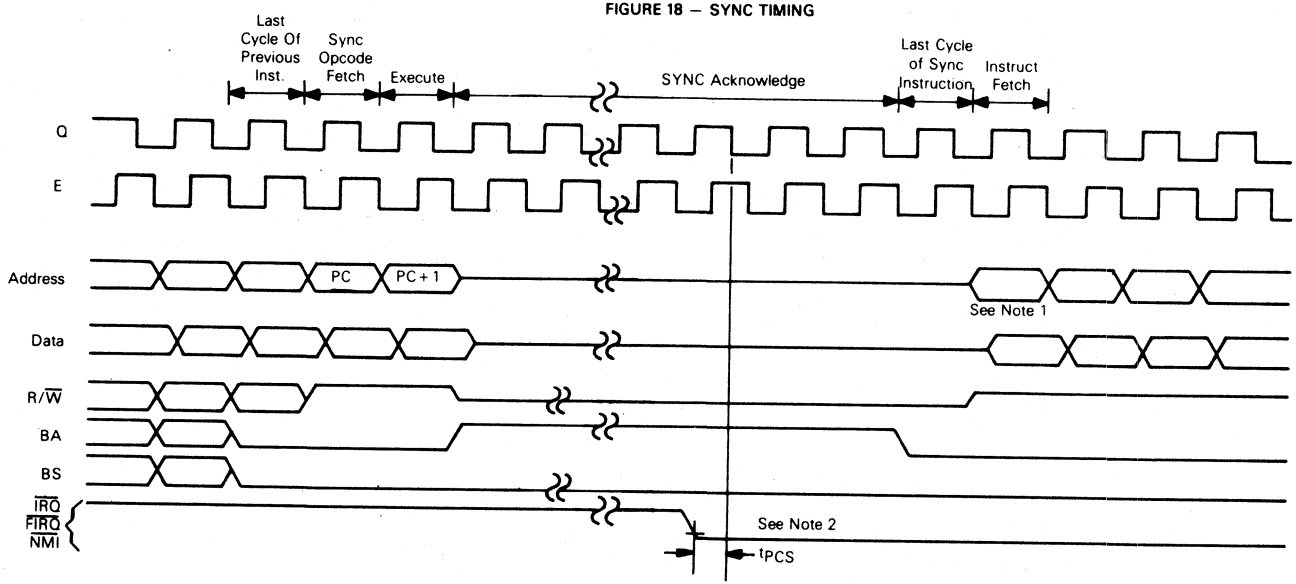
8-Bit operation (Table 4)
16-Bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 21 contains a compilation of data that will assist in programming the MC6809.

FIGURE 18 — SYNC TIMING

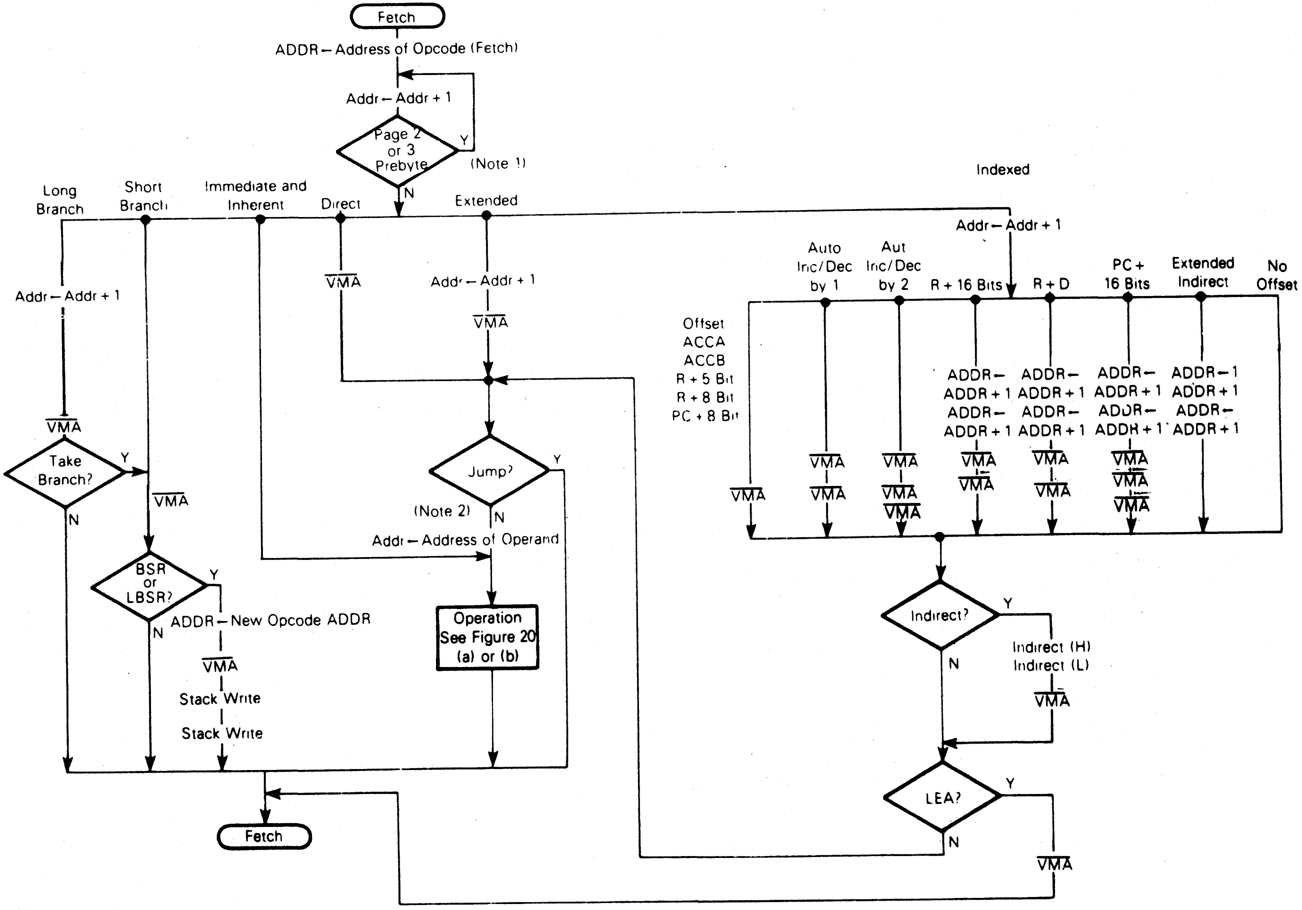


NOTES:

1. If the associated mask bit is set when the interrupt is requested, this cycle will be an instruction fetch from address location PC+1. However, if the interrupt is accepted ($\overline{\text{NMI}}$ or an unmasked $\overline{\text{FIRQ}}$ or $\overline{\text{IRQ}}$) interrupt processing continues with this cycle as (m) on Figures 10 and 11 (Interrupt Timing).
2. If mask bits are clear, $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ must be held low for three cycles to guarantee interrupt to be taken, although only one cycle is necessary to bring the processor out of SYNC.

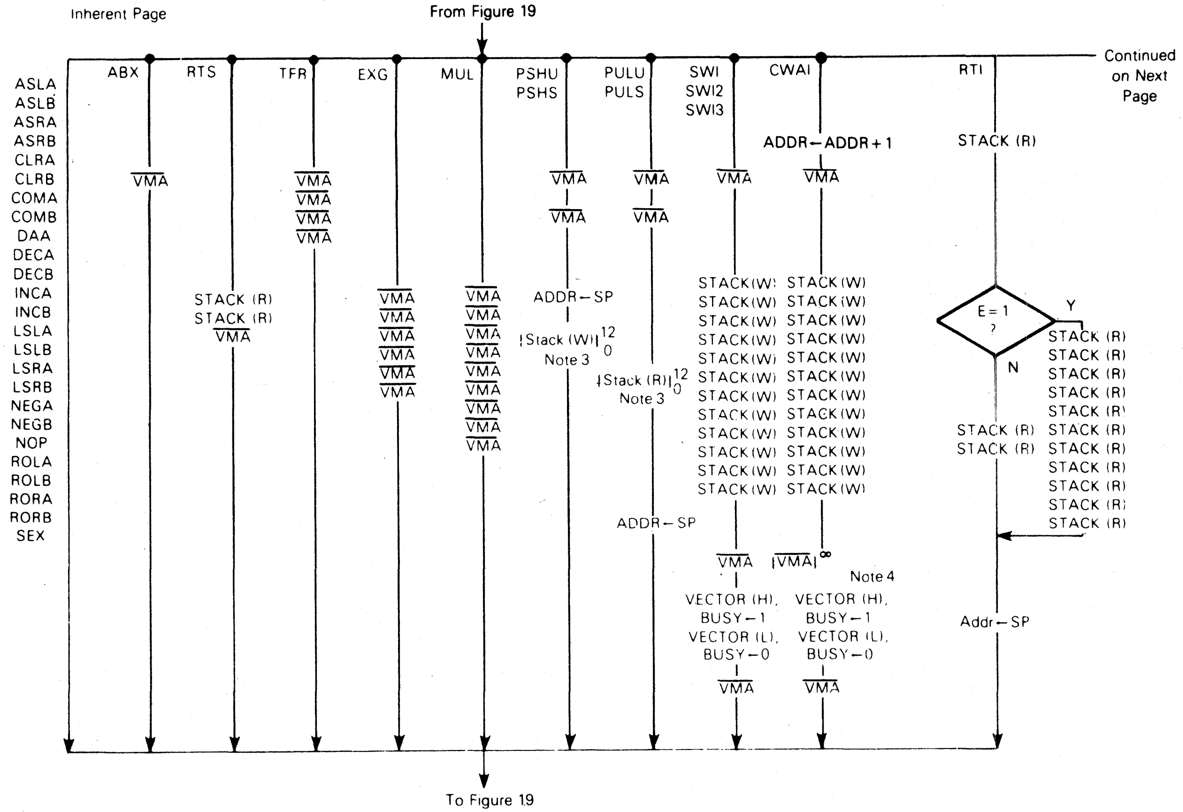
NOTE: Waveform measurements for all inputs and outputs are specified at logic high 2.0 V and logic low 0.8 V unless otherwise specified.

FIGURE 19 — ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE



NOTES:
1. All subsequent Page 2 and Page 3 pre-bytes will be ignored after initial opcode fetch
2. Write operation during store instruction.
3. ADDR refers to the state of the address bus.

FIGURE 20(a) — OPERATION: ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE



NOTES:

1. Stack (W) refers to the following sequence: SP←SP − 1, then ADDR←SP with $R/\overline{W}=0$
   Stack (R) refers to the following sequence: ADDR←SP with $R/\overline{W}=1$, then SP←SP + 1.
   PSHU, PULU instructions use the user stack pointer (i.e., SP ≡ U) and PSHS, PULS use the hardware stack pointer (i.e., SP ≡ S).
2. Vector refers to the address of an interrupt or reset vector (see Table 1).
3. The number of stack accesses will vary according to the number of bytes saved.
4. $\overline{VMA}$ cycles will occur until an interrupt occurs.

FIGURE 20(b) — OPERATION: ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE (CONTINUED)

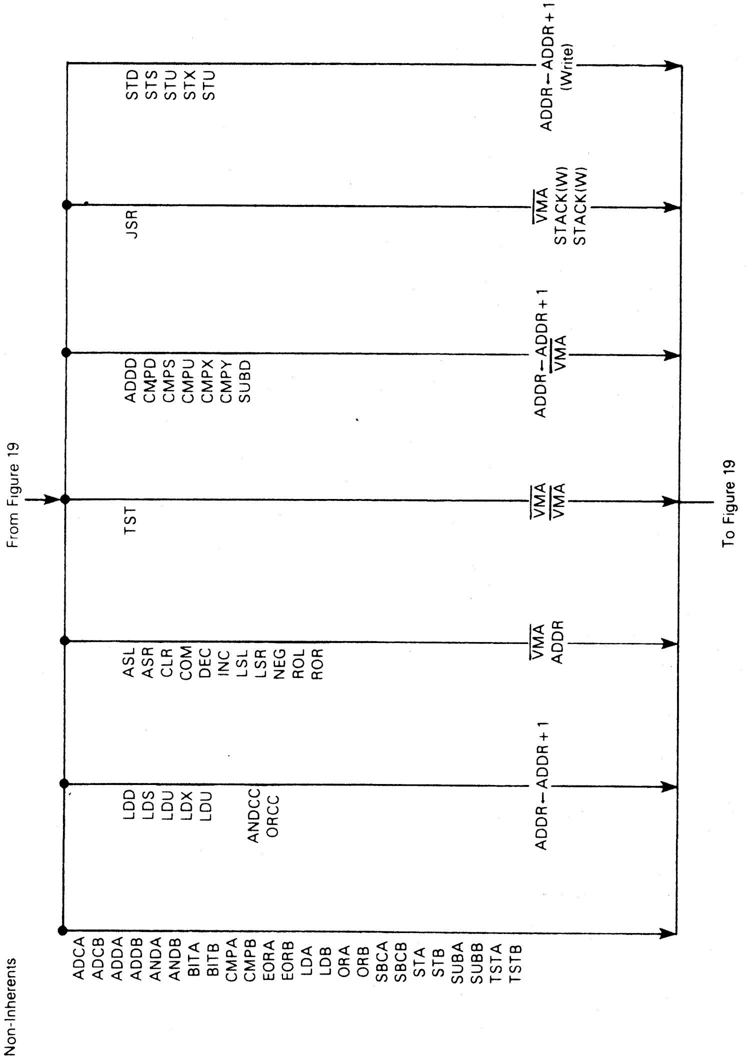


TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE: A, B, CC or DP may be pushed to (pulled from) either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U or PC
TFR R, D	Transfer X, Y, S, U or PC to D

NOTE: D may be pushed (pulled) to either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, S, U or PC with D, X, Y, S, U or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 — BRANCH INSTRUCTIONS

Instruction	Description
SIMPLE BRANCHES	
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
SIGNED BRANCHES	
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2's complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2's complement result
BLT, LBLT	Branch if less than (signed)
UNSIGNED BRANCHES	
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
OTHER BRANCHES	
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 — MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	•				31	LEAY		4+	2+	61	•			
02	•				32	LEAS		4+	2+	62	•			
03	COM		6	2	33	LEAU	Indexed	4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Inherent	5+	2	64	LSR		6+	2+
05	•				35	PULS		5+	2	65	•			
06	ROR		6	2	36	PSHU		5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU		5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	•				68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS		5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX		3	1	6A	DEC		6+	2+
0B	•				3B	RTI		6/15	1	6B	•			
0C	INC		6	2	3C	CWAI		≥20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL		11	1	6D	TST		6+	2+
0E	JMP		3	2	3E	•				6E	JMP		3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	–	–	–	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	–	–	–	41	•				71	•			
12	NOP	Inherent	2	1	42	•				72	•			
13	SYNC	Inherent	≥4	1	43	COMA		2	1	73	COM		7	3
14	•				44	LSRA		2	1	74	LSR		7	3
15	•				45	•				75	•			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	•				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	•	–			4B	•				7B	•			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG		8	2	4E	•				7E	JMP		4	3
1F	TFR	Inherent	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN		3	2	51	•				81	CMPA		2	2
22	BHI		3	2	52	•				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	•				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	•			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	•				8B	ADDA		2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT		3	2	5E	•				8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	•			

LEGEND:

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)
# Number of program bytes
• Denotes unused opcode

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

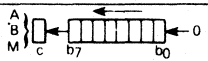
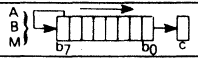
NOTE All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	•			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	•			
CE	LDU	Immed	3	3
CF	•			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

Page 2 and 3 Machine Codes

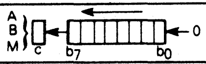
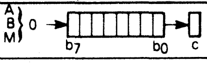
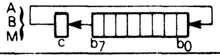
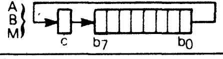
OP	Mnem	Mode	~	#
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

FIGURE 21 — PROGRAMMING AID

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0
		Immediate			Direct			Indexed			Extended			Inherent				H	N	Z	V	C
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#						
ABX														3A	3	1	B + X → X (Unsigned)	●	●	●	●	●
ADC	ADCA	89	2	2	99	4	2	A9	4+	2+	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4+	2+	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4+	2+	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4+	2+	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6+	2+	F3	7	3				D + M:M + 1 → D	●	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4+	2+	B4	5	3				A Λ M → A	●	↕	↕	0	●
	ANDB	C4	2	2	D4	4	2	E4	4+	2+	F4	5	3				B Λ M → B	●	↕	↕	0	●
	ANDCC	1C	3	2													CC Λ IMM → CC					7
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6+	2+	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	●	↕
	ASRB													57	2	1		8	↕	↕	●	↕
	ASR				07	6	2	67	6+	2+	77	7	3					8	↕	↕	●	↕
BIT	BITA	85	2	2	95	4	2	A5	4+	2+	B5	5	3				Bit Test A (M Λ A)	●	↕	↕	0	●
	BITB	C5	2	2	D5	4	2	E5	4+	2+	F5	5	3				Bit Test B (M Λ B)	●	↕	↕	0	●
CLR	CLRA													4F	2	1	0 → A	●	0	1	0	0
	CLRB													5F	2	1	0 → B	●	0	1	0	0
	CLR				0F	6	2	6F	6+	2+	7F	7	3				0 → M	●	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4+	2+	B1	5	3				Compare M from A	8	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4+	2+	F1	5	3				Compare M from B	8	↕	↕	↕	↕
	CMPD	10 83	5	4	10 93	7	3	10 A3	7+	3+	10 B3	8	4				Compare M:M + 1 from D	●	↕	↕	↕	↕
	CMPS	11 8C	5	4	11 9C	7	3	11 AC	7+	3+	11 BC	8	4				Compare M:M + 1 from S	●	↕	↕	↕	↕
	CMPU	11 83	5	4	11 93	7	3	11 A3	7+	3+	11 B3	8	4				Compare M:M + 1 from U	●	↕	↕	↕	↕
	CMPX	8C	4	3	9C	6	2	AC	6+	2+	BC	7	3				Compare M:M + 1 from X	●	↕	↕	↕	↕
	CMPY	10 8C	5	4	10 9C	7	3	10 AC	7+	3+	10 BC	8	4				Compare M:M + 1 from Y	●	↕	↕	↕	↕
COM	COMA													43	2	1	$\overline{A}$ → A	●	↕	↕	0	1
	COMB													53	2	1	$\overline{B}$ → B	●	↕	↕	0	1
	COM				03	6	2	63	6+	2+	73	7	3				$\overline{M}$ → M	●	↕	↕	0	1
CWAI		3C	≥20	2													CC Λ IMM → CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	●	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	●	↕	↕	↕	●
	DECB													5A	2	1	B − 1 → B	●	↕	↕	↕	●
	DEC				0A	6	2	6A	6+	2+	7A	7	3				M − 1 → M	●	↕	↕	↕	●
EOR	EORA	88	2	2	98	4	2	A8	4+	2+	B8	5	3				A ⊻ M → A	●	↕	↕	0	●
	EORB	C8	2	2	D8	4	2	E8	4+	2+	F8	5	3				B ⊻ M → B	●	↕	↕	0	●
EXG	R1, R2													1E	8	2	R1 ↔ R2^2	●	●	●	●	●
INC	INCA													4C	2	1	A + 1 → A	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	↕	↕	↕	●
	INC				0C	6	2	6C	6+	2+	7C	7	3				M + 1 → M	●	↕	↕	↕	●
JMP					0E	3	2	6E	3+	2+	7E	4	3				EA^3 → PC	●	●	●	●	●
JSR					9D	7	2	AD	7+	2+	BD	8	3				Jump to Subroutine	●	●	●	●	●
LD	LDA	86	2	2	96	4	2	A6	4+	2+	B6	5	3				M → A	●	↕	↕	0	●
	LDB	C6	2	2	D6	4	2	E6	4+	2+	F6	5	3				M → B	●	↕	↕	0	●
	LDD	CC	3	3	DC	5	2	EC	5+	2+	FC	6	3				M:M + 1 → D	●	↕	↕	0	●
	LDS	10 CE	4	4	10 DE	6	3	10 EE	6+	3+	10 FE	7	4				M:M + 1 → S	●	↕	↕	0	●
	LDU	CE	3	3	DE	5	2	EE	5+	2+	FE	6	3				M:M + 1 → U	●	↕	↕	0	●
	LDX	8E	3	3	9E	5	2	AE	5+	2+	BE	6	3				M:M + 1 → X	●	↕	↕	0	●
	LDY	10 8E	4	4	10 9E	6	3	10 AE	6+	3+	10 BE	7	4				M:M + 1 → Y	●	↕	↕	0	●
LEA	LEAS							32	4+	2+							EA^3 → S	●	●	●	●	●
	LEAU							33	4+	2+							EA^3 → U	●	●	●	●	●
	LEAX							30	4+	2+							EA^3 → X	●	●	↕	●	●
	LEAY							31	4+	2+							EA^3 → Y	●	●	↕	●	●

Legend:		$\overline{M}$	Complement of M	↕	Test and set if true, cleared otherwise
OP	O ıtion Code (Hexadecimal)	→	Transfer Into	●	Not Affected
~	N ıer of MPU Cycles	H	Half-carry (from bit 3)	CC	Condition Code Register
#	N ıer of Program Bytes	N	Negative (sign bit)	:	Concatenation
+	A ıetic Plus	Z	Zero result	V	Logical or
−	A ıetic Minus	V	Overflow, 2's complement	Λ	Logical and
●	M y	C	Carry from ALU	⊻	Logical Exclusive or

FIGURE 21 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Addressing Modes															Description	5	3	2	1	0
		Immediate			Direct			Indexed[1]			Extended			Inherent				H	N	Z	V	C
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#						
LSL	LSLA													48	2	1		•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1		•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}$ + 1 → A	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}$ + 1 → B	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}$ + 1 → M	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A V M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B V M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC V IMM → CC				7	
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1		•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1		•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A − M − C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B − M − C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M M + 1	•	↕	↕	0	•
	STS				10 DF	6	3	10 EF	6+	3+	10 FF	7	4				S → M M + 1	•	↕	↕	0	•
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M M + 1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M M + 1	•	↕	↕	0	•
	STY				10 9F	6	3	10 AF	6+	3+	10 BF	7	4				Y → M M + 1	•	↕	↕	0	•
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A − M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B − M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D − M:M + 1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI[6]													11 3F	20	1	Software Interrupt 3	•	•	•	•	•
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2													1F	6	2	R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

Notes:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET

FIGURE 21 — PROGRAMMING AID (CONCLUDED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative			Description	5	3	2	1	0
		OP	~9	#		H	N	Z	V	C
BCC	BCC	24	3	2	Branch C = 0	•	•	•	•	•
	LBCC	10 24	5(6)	4	Long Branch C = 0	•	•	•	•	•
BCS	BCS	25	3	2	Branch C = 1	•	•	•	•	•
	LBCS	10 25	5(6)	4	Long Branch C = 1	•	•	•	•	•
BEQ	BEQ	27	3	2	Branch Z = 1	•	•	•	•	•
	LBEQ	10 27	5(6)	4	Long Branch Z = 0	•	•	•	•	•
BGE	BGE	2C	3	2	Branch ≥ Zero	•	•	•	•	•
	LBGE	10 2C	5(6)	4	Long Branch ≥ Zero	•	•	•	•	•
BGT	BGT	2E	3	2	Branch > Zero	•	•	•	•	•
	LBGT	10 2E	5(6)	4	Long Branch > Zero	•	•	•	•	•
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10 22	5(6)	4	Long Branch Higher	•	•	•	•	•
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10 24	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
BLE	BLE	2F	3	2	Branch ≤ Zero	•	•	•	•	•
	LBLE	10 2F	5(6)	4	Long Branch ≤ Zero	•	•	•	•	•
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10 25	5(6)	4	Long Branch Lower	•	•	•	•	•

Instruction	Forms	Addressing Mode Relative			Description	5	3	2	1	0
		OP	~	#		H	N	Z	V	C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10 23	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
BLT	BLT	2D	3	2	Branch < Zero	•	•	•	•	•
	LBLT	10 2D	5(6)	4	Long Branch < Zero	•	•	•	•	•
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10 2B	5(6)	4	Long Branch Minus	•	•	•	•	•
BNE	BNE	26	3	2	Branch Z = 0	•	•	•	•	•
	LBNE	10 26	5(6)	4	Long Branch Z ≠ 0	•	•	•	•	•
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10 2A	5(6)	4	Long Branch Plus	•	•	•	•	•
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10 21	5	4	Long Branch Never	•	•	•	•	•
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V = 0	•	•	•	•	•
	LBVC	10 28	5(6)	4	Long Branch V = 0	•	•	•	•	•
BVS	BVS	29	3	2	Branch V = 1	•	•	•	•	•
	LBVS	10 29	5(6)	4	Long Branch V = 1	•	•	•	•	•

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24

Notes:

1. All conditional branches have both short and long variations.
2. All short branches are 2 bytes and require 3 cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require 4 bytes and 6 cycles if the branch is taken or 5 cycles if the branch is not taken.

ORDERING INFORMATION

MC68A09C P

Motorola Integrated Circuit
M6800 Family
Blanks = 1.0 MHz
A = 1.5 MHz
B = 2.0 MHz
Device Designation
In M6800 Family
Temperature Range
Blank = 0° → +70°C
C = −40° → +85°C
Package
P = Plastic
S = Cerdip
L = Ceramic

MOTOROLA

MC6809E
(1.0 MHz)
MC68A09E
(1.5 MHz)
MC68B09E
(2.0 MHz)

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

8-BIT
MICROPROCESSING
UNIT

8-BIT MICROPROCESSING UNIT

The MC6809E is a revolutionary high performance 8-bit microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the M6800 family has major architectural improvements which include additional registers, instructions and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809E has the most complete set of addressing modes available on any 8-bit microprocessor today.

The MC6809E has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications. External clock inputs are provided to allow synchronization with peripherals, systems or other MPUs.

MC6800 COMPATIBLE

- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

ARCHITECTURAL FEATURES

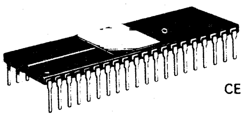
- Two 16-bit Index Registers
- Two 16-bit Indexable Stack Pointers
- Two 8-bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

HARDWARE FEATURES

- External Clock Inputs, E and Q, Allow Synchronization
- TSC Input Controls Internal Bus Buffers
- LIC Indicates Opcode Fetch
- AVMA Allows Efficient Use of Common Resources in A Multiprocessor System
- BUSY is a Status Line for Multiprocessing
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- Interrupt Acknowledge Output Allows Vectoring By Devices
- SYNC Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use With Slower Memories
- Early Write-Data for Dynamic Memories

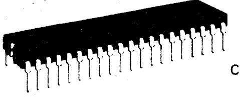
SOFTWARE FEATURES

- 10 Addressing Modes
  - M6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing:
    0, 5, 8, or 16-bit Constant Offsets
    8, or 16-bit Accumulator Offsets
    Auto-Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instructions with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address



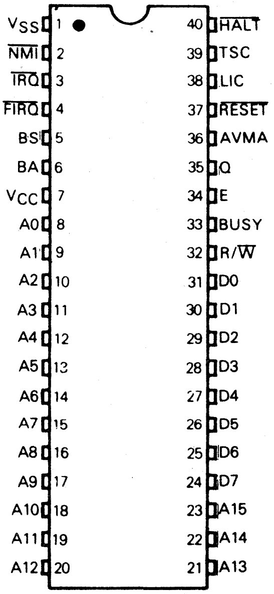
L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

FIGURE 1 — PIN ASSIGNMENT



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 MC6809E, MC68A09E, MC68B09E	T_A	T_L to T_H 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.

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		50	
Cerdip	θ_{JA}	60	°C/W
Plastic		100	

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

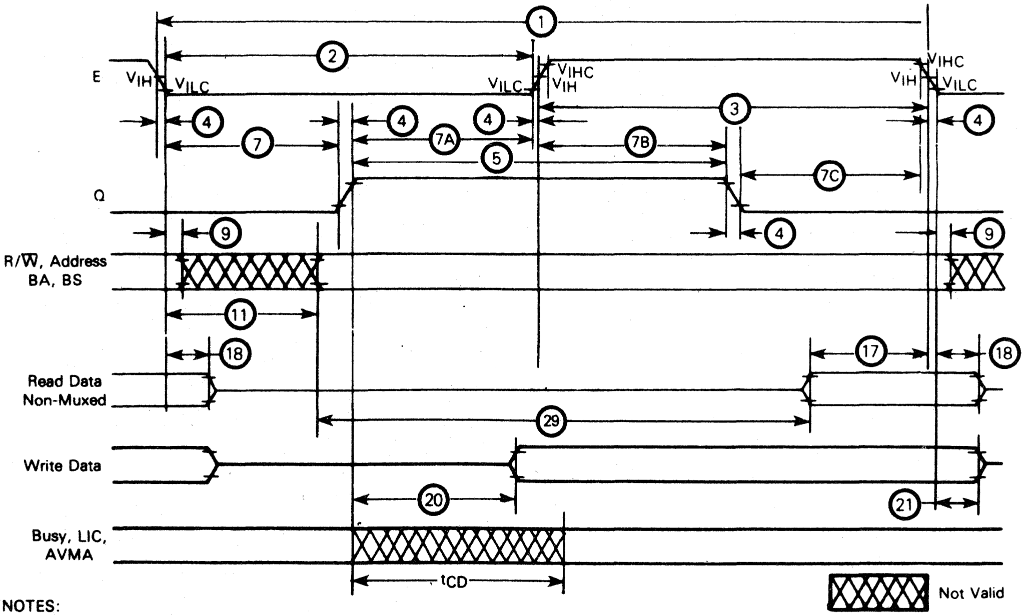
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, Q,	V_{IH}	V_{SS} + 2.0		V_{CC}	V
	$\overline{\text{RESET}}$	V_{IHR}	V_{SS} + 4.0	–	V_{CC}	
	E	V_{IHC}	V_{CC} − 0.75	–	V_{CC} + 0.3	
Input Low Voltage	Logic, Q, $\overline{\text{RESET}}$	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
	E	V_{ILC}	V_{SS} − 0.3	–	V_{SS} + 0.4	
Input Leakage Current	Logic, Q, $\overline{\text{RESET}}$	I_{in}	–	–	2.5	µA
(V_{in} = 0 to 5.25 V, V_{CC} = max)	E		–	–	100	
DC Output High Voltage						
(I_{Load} = −205 µA, V_{CC} = min)	D0-D7	V_{OH}	V_{SS} + 2.4	–	–	V
(I_{Load} = −145 µA, V_{CC} = min)	A0-A15, R/$\overline{\text{W}}$		V_{SS} + 2.4	–	–	
(I_{Load} = −100 µA, V_{CC} = min)	BA, BS, LIC, AVMA, BUSY		V_{SS} + 2.4	–	–	
DC Output Low Voltage (I_{Load} = 2.0 mA, V_{CC} = min)		V_{OL}	–	–	V_{SS} + 0.5	V
Internal Power Dissipation (Measured at $T_A = T_L$ in Steady State Operation)		P_{INT}	–	–	1.0	W
Capacitance*		C_{in}				
(V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7, Logic Inputs, Q, $\overline{\text{RESET}}$		–	10	15	pF
	E		–	30	50	
	A0-A15, R/$\overline{\text{W}}$, BA, BS LIC, AVMA, BUSY	C_{out}	–	10	15	pF
Frequency of Operation	MC6809E	f	0.1	–	1.0	MHz
	MC68A09E		0.1	–	1.5	
(E and Q Inputs)	MC68B09E		0.1	–	2.0	
Three-State (Off State) Input Current	D0-D7	I_{TSI}	–	2.0	10	µA
(V_{in} = 0.4 to 2.4 V, V_{CC} = max)	A0-A15, R/$\overline{\text{W}}$		–	–	100	

*Capacitances are periodically tested rather than 100% tested.

BUS TIMING CHARACTERISTICS (See Notes 1, 2, 3, and 4)

Ident. Number	Characteristics	Symbol	MC6809E		MC68A09E		MC68B09E		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	450	9500	295	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	450	9500	280	9500	220	9500	ns
7	Delay Time, E to Q Rise	t_{EQ1}	200	–	130	–	100	–	ns
7A	Delay Time, Q High to E Rise	t_{EQ2}	200	–	130	–	100	–	ns
7B	Delay Time, E High to Q Fall	t_{EQ3}	200	–	130	–	100	–	ns
7C	Delay Time, Q High to E Fall	t_{EQ4}	200	–	130	–	100	–	ns
9	Address Hold Time	t_{AH}	20	–	20	–	20	–	ns
11	Address Delay Time from E Low (BA, BS, R/$\overline{W}$)	t_{AD}	–	200	–	140	–	110	ns
17	Read Data Setup Time	t_{DSR}	80	–	60	–	40	–	ns
18	Read Data Hold Time	t_{DHR}	10	–	10	–	10	–	ns
20	Data Delay Time from Q	t_{DDQ}	–	200	–	140	–	110	ns
21	Write Data Hold Time	t_{DHW}	30	–	30	–	30	–	ns
29	Usable Access Time	t_{ACC}	695	–	440	–	330	–	ns
	Control Delay Time (Figure 2)	t_{CD}	–	300	–	250	–	200	ns
	Interrupts, $\overline{HALT}$, $\overline{RESET}$, and TSC Setup Time (Figures 7, 8, 9, 10, 13, and 14)	t_{PCS}	200	–	140	–	110	–	ns
	TSC Drive to Valid Logic Level (Figure 14)	t_{TSV}	–	210	–	150	–	120	ns
	TSC Release MOS Buffers to High Impedance (Figure 14)	t_{TSR}	–	200	–	140	–	110	ns
	TSC Three-State Delay Time (Figure 14)	t_{TSC}	–	120	–	85	–	80	ns
	Processor Control Rise and Fall Time (Figure 8)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

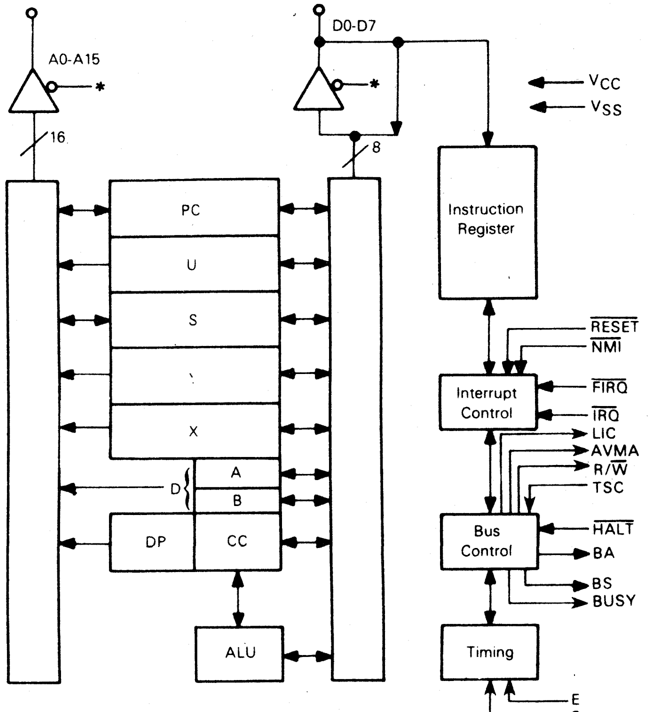
FIGURE 2 — READ/WRITE DATA TO MEMORY OR PERIPHERALS



NOTES:

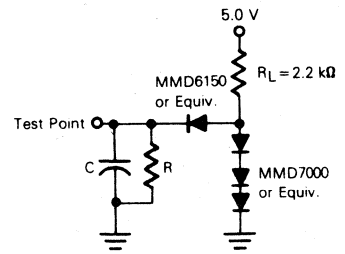
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_{IH} \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Hold time (⑨) for BA and BS is not specified.
4. Usable access time is computed by: 1 – 4 – 11 max – 17.

FIGURE 3 — MC6809E EXPANDED BLOCK DIAGRAM



* Internal Three-State Control

FIGURE 4 — BUS TIMING TEST LOAD



C = 30 pF for BA, BS, LIC, AVMA, BUSY
130 pF for D0-D7
90 pF for A0-A15, R/$\overline{W}$

R = 11.7 kΩ for D0-D7
16.5 kΩ for A0-A15, R/$\overline{W}$
24 kΩ for BA, BS
LIC, AVMA, BUSY

PROGRAMMING MODEL

As shown in Figure 5, the MC6809E adds three registers to the set available in the MC6800. The added registers include a Direct Page Register, the User Stack pointer and a second Index Register.

ACCUMULATORS (A, B, D)

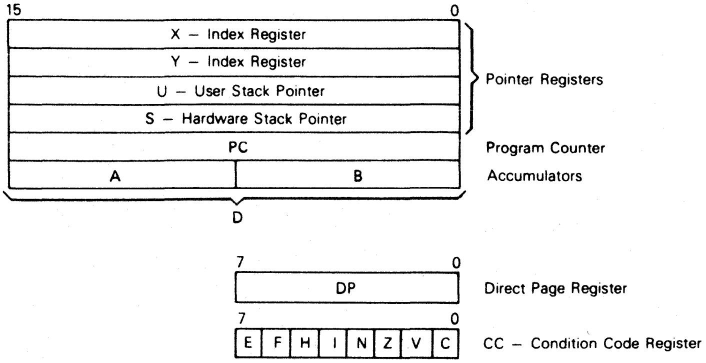
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

Certain instructions concatenate the A and B registers to form a single 16-bit accumulator. This is referred to as the D Register, and is formed with the A Register as the most significant byte.

DIRECT PAGE REGISTER (DP)

The Direct Page Register of the MC6809E serves to enhance the Direct Addressing Mode. The content of this register appears at the higher address outputs (A8-A15) during direct addressing instruction execution. This allows the direct mode to be used at any place in memory, under program control. To ensure M6800 compatibility, all bits of this register are cleared during Processor Reset.

FIGURE 5 — PROGRAMMING MODEL OF THE MICROPROCESSING UNIT



INDEX REGISTERS (X, Y)

The Index Registers are used in indexed mode of addressing. The 16-bit address in this register takes part in the calculation of effective addresses. This address may be used to point to data directly or may be modifed by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented or decremented to point to the next item of tabular type data. All four pointer register (X, Y, U, S) may be used as index registers.

STACK POINTER (U, S)

The Hardware Stack Pointer (S) is used automatically by the processor during subroutine calls and interrupts. The User Stack Pointer (U) is controlled exclusively by the programmer thus allowing arguments to be passed to and from subroutines with ease. The U-register is frequently used as a stack marker. Both Stack Pointers have the same indexed mode addressing capabilities as the X and Y registers, but also support **Push** and **Pull** instructions. This allows the MC6809E to be used efficiently as a stack processor, greatly enhancing its ability to support higher level languages and modular programming.

NOTE

The stack pointers of the MC6809E point to the top of the stack, in contrast to the MC6800 stack pointer, which pointed to the next free location on stack.

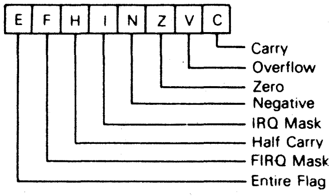
PROGRAM COUNTER

The Program Counter is used by the processor to point to the address of the next instruction to be executed by the processor. Relative Addressing is provided allowing the Program Counter to be used like an index register in some situations.

CONDITION CODE REGISTER

The Condition Code Register defines the state of the processor at any given time. See Figure 6.

FIGURE 6 — CONDITION CODE REGISTER FORMAT



CONDITION CODE REGISTER DESCRIPTION

BIT 0 (C)

Bit 0 is the carry flag, and is usually the carry from the binary ALU. C is also used to represent a 'borrow' from subtract like instructions (CMP, NEG, SUB, SBC) and is the complement of the carry from the binary ALU.

BIT 1 (V)

Bit 1 is the overflow flag, and is set to a one by an operation which causes a signed two's complement arithmetic overflow. This overflow is detected in an operation in which the carry from the MSB in the ALU does not match the carry from the MSB-1.

BIT 2 (Z)

Bit 2 is the zero flag, and is set to a one if the result of the previous operation was identically zero.

BIT 3 (N)

Bit 3 is the negative flag, which contains exactly the value of the MSB of the result of the preceding operation. Thus, a negative two's-complement result will leave N set to a one.

BIT 4 (I)

Bit 4 is the $\overline{\text{IRQ}}$ mask bit. The processor will not recognize interrupts from the $\overline{\text{IRQ}}$ line if this bit is set to a one. $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, and SWI all set I to a one; SWI2 and SWI3 do not affect I.

BIT 5 (H)

Bit 5 is the half-carry bit, and is used to indicate a carry from bit 3 in the ALU as a result of an 8-bit addition only (ADC or ADD). This bit is used by the DAA instruction to perform a BCD decimal add adjust operation. The state of this flag is undefined in all subtract-like instructions.

BIT 6 (F)

Bit 6 is the $\overline{\text{FIRQ}}$ mask bit. The processor will not recognize interrupts from the $\overline{\text{FIRQ}}$ line if this bit is a one. $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, SWI, and $\overline{\text{RESET}}$ all set F to a one. $\overline{\text{IRQ}}$, SWI2 and SWI3 do not affect F.

BIT 7 (E)

Bit 7 is the entire flag, and when set to a one indicates that the complete machine state (all the registers) was stacked, as opposed to the subset state (PC and CC). The E bit of the stacked CC is used on a return from interrupt (RTI) to determine the extent of the unstacking. Therefore, the current E left in the Condition Code Register represents past action.

PIN DESCRIPTIONS

POWER (V_{SS}, V_{CC})

Two pins are used to supply power to the part: V_{SS} is ground or 0 volts, while V_{CC} is +5.0 V ±5%.

ADDRESS BUS (A0-A15)

Sixteen pins are used to output address information from the MPU onto the Address Bus. When the processor does not require the bus for a data transfer, it will output address FFFF$_{16}$, R/$\overline{\text{W}}$ = 1, and BS = 0; this is a "dummy access" or $\overline{\text{VMA}}$ cycle. All address bus drivers are made high-impedance when output Bus Available (BA) is high or when TSC is asserted. Each pin will drive one Schottky TTL load or four LS TTL loads, and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bi-directional data bus. Each pin will drive one Schottky TTL load or four LS TTL loads, and 130 pF.

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

This signal indicates the direction of data transfer on the data bus. A low indicates that the MPU is writing data onto the data bus. R/$\overline{\text{W}}$ is made high impedance when BA is high or when TSC is asserted.

$\overline{\text{RESET}}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 7. The Reset vectors are fetched from locations FFFE$_{16}$ and FFFF$_{16}$ (Table 1) when Interrupt Acknowledge is true, ($\overline{\text{BA}}$ • BS = 1). During initial power-on, the Reset line should be held low until the clock input signals are fully operational.

Because the MC6809E Reset pin has a Schmitt-trigger input with a threshold voltage higher than that of standard peripherals, a simple R/C network may be used to reset the entire system. This higher threshold voltage ensures that all peripherals are out of the reset state before the Processor.

$\overline{\text{HALT}}$

A low level on this input pin will cause the MPU to stop running at the end of the present instruction and remain halted indefinitely without loss of data. When halted, the BA output is driven high indicating the buses are high impedance. BS is also high which indicates the processor is in the Halt state. While halted, the MPU will not respond to external real-time requests ($\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$) although $\overline{\text{NMI}}$ or $\overline{\text{RESET}}$ will be latched for later response. During the Halt state Q and E should continue to run normally. A halted state (BA•BS = 1) can be achieved by pulling $\overline{\text{HALT}}$ low while $\overline{\text{RESET}}$ is still low. See Figure 8.

BUS AVAILABLE, BUS STATUS (BA, BS)

The Bus Available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. When BA goes low, a dead cycle will elapse before the MPU acquires the bus. BA will not be asserted when TSC is active, thus allowing dead cycle consistency.

The Bus Status output signal, when decoded with BA, represents the MPU state (valid with leading edge of Q).

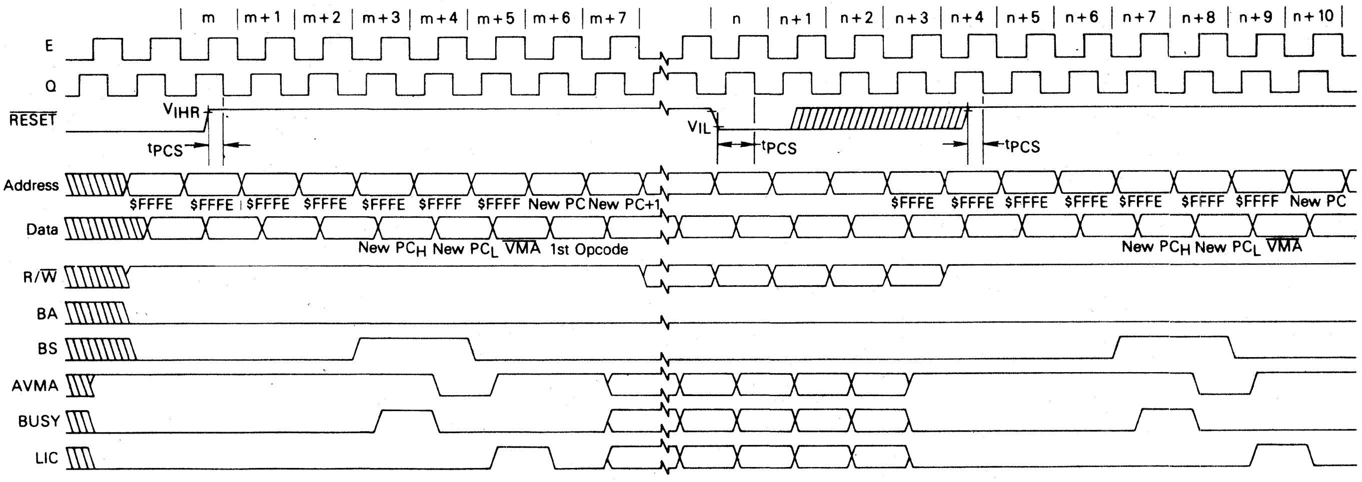
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or RESET Acknowledge
1	0	SYNC Acknowledge
1	1	HALT Acknowledge

Interrupt Acknowledge is indicated during both cycles of a hardware-vector-fetch ($\overline{\text{RESET}}$, $\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$, SWI, SWI2, SWI3). This signal, plus decoding of the lower four address lines, can provide the user with an indication of which interrupt level is being serviced and allow vectoring by device. See Table 1.

TABLE 1 — MEMORY MAP FOR INTERRUPT VECTORS

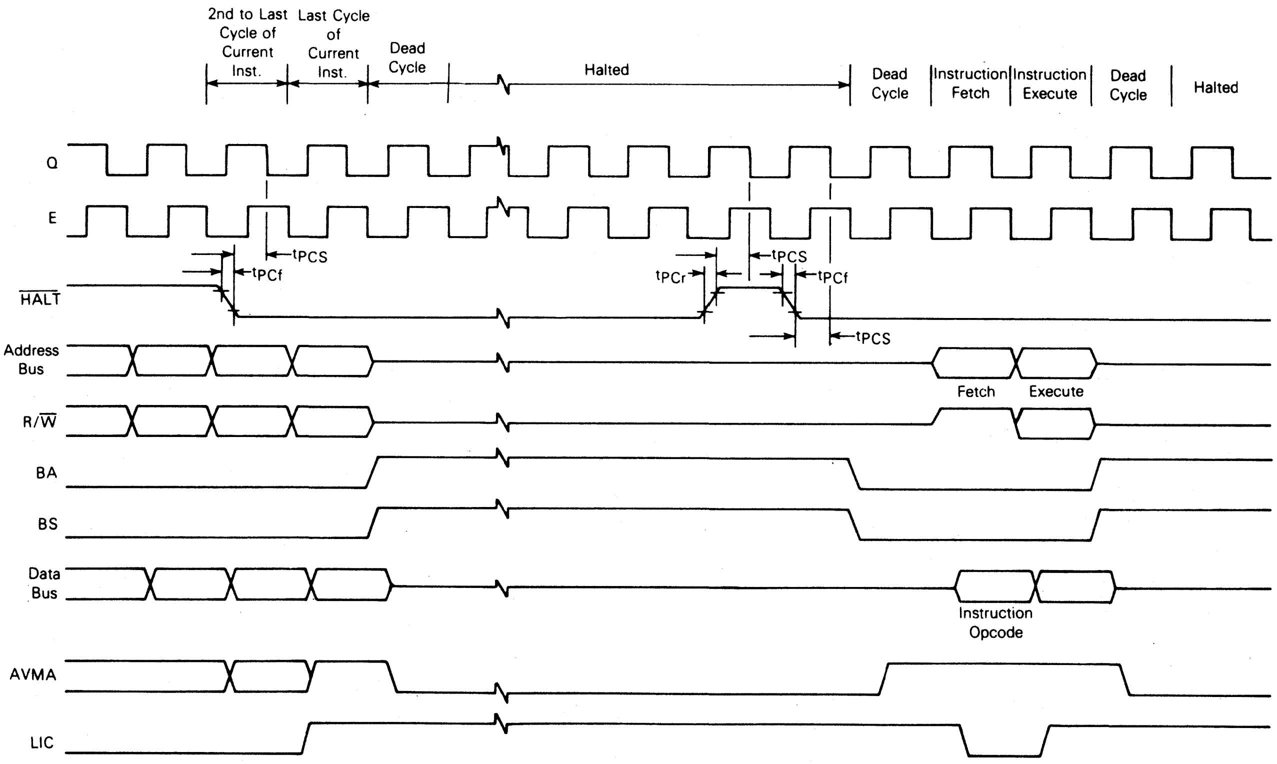
Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{\text{RESET}}$
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{\text{IRQ}}$
FFF6	FFF7	$\overline{\text{FIRQ}}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

FIGURE 7 — $\overline{\text{RESET}}$ TIMING



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 8 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION TIMING FOR SYSTEM DEBUG



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.

Sync Acknowledge is indicated while the MPU is waiting for external synchronization on an interrupt line.

Halt/Acknowledge is indicated when the MC6809E is in a Halt condition.

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

A negative transition on this input requests that a non-maskable interrupt sequence be generated. A non-maskable interrupt cannot be inhibited by the program, and also has a higher priority than $\overline{\text{FIRQ}}$, $\overline{\text{IRQ}}$ or software interrupts. During recognition of an $\overline{\text{NMI}}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{\text{NMI}}$ will not be recognized until the first program load of the Hardware Stack Pointer (S). The pulse width of $\overline{\text{NMI}}$ low must be at least one E cycle. If the $\overline{\text{NMI}}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 9.

FAST-INTERRUPT REQUEST ($\overline{\text{FIRQ}}$)*

A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard Interrupt Request ($\overline{\text{IRQ}}$), and is fast in the sense that it stacks only the contents of the condition code register and the program counter. The interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 10.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$)*

A low level input on this pin will initiate an Interrupt Request sequence provided the mask bit (I) in the CC is clear. Since $\overline{\text{IRQ}}$ stacks the entire machine state it provides a slower response to interrupts than $\overline{\text{FIRQ}}$. $\overline{\text{IRQ}}$ also has a lower priority than $\overline{\text{FIRQ}}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 9.

CLOCK INPUTS E, Q

E and Q are the clock signals required by the MC6809E. Q must lead E; that is, a transition on Q must be followed by a similar transition on E after a minimum delay. Addresses will be valid from the MPU, t_{AD} after the falling edge of E, and data will be latched from the bus by the falling edge of E. While the Q input is fully TTL compatible, the E input directly drives internal MOS circuitry and, thus, requires a high level above normal TTL levels. This approach minimizes clock skew inherent with an internal buffer. Timing and waveforms for E and Q are shown in Figure 2 while Figure 11 shows a simple clock generator for the MC6809E.

BUSY

Busy will be high for the read and modify cycles of a read-modify-write instruction and during the access of the first byte of a double-byte operation (e.g., LDX, STD, ADDD). Busy is also high during the first byte of any indirect or other vector fetch (e.g., jump extended, SWI indirect etc.).

In a multi-processor system, busy indicates the need to defer the rearbitration of the next bus cycle to insure the integrity of the above operations. This difference provides the indivisible memory access required for a "test-and-set" primitive, using any one of several read-modify-write instructions.

Busy does not become active during PSH or PUL operations. A typical read-modify-write instruction (ASL) is shown in Figure 12. Timing information is given in Figure 13. Busy is valid t_{CD} after the rising edge of Q.

AVMA

AVMA is the Advanced VMA signal and indicates that the MPU will use the bus in the following bus cycle. The predictive nature of the AVMA signal allows efficient shared-bus multiprocessor systems. AVMA is LOW when the MPU is in either a HALT or SYNC state. AVMA is valid t_{CD} after the rising edge of Q.

LIC

LIC (Last Instruction Cycle) is HIGH during the last cycle of every instruction, and its transition from HIGH to LOW will indicate that the first byte of an opcode will be latched at the end of the present bus cycle. LIC will be HIGH when the MPU is Halted at the end of an instruction, (i.e., not in CWAI or RESET) in SYNC state or while stacking during interrupts. LIC is valid t_{CD} after the rising edge of Q.

TSC

TSC (Three-State Control) will cause MOS address, data, and R/$\overline{\text{W}}$ buffers to assume a high-impedance state. The control signals (BA, BS, BUSY, AVMA and LIC) will not go to the high-impedance state. TSC is intended to allow a single bus to be shared with other bus masters (processors or DMA controllers).

While E is low, TSC controls the address buffers and R/$\overline{\text{W}}$ directly. The data bus buffers during a write operation are in a high-impedance state until Q rises at which time, if TSC is true, they will remain in a high-impedance state. If TSC is held beyond the rising edge of E, then it will be internally latched, keeping the bus drivers in a high-impedance state for the remainder of the bus cycle. See Figure 14.

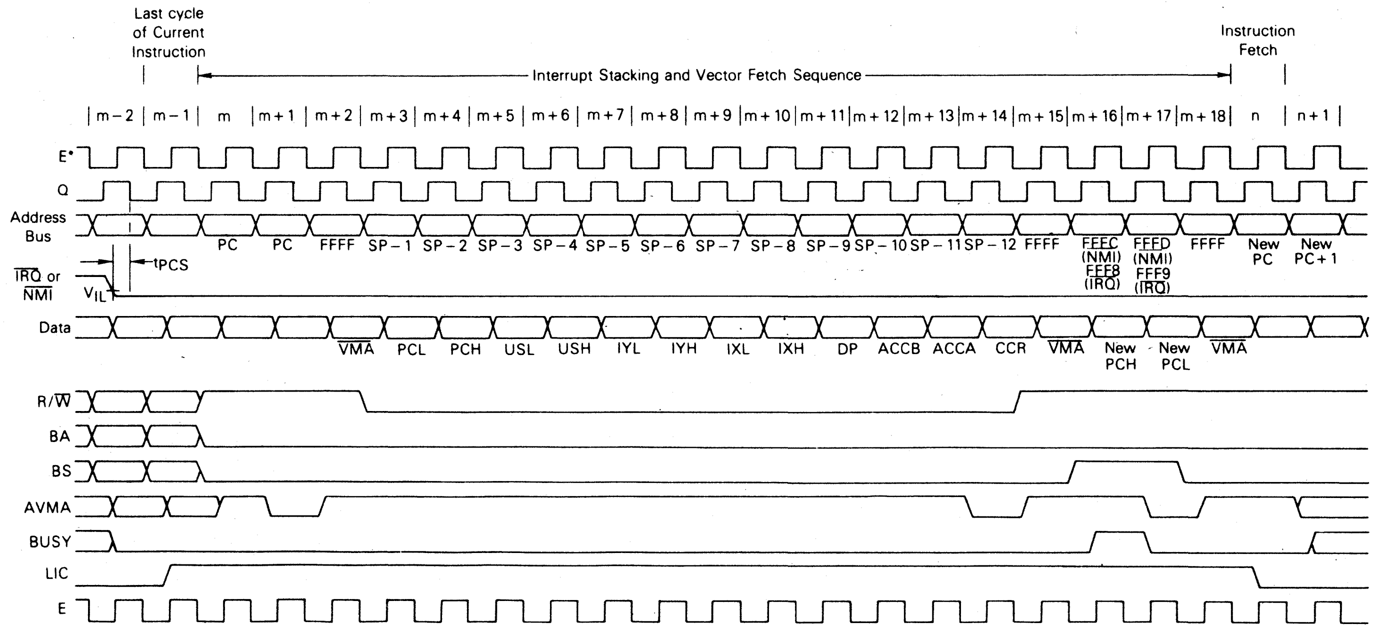
MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{\text{RESET}}$ and is repeated indefinitely unless altered by a special instruction or hardware occurrence. Software instructions that alter normal MPU operation are: SWI, SWI2, SWI3, CWAI, RTI and SYNC. An interrupt or $\overline{\text{HALT}}$ input can also alter the normal execution of instructions. Figure 15 is the flow chart for the MC6809E.

---

*$\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, and $\overline{\text{IRQ}}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless a SYNC or CWAI condition is present. If $\overline{\text{IRQ}}$ and $\overline{\text{FIRQ}}$ do not remain low until completion of the current instruction they may not be recognized. However, $\overline{\text{NMI}}$ is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{\text{RESET}}$ and the rising edge of BS indicating $\overline{\text{RESET}}$ acknowledge. See $\overline{\text{RESET}}$ sequence in the MPU flowchart in Figure 15.

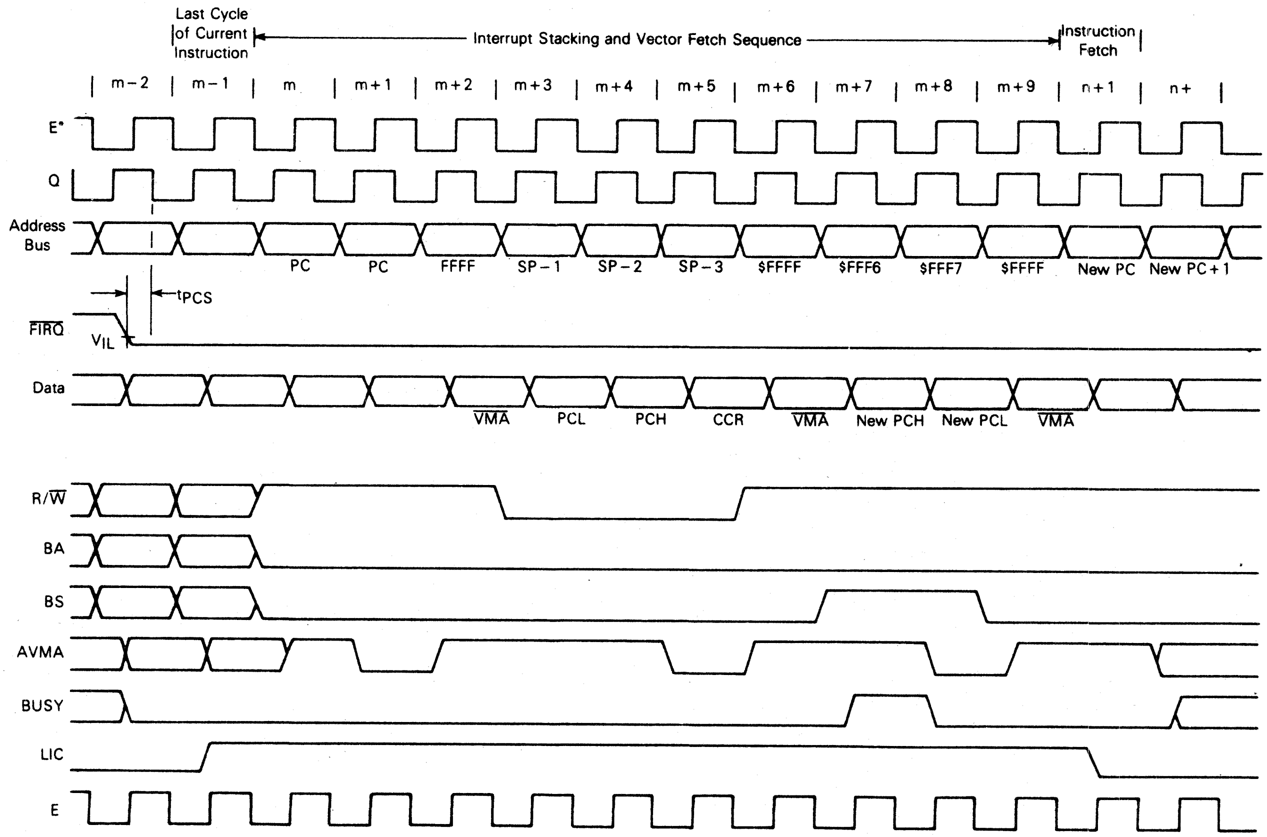
FIGURE 9 — $\overline{\text{IRQ}}$ AND $\overline{\text{NMI}}$ INTERRUPT TIMING



*E clock shown for reference only.

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.

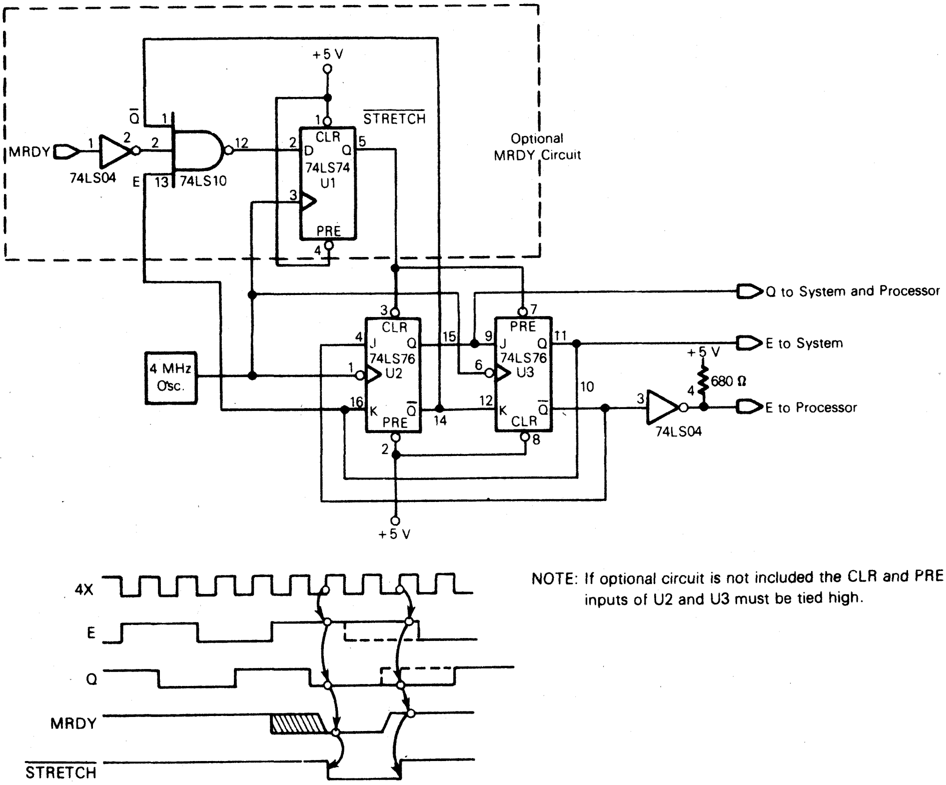
FIGURE 10 — $\overline{\text{FIRQ}}$ INTERRUPT TIMING



*E clock shown for reference only.

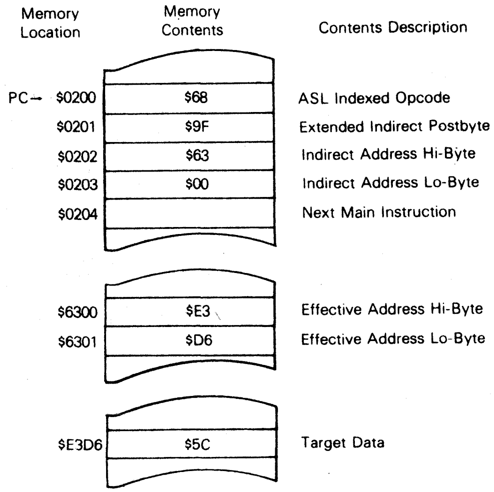
NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 11 – MC6809E CLOCK GENERATOR



NOTE: If optional circuit is not included the CLR and PRE inputs of U2 and U3 must be tied high.

FIGURE 12 – READ-MODIFY-WRITE INSTRUCTION EXAMPLE (ASL EXTENDED INDIRECT)



Memory Location	Memory Contents	Contents Description
PC→ $0200	$68	ASL Indexed Opcode
$0201	$9F	Extended Indirect Postbyte
$0202	$63	Indirect Address Hi-Byte
$0203	$00	Indirect Address Lo-Byte
$0204		Next Main Instruction
$6300	$E3	Effective Address Hi-Byte
$6301	$D6	Effective Address Lo-Byte
$E3D6	$5C	Target Data

FIGURE 13 — BUSY TIMING

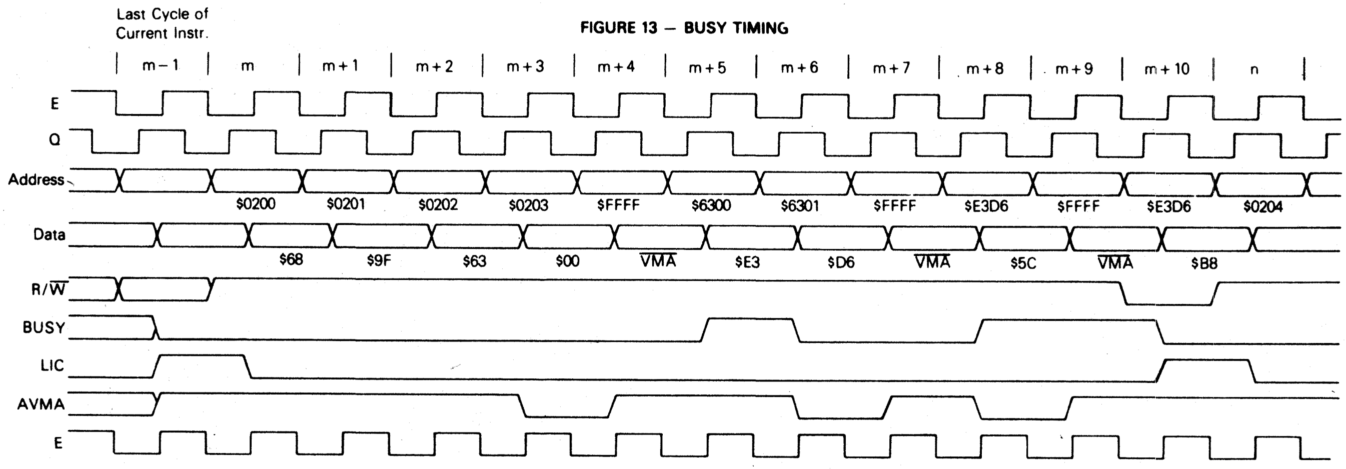
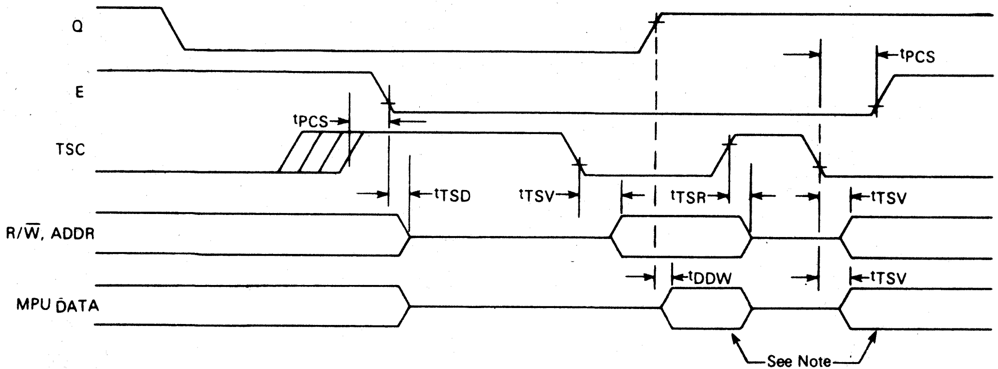
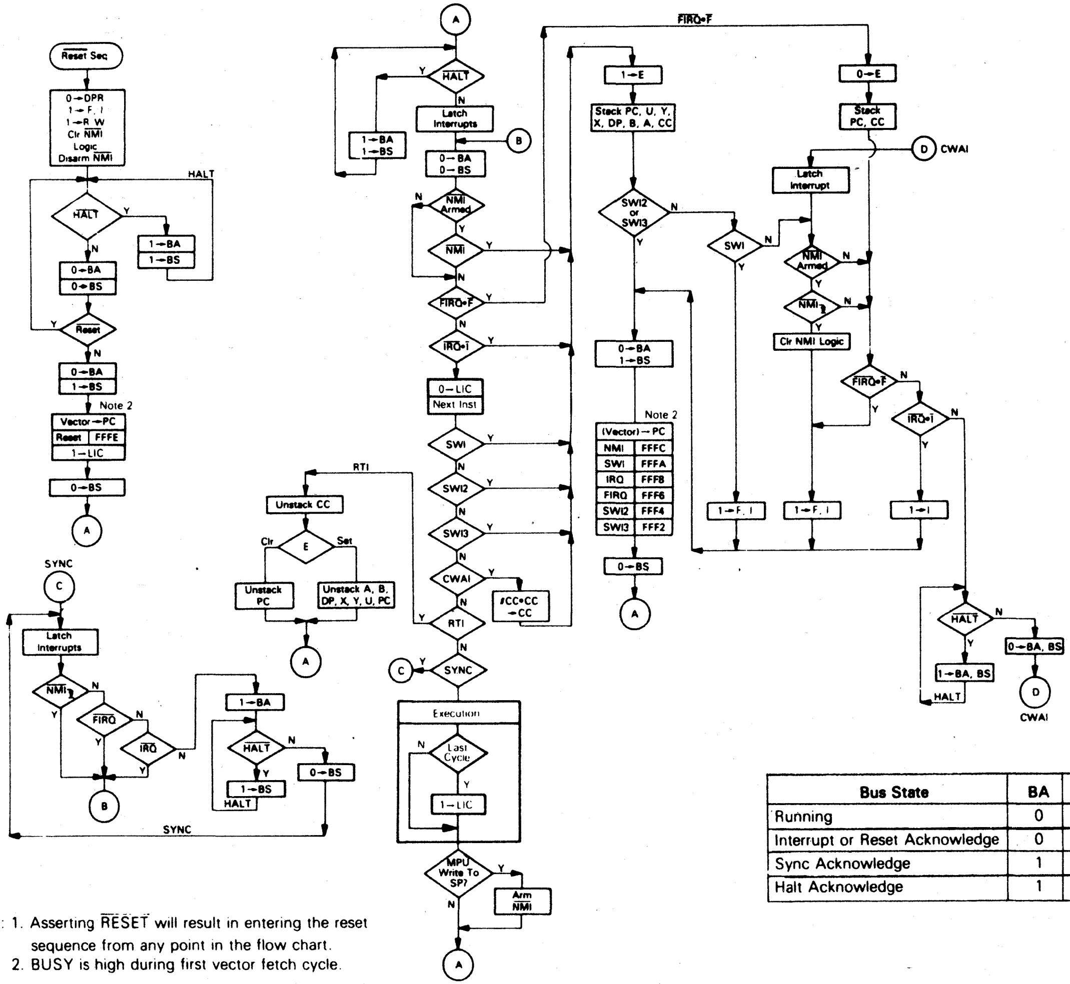


FIGURE 14 — TSC TIMING



NOTE: Data will be asserted by the MPU only during the interval while R/W is low **and** (E or Q) is high.
A composite bus cycle is shown to give most cases of timing.
Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unltess otherwise noted.

FIGURE 15 — FLOWCHART FOR MC6809E INSTRUCTIONS



Bus State	BA	BS
Running	0	0
Interrupt or Reset Acknowledge	0	1
Sync Acknowledge	1	0
Halt Acknowledge	1	1

Notes: 1. Asserting $\overline{\text{RESET}}$ will result in entering the reset sequence from any point in the flow chart.
2. BUSY is high during first vector fetch cycle.

ADDRESSING MODES

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC6809E has the most complete set of addressing modes available on any microcomputer today. For example, the MC6809E has 59 basic instructions; however, it recognizes 1464 different variations of instructions and addressing modes. The addressing modes support modern programming techniques. The following addressing modes are available on the MC6809E:

- Inherent (Includes Accumulator)
- Immediate
- Extended
  - Extended Indirect
- Direct
- Register
- Indexed
  - Zero-Offset
  - Constant Offset
  - Accumulator Offset
  - Auto Increment/Decrement
  - Indexed Indirect
- Relative
  - Short/Long Relative Branching
  - Program Counter Relative Addressing

INHERENT (INCLUDES ACCUMULATOR)

In this addressing mode, the opcode of the instruction contains all the address information necessary. Examples of Inherent Addressing are: ABX, DAA, SWI, ASRA, and CLRB.

IMMEDIATE ADDRESSING

In Immediate Addressing, the effective address of the data is the location immediately following the opcode (i.e., the data to be used in the instruction immediately follows the opcode of the instruction). The MC6809E uses both 8 and 16-bit immediate values depending on the size of argument specified by the opcode. Examples of instructions with Immediate Addressing are:

```
LDA #$20
LDX #$F000
LDY #CAT
```

NOTE: # signifies Immediate addressing, $ signifies hexadecimal value to the MC6809 assembler.

EXTENDED ADDRESSING

In Extended Addressing, the contents of the two bytes immediately following the opcode fully specify the 16-bit effective address used by the instruction. Note that the address generated by an extended instruction defines an absolute address and is not position independent. Examples of Extended Addressing include:

```
LDA CAT
STX MOUSE
LDD $2000
```

EXTENDED INDIRECT

As a special case of indexed addressing (discussed below), one level of indirection may be added to Extended Addressing. In Extended Indirect, the two bytes following the postbyte of an Indexed instruction contain the address of the data.

```
LDA [CAT]
LDX [$FFFE]
STU [DOG]
```

DIRECT ADDRESSING

Direct addressing is similar to extended addressing except that only one byte of address follows the opcode. This byte specifies the lower 8 bits of the address to be used. The upper 8 bits of the address are supplied by the direct page register. Since only one byte of address is required in direct addressing, this mode requires less memory and executes faster than extended addressing. Of course, only 256 locations (one page) can be accessed without redefining the contents of the DP register. Since the DP register is set to $00 on Reset, direct addressing on the MC6809E is upward compatible with direct addressing on the M6800. Indirection is not allowed in direct addressing. Some examples of direct addressing are:

```
LDA    where DP = $00
LDB    where DP = $10
LDD    <CAT
```

NOTE: < is an assembler directive which forces direct addressing.

REGISTER ADDRESSING

Some opcodes are followed by a byte that defines a register or set of registers to be used by the instruction. This is called a postbyte. Some examples of register addressing are:

TFR	X, Y	Transfers X into Y
EXG	A, B	Exchanges A with B
PSHS	A, B, X, Y	Push Y, X, B and A onto S stack
PULU	X, Y, D	Pull D, X, and Y from U stack

INDEXED ADDRESSING

In all indexed addressing, one of the pointer registers (X, Y, U, S, and sometimes PC) is used in a calculation of the effective address of the operand to be used by the instruction. Five basic types of indexing are available and are discussed below. The postbyte of an indexed instruction specifies the basic type and variation of the addressing mode as well as the pointer register to be used. Figure 16 lists the legal formats for the postbyte. Table 2 gives the assembler form and the number of cycles and bytes added to the basic values for indexed addressing for each variation.

FIGURE 16 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit								Indexed Addressing Mode
7	6	5	4	3	2	1	0	
0	R	R	d	d	d	d	d	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R+
1	R	R	i	0	0	0	1	,R++
1	R	R	0	0	0	1	0	,−R
1	R	R	i	0	0	1	1	,−−R
1	R	R	i	0	1	0	0	EA = ,R + 0 Offset
1	R	R	i	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	i	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	i	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	i	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	i	1	0	1	1	EA = ,R + D Offset
1	x	x	i	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	i	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	i	1	1	1	1	EA = [,Address]

Addressing Mode Field

Indirect Field
(Sign bit when b_7 = 0)

Register Field: RR
00 = X
01 = Y
10 = U
11 = S

x = Don't Care
d = Offset Bit
i = 0 = Not Indirect
1 = Indirect

Zero-Offset Indexed — In this mode, the selected pointer register contains the effective address of the data to be used by the instruction. This is the fastest indexing mode.

Examples are:

```
LDD    0, X
LDA    ,S
```

Constant Offset Indexed — In this mode, a two's-complement offset and the contents of one of the pointer registers are added to form the effective address of the operand. The pointer register's initial content is unchanged by the addition.

Three sizes of offsets are available:

5 -bit (−16 to +15)
8 -bit (−128 to +127)
16-bit (−32768 to +32767)

The two's complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles. The two's complement 8-bit offset is contained in a single byte following the postbyte. The two's complement 16-bit offset is in the two bytes following the postbyte. In most cases the programmer need not be concerned with the size of this offset since the assembler will select the optimal size automatically.

Examples of constant-offset indexing are:

```
LDA    23,X
LDX    -2,S
LDY    300,X
LDU    CAT,Y
```

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect				Indirect			
		Assembler Form	Postbyte OP Code	$\overset{+}{\sim}$	$\overset{+}{\#}$	Assembler Form	Postbyte OP Code	$\overset{+}{\sim}$	$\overset{+}{\#}$
Constant Offset From R (2's Complement Offsets)	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
	5 Bit Offset	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8 Bit Offset	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16 Bit Offset	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
Accumulator Offset From R (2's Complement Offsets)	A Register Offset	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B Register Offset	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D Register Offset	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
Auto Increment/Decrement R	Increment By 1	,R+	1RR00000	2	0	not allowed			
	Increment By 2	,R++	1RR00001	3	0	[,R++]	1RR10001	6	0
	Decrement By 1	,−R	1RR00010	2	0	not allowed			
	Decrement By 2	,−−R	1RR00011	3	0	[,−−R]	1RR10011	6	0
Constant Offset From PC (2's Complement Offsets)	8 Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
	16 Bit Offset	n, PCR	1xx01101	5	2	[n, PCR]	1xx11101	8	2
Extended Indirect	16 Bit Address	—	—	—	—	[n]	10011111	5	2

R = X, Y, U or S
x = Don't Care

RR:
00 = X
01 = Y
10 = U
11 = S

$\overset{+}{\sim}$ and $\overset{+}{\#}$ indicate the number of additional cycles and bytes respectively for the particular indexing variation.

Accumulator-Offset Indexed — This mode is similar to constant offset indexed except that the two's-complement value in one of the accumulators (A, B or D) and the contents of one of the pointer registers are added to form the effective address of the operand. The contents of both the accumulator and the pointer register are unchanged by the addition. The postbyte specifies which accumulator to use as an offset and no additional bytes are required. The advantage of an accumulator offset is that the value of the offset can be calculated by a program at run-time.

Some examples are:

```
LDA    B,Y
LDX    D,Y
LEAX   B,X
```

Auto Increment/Decrement Indexed — In the auto increment addressing mode, the pointer register contains the address of the operand. Then, after the pointer register is used it is incremented by one or two. This addressing mode is useful in stepping through tables, moving data, or for the creation of software stacks. In auto decrement, the pointer register is decremented prior to use as the address of the data. The use of auto decrement is similar to that of auto increment; but the tables, etc., are scanned from the high to low addresses. The size of the increment/decrement can be either one or two to allow for tables of either 8- or 16-bit data to be accessed and is selectable by the programmer. The pre-decrement, post-increment nature of these modes allow them to be used to create additional software stacks that behave identically to the U and S stacks.

Some examples of the auto increment/decrement addressing modes are:

```
LDA    ,X+
STD    ,Y++
LDB    ,-Y
LDX    ,--S
```

Care should be taken in performing operations on 16-bit pointer registers (X, Y, U, S) where the same register is used to calculate the effective address.

Consider the following instruction:

```
STX 0,X++ (X initialized to 0)
```

The desired result is to store a 0 in locations $0000 and $0001 then increment X to point to $0002. In reality, the following occurs:

0→temp	calculate the EA; temp is a holding register
X+2→X	perform autoincrement
X→(temp)	do store operation

INDEXED INDIRECT

All of the indexing modes with the exception of auto increment/decrement by one, or a ±5-bit offset may have an additional level of indirection specified. In indirect addressing, the effective address is contained at the location specified by the contents of the Index Register plus any offset. In the example below, the A accumulator is loaded indirectly using an effective address calculated from the Index Register and an offset.

```
Before Execution
A=XX (don't
care)
X=$F000

$0100   LDA  [$10,X]     EA is now $F010

$F010   $F1              $F150 is now the
$F011   $50              new EA

$F150   $AA

        After Execution
        A=$AA (Actual Data Loaded)
        X=$F000
```

All modes of indexed indirect are included except those which are meaningless (e.g., auto increment/decrement by 1 indirect). Some examples of indexed indirect are:

```
LDA    [,X]
LDD    [10,S]
LDA    [B,Y]
LDD    [,X++]
```

RELATIVE ADDRESSING

The byte(s) following the branch opcode is (are) treated as a signed offset which may be added to the program counter. If the branch condition is true then the calculated address (PC + signed offset) is loaded into the program counter. Program execution continues at the new location as indicated by the PC; short (1 byte offset) and long (2 bytes offset) relative addressing modes are available. All of memory can be reached in long relative addressing as an effective address interpreted modulo 2^{16}. Some examples of relative addressing are:

```
          BEQ     CAT       (short)
          BGT     DOG       (short)
CAT       LBEQ    RAT       (long)
DOG       LBGT    RABBIT    (long)
          •
          •
          •
RAT       NOP
RABBIT    NOP
```

PROGRAM COUNTER RELATIVE

The PC can be used as the pointer register with 8 or 16-bit signed offsets. As in relative addressing, the offset is added to the current PC to create the effective address. The effective address is then used as the address of the operand or data. Program Counter Relative Addressing is used for writing position independent programs. Tables related to a particular routine will maintain the same relationship after the routine is moved, if referenced relative to the Program Counter. Examples are:

```
LDA    CAT, PCR
LEAX   TABLE, PCR
```

Since program counter relative is a type of indexing, an additional level of indirection is available.

```
LDA    [CAT, PCR]
LDU    [DOG, PCR]
```

MC6809E INSTRUCTION SET

The instruction set of the MC6809E is similar to that of the MC6800 and is upward compatible at the source code level. The number of opcodes has been reduced from 72 to 59, but because of the expanded architecture and additional addressing modes, the number of available opcodes (with different addressing modes) has risen from 197 to 1464.

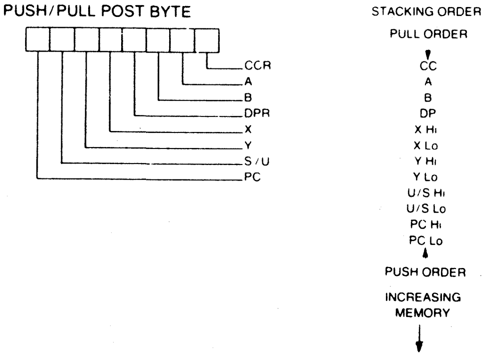
Some of the new instructions are described in detail below:

PSHU/PSHS

The push instructions have the capability of pushing onto either the hardware stack (S) or user stack (U) any single register, or set of registers with a single instruction.

PULU/PULS

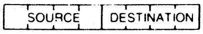
The pull instructions have the same capability of the push instruction, in reverse order. The byte immediately following the push or pull opcode determines which register or registers are to be pushed or pulled. The actual PUSH/PULL sequence is fixed; each bit defines a unique register to push or pull, as shown below.



TFR/EXG

Within the MC6809E, any register may be transferred to or exchanged with another of like-size; i.e., 8-bit to 8-bit or 16-bit to 16-bit. Bits 4-7 of postbyte define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

TRANSFER/EXCHANGE POST BYTE



REGISTER FIELD

0000	D (A B)	1000	A
0001	X	1001	B
0010	Y	1010	CCR
0011	U	1011	DPR
0100	S		
0101	PC		

NOTE: All other combinations are undefined and INVALID.

LEAX/LEAY/LEAU/LEAS

The LEA (Load Effective Address) works by calculating the effective address used in an indexed instruction and stores that address value, rather than the data at that address, in a pointer register. This makes all the features of the internal addressing hardware available to the programmer. Some of the implications of this instruction are illustrated in Table 3.

The LEA instruction also allows the user to access data and tables in a position independent manner. For example:

```
        LEAX   MSG1, PCR
        LBSR   PDATA (Print message routine)
        •
        •
MSG1    FCC    'MESSAGE'
```

This sample program prints: 'MESSAGE'. By writing MSG1, PCR, the assembler computes the distance between the present address and MSG1. This result is placed as a constant into the LEAX instruction which will be indexed from the PC value at the time of execution. No matter where the code is located, when it is executed, the computed offset from the PC will put the absolute address of MSG1 into the X pointer register. This code is totally position independent.

The LEA instructions are very powerful and use an internal holding register (temp). Care must be exercised when using the LEA instructions with the autoincrement and autodecrement addressing modes due to the sequence of internal operations. The LEA internal sequence is outlined as follows:

LEAa ,b+ (any of the 16-bit pointer registers X, Y, U or S may be substituted for a and b.)

1. b→temp (calculate the EA)
2. b+1→b (modify b, postincrement)
3. temp→a (load a)

LEAa ,−b

1. b−1→temp (calculate EA with predecrement)
2. b−1→b (modify b, predecrement)
3. temp→a (load a)

TABLE 3 — LEA EXAMPLES

Instruction	Operation	Comment
LEAX 10, X	X + 10 → X	Adds 5-bit constant 10 to X
LEAX 500, X	X + 500 → X	Adds 16-bit constant 500 to X
LEAY A, Y	Y + A → Y	Adds 8-bit A accumulator to Y
LEAY D, Y	Y + D → Y	Adds 16-bit D accumulator to Y
LEAU −10, U	U − 10 → U	Subtracts 10 from U
LEAS −10, S	S − 10 → S	Used to reserve area on stack
LEAS 10, S	S + 10 → S	Used to 'clean up' stack
LEAX 5, S	S + 5 → X	Transfers as well as adds

Autoincrement-by-two and autodecrement-by-two instructions work similarly. Note that LEAX ,X+ does not change X, however LEAX , −X does decrement X. LEAX 1,X should be used to increment X by one.

MUL

Multiplies the unsigned binary numbers in the A and B accumulator and places the unsigned result into the 16-bit D accumulator. This unsigned multiply also allows multiple-precision multiplications.

Long And Short Relative Branches

The MC6809E has the capability of program counter relative branching throughout the entire memory map. In this mode, if the branch is to be taken, the 8 or 16-bit signed offset is added to the value of the program counter to be used as the effective address. This allows the program to branch anywhere in the 64K memory map. Position independent code can be easily generated through the use of relative branching. Both short (8-bit) and long (16-bit) branches are available.

SYNC

After encountering a Sync instruction, the MPU enters a Sync state, stops processing instructions and waits for an interrupt. If the pending interrupt is non-maskable (NMI) or maskable (FIRQ, IRQ) with its mask bit (F or I) clear, the processor will clear the Sync state and perform the normal interrupt stacking and service routine. Since FIRQ and IRQ are not edge-triggered, a low level with a minimum duration of three bus cycles is required to assure that the interrupt will be taken. If the pending interrupt is maskable (FIRQ, IRQ) with its mask bit (F or I) set, the processor will clear the Sync state and continue processing by executing the next inline instruction. Figure 17 depicts Sync timing.

Software Interrupts

A Software Interrupt is an instruction which will cause an interrupt, and its associated vector fetch. These Software Interrupts are useful in operating system calls, software debugging, trace operations, memory mapping, and software development systems. Three levels of SWI are available on this MC6809E, and are prioritized in the following order: SWI, SWI2, SWI3.

16-Bit Operation

The MC6809E has the capability of processing 16-bit data. These instructions include loads, stores, compares, adds, subtracts, transfers, exchanges, pushes and pulls.

CYCLE-BY-CYCLE OPERATION

The address bus cycle-by-cycle performance chart illustrates the memory-access sequence corresponding to each possible instruction and addressing mode in the MC6809E. Each instruction begins with an opcode fetch. While that opcode is being internally decoded, the next program byte is always fetched. (Most instructions will use the next byte, so this technique considerably speeds throughput.) Next, the operation of each opcode will follow the flow chart. $\overline{\text{VMA}}$ is an indication of FFFF$_{16}$ on the address bus, $R/\overline{W}$ = 1 and BS = 0. The following examples illustrate the use of the chart; see Figure 18.

Example 1: LBSR (Branch Taken)
Before Execution SP = F000

```
                        •
                        •
                        •
$8000                 LBSR      CAT
                        •
                        •
                        •
$A000    CAT            •
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/W	Description
1	8000	17	1	Opcode Fetch
2	8001	20	1	Offset High Byte
3	8002	00	1	Offset Low Byte
4	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
5	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
8	EFFF	80	0	Stack High Order Byte of Return Address
9	EFFE	03	0	Stack Low Order Byte of Return Address

Example 2: DEC (Extended)

```
$8000          DEC          $A000
$A000          FCB          $80
```

CYCLE-BY-CYCLE FLOW

Cycle #	Address	Data	R/W	Description
1	8000	7A	1	Opcode Fetch
2	8001	A0	1	Operand Address, High Byte
3	8002	00	1	Operand Address, Low Byte
4	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{\text{VMA}}$ Cycle
7	A000	7F	0	Store the Decremented Data

*The data bus has the data at that particular address.

MC6809E INSTRUCTION SET TABLES

The instructions of the MC6809E have been broken down into five different categories. They are as follows:

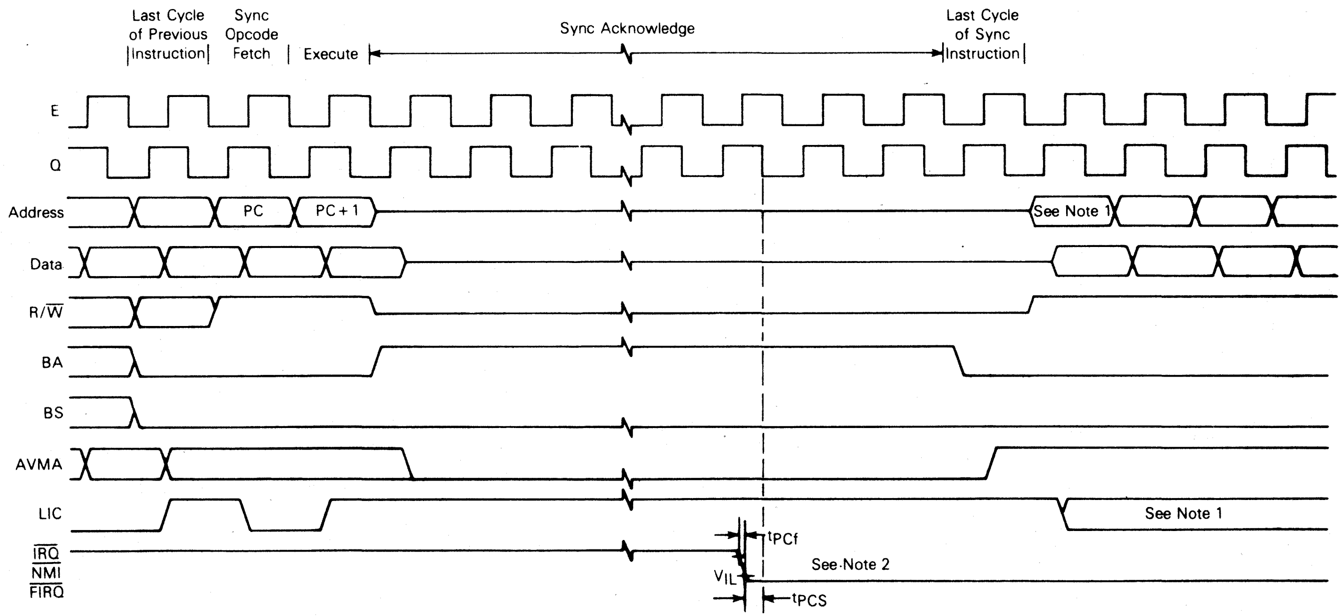
8-Bit operation (Table 4)
16-Bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 18 contains a compilation of data that will assist you in programming the MC6809E.

FIGURE 17 — SYNC TIMING

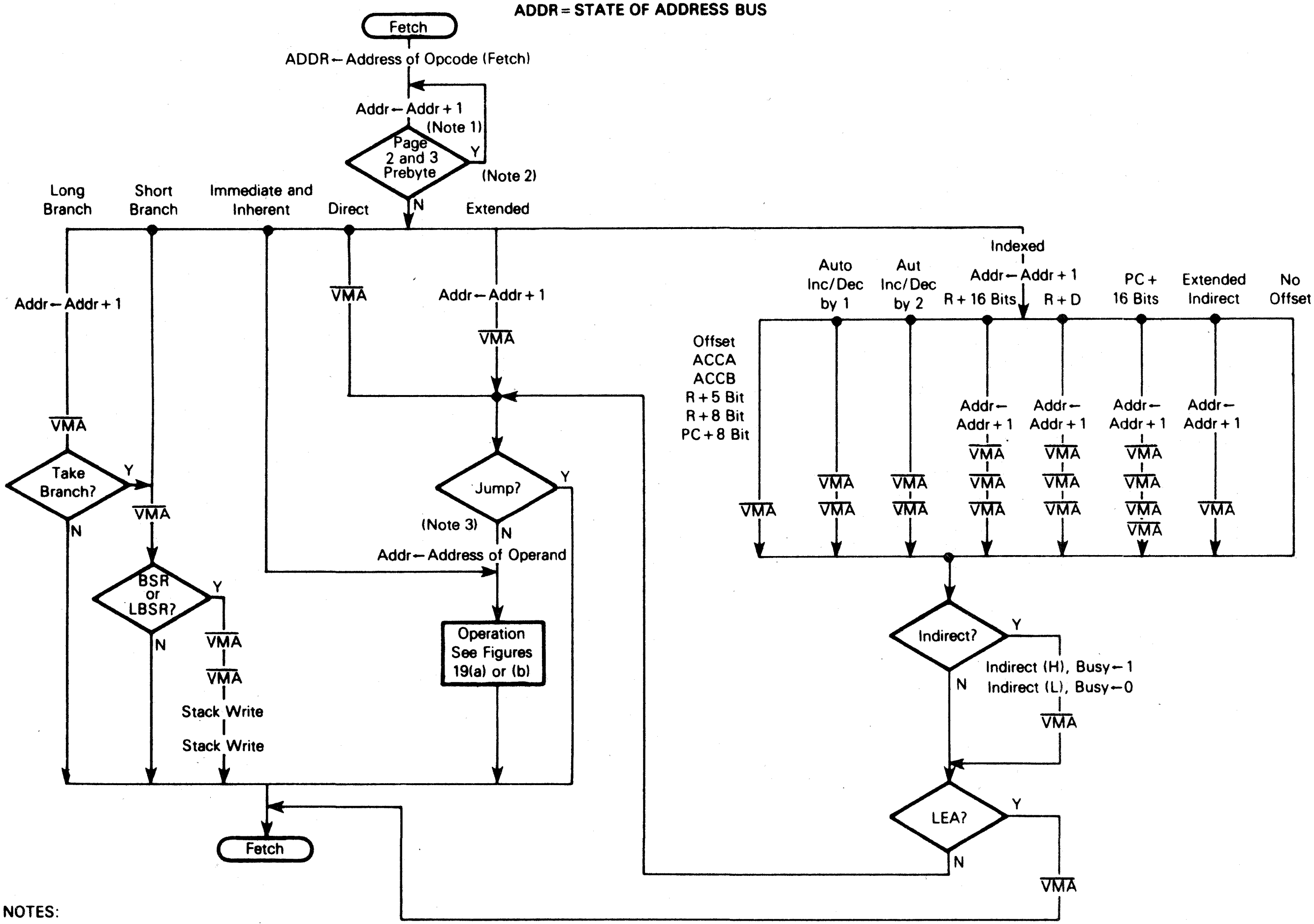


Notes: 1. If the associated mask bit is set when the interrupt is requested, LIC will go low and this cycle will be an instruction fetch from address location PC + 1. However, if the interrupt is accepted ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) LIC will remain high and interrupt processing will start with this cycle as (m) on Figures 9 and 10 (Interrupt Timing).

2. If mask bits are clear, $\overline{IRQ}$ and $\overline{FIRQ}$ must be held low for three cycles to guarantee that interrupt will be taken, although only one cycle is necessary to bring the processor out of SYNC.

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.

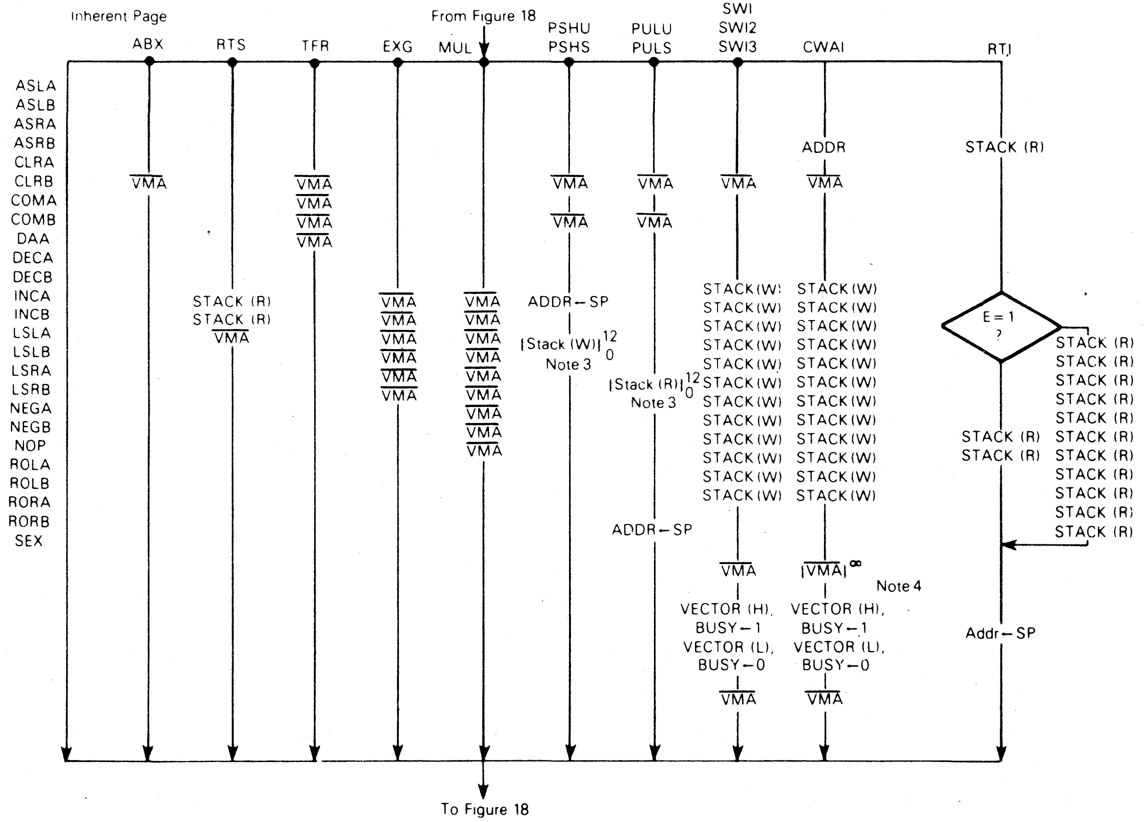
FIGURE 18 — ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE
ADDR = STATE OF ADDRESS BUS



NOTES:

1. Busy = 1 during access of first byte of double byte immediate load.
2. All subsequent Page 2 and Page 3 prebytes will be ignored after initial opcode fetch.
3. Write operation during store instruction. Busy = 1 during first two cycles of a double-byte access and the first cycle of read-modify-write access.
4. AVMA is asserted on the cycle before a $\overline{\text{VMA}}$ cycle.

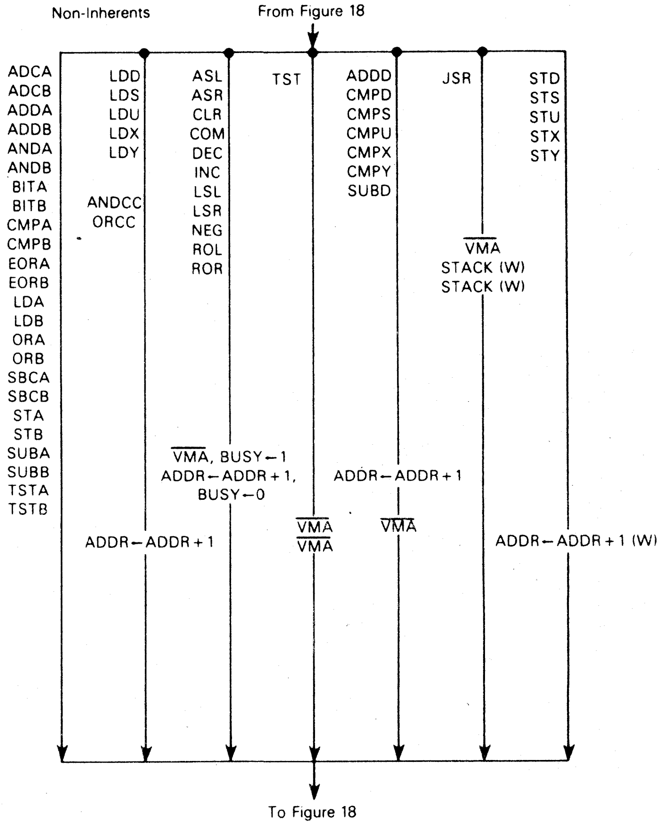
FIGURE 19(a) — OPERATIONS: ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE



NOTES:

1. Stack (W) refers to the following sequence: SP←SP − 1, then ADDR←SP with $R/\overline{W} = 0$.
   Stack (R) refers to the following sequence: ADDR←SP with $R/\overline{W} = 1$, then SP←SP + 1
   PSHU, PULU instructions use the user stack pointer (i.e., SP ≡ U) and PSHS, PULS use the hardware stack pointer (i.e., SP ≡ S).
2. Vector refers to the address of an interrupt or reset vector (see Table 1).
3. The number of stack accesses will vary according to the number of bytes saved
4. $\overline{VMA}$ cycles will occur until an interrupt occurs.

FIGURE 19(b) — OPERATIONS: ADDRESS BUS CYCLE-BY-CYCLE PERFORMANCE



NOTES:

1. Stack (W) refers to the following sequence: SP←SP−1, then ADDR←SP with $R/\overline{W}=0$.
   Stack (R) refers to the following sequence; ADDR←SP with $R/\overline{W}=1$, then SP←SP+1.
   PSHU, PULU instructions use the user stack pointer (i.e., SP=U) and PSHS, PULS use the hardware stack pointer (i.e., SP=S).
2. Vector refers to the address of an interrupt or reset vector (see Table 1).
3. The number of stack accesses will vary according to the number of bytes saved.
4. VMA cycles will occur until an interrupt occurs.

TABLE 4 — 8-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADCA, ADCB	Add memory to accumulator with carry
ADDA, ADDB	Add memory to accumulator
ANDA, ANDB	And memory with accumulator
ASL, ASLA, ASLB	Arithmetic shift of accumulator or memory left
ASR, ASRA, ASRB	Arithmetic shift of accumulator or memory right
BITA, BITB	Bit test memory with accumulator
CLR, CLRA, CLRB	Clear accumulator or memory location
CMPA, CMPB	Compare memory from accumulator
COM, COMA, COMB	Complement accumulator or memory location
DAA	Decimal adjust A accumulator
DEC, DECA, DECB	Decrement accumulator or memory location
EORA, EORB	Exclusive or memory with accumulator
EXG R1, R2	Exchange R1 with R2 (R1, R2 = A, B, CC, DP)
INC, INCA, INCB	Increment accumulator or memory location
LDA, LDB	Load accumulator from memory
LSL, LSLA, LSLB	Logical shift left accumulator or memory location
LSR, LSRA, LSRB	Logical shift right accumulator or memory location
MUL	Unsigned multiply (A × B → D)
NEG, NEGA, NEGB	Negate accumulator or memory
ORA, ORB	Or memory with accumulator
ROL, ROLA, ROLB	Rotate accumulator or memory left
ROR, RORA, RORB	Rotate accumulator or memory right
SBCA, SBCB	Subtract memory from accumulator with borrow
STA, STB	Store accumulator to memory
SUBA, SUBB	Subtract memory from accumulator
TST, TSTA, TSTB	Test accumulator or memory location
TFR R1, R2	Transfer R1 to R2 (R1, R2 = A, B, CC, DP)

NOTE: A, B, CC or DP may be pushed to (pulled from) either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 5 — 16-BIT ACCUMULATOR AND MEMORY INSTRUCTIONS

Mnemonic(s)	Operation
ADDD	Add memory to D accumulator
CMPD	Compare memory from D accumulator
EXG D, R	Exchange D with X, Y, S, U or PC
LDD	Load D accumulator from memory
SEX	Sign Extend B accumulator into A accumulator
STD	Store D accumulator to memory
SUBD	Subtract memory from D accumulator
TFR D, R	Transfer D to X, Y, S, U or PC
TFR R, D	Transfer X, Y, S, U or PC to D

NOTE: D may be pushed (pulled) to either stack with PSHS, PSHU (PULS, PULU) instructions.

TABLE 6 — INDEX REGISTER/STACK POINTER INSTRUCTIONS

Instruction	Description
CMPS, CMPU	Compare memory from stack pointer
CMPX, CMPY	Compare memory from index register
EXG R1, R2	Exchange D, X, Y, S, U or PC with D, X, Y, S, U or PC
LEAS, LEAU	Load effective address into stack pointer
LEAX, LEAY	Load effective address into index register
LDS, LDU	Load stack pointer from memory
LDX, LDY	Load index register from memory
PSHS	Push A, B, CC, DP, D, X, Y, U, or PC onto hardware stack
PSHU	Push A, B, CC, DP, D, X, Y, S, or PC onto user stack
PULS	Pull A, B, CC, DP, D, X, Y, U or PC from hardware stack
PULU	Pull A, B, CC, DP, D, X, Y, S or PC from hardware stack
STS, STU	Store stack pointer to memory
STX, STY	Store index register to memory
TFR R1, R2	Transfer D, X, Y, S, U or PC to D, X, Y, S, U or PC
ABX	Add B accumulator to X (unsigned)

TABLE 7 — BRANCH INSTRUCTIONS

Instruction	Description
	SIMPLE BRANCHES
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BMI, LBMI	Branch if minus
BPL, LBPL	Branch if plus
BCS, LBCS	Branch if carry set
BCC, LBCC	Branch if carry clear
BVS, LBVS	Branch if overflow set
BVC, LBVC	Branch if overflow clear
	SIGNED BRANCHES
BGT, LBGT	Branch if greater (signed)
BVS, LBVS	Branch if invalid 2's complement result
BGE, LBGE	Branch if greater than or equal (signed)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLE, LBLE	Branch if less than or equal (signed)
BVC, LBVC	Branch if valid 2's complement result
BLT, LBLT	Branch if less than (signed)
	UNSIGNED BRANCHES
BHI, LBHI	Branch if higher (unsigned)
BCC, LBCC	Branch if higher or same (unsigned)
BHS, LBHS	Branch if higher or same (unsigned)
BEQ, LBEQ	Branch if equal
BNE, LBNE	Branch if not equal
BLS, LBLS	Branch if lower or same (unsigned)
BCS, LBCS	Branch if lower (unsigned)
BLO, LBLO	Branch if lower (unsigned)
	OTHER BRANCHES
BSR, LBSR	Branch to subroutine
BRA, LBRA	Branch always
BRN, LBRN	Branch never

TABLE 8 — MISCELLANEOUS INSTRUCTIONS

Instruction	Description
ANDCC	AND condition code register
CWAI	AND condition code register, then wait for interrupt
NOP	No operation
ORCC	OR condition code register
JMP	Jump
JSR	Jump to subroutine
RTI	Return from interrupt
RTS	Return from subroutine
SWI, SWI2, SWI3	Software interrupt (absolute indirect)
SYNC	Synchronize with interrupt line

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES

OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#	OP	Mnem	Mode	~	#
00	NEG	Direct	6	2	30	LEAX	Indexed	4+	2+	60	NEG	Indexed	6+	2+
01	•	↑			31	LEAY	↑	4+	2+	61	•	↑		
02	•				32	LEAS	↓	4+	2+	62	•			
03	COM		6	2	33	LEAU	Indexed	4+	2+	63	COM		6+	2+
04	LSR		6	2	34	PSHS	Inherent	5+	2	64	LSR		6+	2+
05	•				35	PULS	↑	5+	2	65	•			
06	ROR		6	2	36	PSHU		5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU		5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	•				68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS		5	1	69	ROL		6+	2+
0A	DEC		6	2	3A	ABX		3	1	6A	DEC		6+	2+
0B	•				3B	RTI		6/15	1	6B	•			
0C	INC		6	2	3C	CWAI		≥20	2	6C	INC		6+	2+
0D	TST		6	2	3D	MUL		11	1	6D	TST		6+	2+
0E	JMP	↓	3	2	3E	•	↓			6E	JMP	↓	3+	2+
0F	CLR	Direct	6	2	3F	SWI	Inherent	19	1	6F	CLR	Indexed	6+	2+
10	Page 2	–	–	–	40	NEGA	Inherent	2	1	70	NEG	Extended	7	3
11	Page 3	–	–	–	41	•	↑			71	•	↑		
12	NOP	Inherent	2	1	42	•				72	•			
13	SYNC	Inherent	≥4	1	43	COMA		2	1	73	COM		7	3
14	•				44	LSRA		2	1	74	LSR		7	3
15	•				45	•				75	•			
16	LBRA	Relative	5	3	46	RORA		2	1	76	ROR		7	3
17	LBSR	Relative	9	3	47	ASRA		2	1	77	ASR		7	3
18	•				48	ASLA, LSLA		2	1	78	ASL, LSL		7	3
19	DAA	Inherent	2	1	49	ROLA		2	1	79	ROL		7	3
1A	ORCC	Immed	3	2	4A	DECA		2	1	7A	DEC		7	3
1B	•	–			4B	•				7B	•			
1C	ANDCC	Immed	3	2	4C	INCA		2	1	7C	INC		7	3
1D	SEX	Inherent	2	1	4D	TSTA		2	1	7D	TST		7	3
1E	EXG	↕	8	2	4E	•	↓			7E	JMP	↓	4	3
1F	TFR	Inherent	6	2	4F	CLRA	Inherent	2	1	7F	CLR	Extended	7	3
20	BRA	Relative	3	2	50	NEGB	Inherent	2	1	80	SUBA	Immed	2	2
21	BRN	↑	3	2	51	•	↑			81	CMPA	↑	2	2
22	BHI		3	2	52	•				82	SBCA		2	2
23	BLS		3	2	53	COMB		2	1	83	SUBD		4	3
24	BHS, BCC		3	2	54	LSRB		2	1	84	ANDA		2	2
25	BLO, BCS		3	2	55	•				85	BITA		2	2
26	BNE		3	2	56	RORB		2	1	86	LDA		2	2
27	BEQ		3	2	57	ASRB		2	1	87	•			
28	BVC		3	2	58	ASLB, LSLB		2	1	88	EORA		2	2
29	BVS		3	2	59	ROLB		2	1	89	ADCA		2	2
2A	BPL		3	2	5A	DECB		2	1	8A	ORA		2	2
2B	BMI		3	2	5B	•				8B	ADDA	↓	2	2
2C	BGE		3	2	5C	INCB		2	1	8C	CMPX	Immed	4	3
2D	BLT		3	2	5D	TSTB		2	1	8D	BSR	Relative	7	2
2E	BGT	↓	3	2	5E	•	↓			8E	LDX	Immed	3	3
2F	BLE	Relative	3	2	5F	CLRB	Inherent	2	1	8F	•			

LEGEND

~ Number of MPU cycles (less possible push pull or indexed-mode cycles)
# Number of program bytes
• Denotes unused opcode

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	ANDA		4+	2+
A5	BITA		4+	2+
A6	LDA		4+	2+
A7	STA		4+	2+
A8	EORA		4+	2+
A9	ADCA		4+	2+
AA	ORA		4+	2+
AB	ADDA		4+	2+
AC	CMPX		6+	2+
AD	JSR		7+	2+
AE	LDX		5+	2+
AF	STX	Indexed	5+	2+
B0	SUBA	Extended	5	3
B1	CMPA		5	3
B2	SBCA		5	3
B3	SUBD		7	3
B4	ANDA		5	3
B5	BITA		5	3
B6	LDA		5	3
B7	STA		5	3
B8	EORA		5	3
B9	ADCA		5	3
BA	ORA		5	3
BB	ADDA		5	3
BC	CMPX		7	3
BD	JSR		8	3
BE	LDX		6	3
BF	STX	Extended	6	3

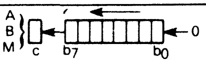
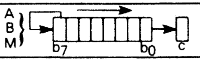
NOTE: All unused opcodes are both undefined and illegal

OP	Mnem	Mode	~	#
C0	SUBB	Immed	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB	Immed	2	2
C6	LDB	Immed	2	2
C7	•			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	•			
CE	LDU	Immed	3	3
CF	•			
D0	SUBB	Direct	4	2
D1	CMPB		4	2
D2	SBCB		4	2
D3	ADDD		6	2
D4	ANDB		4	2
D5	BITB		4	2
D6	LDB		4	2
D7	STB		4	2
D8	EORB		4	2
D9	ADCB		4	2
DA	ORB		4	2
DB	ADDB		4	2
DC	LDD		5	2
DD	STD		5	2
DE	LDU		5	2
DF	STU	Direct	5	2
E0	SUBB	Indexed	4+	2+
E1	CMPB		4+	2+
E2	SBCB		4+	2+
E3	ADDD		6+	2+
E4	ANDB		4+	2+
E5	BITB		4+	2+
E6	LDB		4+	2+
E7	STB		4+	2+
E8	EORB		4+	2+
E9	ADCB		4+	2+
EA	ORB		4+	2+
EB	ADDB		4+	2+
EC	LDD		5+	2+
ED	STD		5+	2+
EE	LDU		5+	2+
EF	STU	Indexed	5+	2+
F0	SUBB	Extended	5	3
F1	CMPB		5	3
F2	SBCB		5	3
F3	ADDD		7	3
F4	ANDB		5	3
F5	BITB		5	3
F6	LDB		5	3
F7	STB		5	3
F8	EORB		5	3
F9	ADCB		5	3
FA	ORB		5	3
FB	ADDB	Extended	5	3
FC	LDD	Extended	6	3
FD	STD		6	3
FE	LDU		6	3
FF	STU	Extended	6	3

Page 2 and 3 Machine Codes

OP	Mnem	Mode	~	#
1021	LBRN	Relative	5	4
1022	LBHI		5(6)	4
1023	LBLS		5(6)	4
1024	LBHS, LBCC		5(6)	4
1025	LBCS, LBLO		5(6)	4
1026	LBNE		5(6)	4
1027	LBEQ		5(6)	4
1028	LBVC		5(6)	4
1029	LBVS		5(6)	4
102A	LBPL		5(6)	4
102B	LBMI		5(6)	4
102C	LBGE		5(6)	4
102D	LBLT		5(6)	4
102E	LBGT		5(6)	4
102F	LBLE	Relative	5(6)	4
103F	SWI2	Inherent	20	2
1083	CMPD	Immed	5	4
108C	CMPY		5	4
108E	LDY	Immed	4	4
1093	CMPD	Direct	7	3
109C	CMPY		7	3
109E	LDY		6	3
109F	STY	Direct	6	3
10A3	CMPD	Indexed	7+	3+
10AC	CMPY		7+	3+
10AE	LDY		6+	3+
10AF	STY	Indexed	6+	3+
10B3	CMPD	Extended	8	4
10BC	CMPY		8	4
10BE	LDY		7	4
10BF	STY	Extended	7	4
10CE	LDS	Immed	4	4
10DE	LDS	Direct	6	3
10DF	STS	Direct	6	3
10EE	LDS	Indexed	6+	3+
10EF	STS	Indexed	6+	3+
10FE	LDS	Extended	7	4
10FF	STS	Extended	7	4
113F	SWI3	Inherent	20	2
1183	CMPU	Immed	5	4
118C	CMPS	Immed	5	4
1193	CMPU	Direct	7	3
119C	CMPS	Direct	7	3
11A3	CMPU	Indexed	7+	3+
11AC	CMPS	Indexed	7+	3+
11B3	CMPU	Extended	8	4
11BC	CMPS	Extended	8	4

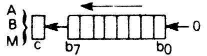
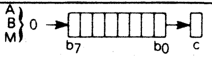
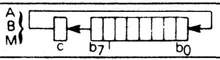
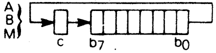
FIGURE 20 — PROGRAMMING AID

		Addressing Modes																				
		Immediate			Direct			Indexed			Extended			Inherent				5	3	2	1	0
Instruction	Forms	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Description	H	N	Z	V	C
ABX														3A	3	1	B + X → X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4 +	2 +	B9	5	3				A + M + C → A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	D9	4	2	E9	4 +	2 +	F9	5	3				B + M + C → B	↕	↕	↕	↕	↕
ADD	ADDA	8B	2	2	9B	4	2	AB	4 +	2 +	BB	5	3				A + M → A	↕	↕	↕	↕	↕
	ADDB	CB	2	2	DB	4	2	EB	4 +	2 +	FB	5	3				B + M → B	↕	↕	↕	↕	↕
	ADDD	C3	4	3	D3	6	2	E3	6 +	2 +	F3	7	3				D + M:M + 1 → D	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	4	2	A4	4 +	2 +	B4	5	3				A Λ M → A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4 +	2 +	F4	5	3				B Λ M → B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC Λ IMM → CC					7
ASL	ASLA													48	2	1		8	↕	↕	↕	↕
	ASLB													58	2	1		8	↕	↕	↕	↕
	ASL				08	6	2	68	6 +	2 +	78	7	3					8	↕	↕	↕	↕
ASR	ASRA													47	2	1		8	↕	↕	•	↕
	ASRB													57	2	1		8	↕	↕	•	↕
	ASR				07	6	2	67	6 +	2 +	77	7	3					8	↕	↕	•	↕
BIT	BITA	85	2	2	95	4	2	A5	4 +	2 +	B5	5	3				Bit Test A (M Λ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4 +	2 +	F5	5	3				Bit Test B (M Λ B)	•	↕	↕	0	•
CLR	CLRA													4F	2	1	0 → A	•	0	1	0	0
	CLRB													5F	2	1	0 → B	•	0	1	0	0
	CLR				0F	6	2	6F	6 +	2 +	7F	7	3				0 → M	•	0	1	0	0
CMP	CMPA	81	2	2	91	4	2	A1	4 +	2 +	B1	5	3				Compare M from A	8	↕	↕	↕	↕
	CMPB	C1	2	2	D1	4	2	E1	4 +	2 +	F1	5	3				Compare M from B	8	↕	↕	↕	↕
	CMPD	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from D	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPS	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from S	•	↕	↕	↕	↕
		8C			9C			AC			BC											
	CMPU	11	5	4	11	7	3	11	7 +	3 +	11	8	4				Compare M:M + 1 from U	•	↕	↕	↕	↕
		83			93			A3			B3											
	CMPX	8C	4	3	9C	6	2	AC	6 +	2 +	BC	7	3				Compare M:M + 1 from X	•	↕	↕	↕	↕
	CMPY	10	5	4	10	7	3	10	7 +	3 +	10	8	4				Compare M:M + 1 from Y	•	↕	↕	↕	↕
		8C			9C			AC			BC											
COM	COMA													43	2	1	$\overline{A}$ → A	•	↕	↕	0	1
	COMB													53	2	1	$\overline{B}$ → B	•	↕	↕	0	1
	COM				03	6	2	63	6 +	2 +	73	7	3				$\overline{M}$ → M	•	↕	↕	0	1
CWAI		3C	≥20	2													CC Λ IMM → CC Wait for Interrupt					7
DAA														19	2	1	Decimal Adjust A	•	↕	↕	0	↕
DEC	DECA													4A	2	1	A − 1 → A	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	↕	↕	↕	•
	DEC				0A	6	2	6A	6 +	2 +	7A	7	3				M − 1 → M	•	↕	↕	↕	•
EOR	EORA	88	2	2	98	4	2	A8	4 +	2 +	B8	5	3				A ⊻ M → A	•	↕	↕	0	•
	EORB	C8	2	2	D8	4	2	E8	4 +	2 +	F8	5	3				B ⊻ M → B	•	↕	↕	0	•
EXG	R1, R2													1E	8	2	R1 → R2^{2}	•	•	•	•	•
INC	INCA													4C	2	1	A + 1 → A	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	↕	↕	↕	•
	INC				0C	6	2	6C	6 +	2 +	7C	7	3				M + 1 → M	•	↕	↕	↕	•
JMP					0E	3	2	6E	3 +	2 +	7E	4	3				EA^{3} → PC	•	•	•	•	•
JSR					9D	7	2	AD	7 +	2 +	BD	8	3				Jump to Subroutine	•	•	•	•	•
LD	LDA	86	2	2	96	4	2	A6	4 +	2 +	B6	5	3				M → A	•	↕	↕	0	•
	LDB	C6	2	2	D6	4	2	E6	4 +	2 +	F6	5	3				M → B	•	↕	↕	0	•
	LDD	CC	3	3	DC	5	2	EC	5 +	2 +	FC	6	3				M:M + 1 → D	•	↕	↕	0	•
	LDS	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → S	•	↕	↕	0	•
		CE			DE			EE			FE											
	LDU	CE	3	3	DE	5	2	EE	5 +	2 +	FE	6	3				M:M + 1 → U	•	↕	↕	0	•
	LDX	8E	3	3	9E	5	2	AE	5 +	2 +	BE	6	3				M:M + 1 → X	•	↕	↕	0	•
	LDY	10	4	4	10	6	3	10	6 +	3 +	10	7	4				M:M + 1 → Y	•	↕	↕	0	•
		8E			9E			AE			BE											
LEA	LEAS							32	4 +	2 +							EA^{3} → S	•	•	•	•	•
	LEAU							33	4 +	2 +							EA^{3} → U	•	•	•	•	•
	LEAX							30	4 +	2 +							EA^{3} → X	•	•	↕	•	•
	LEAY							31	4 +	2 +							EA^{3} → Y	•	•	↕	•	•

Legend:

OP	Operation Code (Hexadecimal)
~	Number of MPU Cycles
#	Number of Program Bytes
+	Arithmetic Plus
−	Arithmetic Minus
•	Multiply
$\overline{M}$	Complement of M
→	Transfer Into
H	Half-carry (from bit 3)
N	Negative (sign bit)
Z	Zero result
V	Overflow, 2's complement
C	Carry from ALU
↕	Test and set if true, cleared otherwise
•	Not Affected
CC	Condition Code Register
:	Concatenation
V	Logical or
Λ	Logical and
⊻	Logical Exclusive or

FIGURE 20 — PROGRAMMING AID (CONTINUED)

		Addressing Modes																5	3	2	1	0
		Immediate			Direct			Indexed[1]			Extended			Inherent								
Instruction	**Forms**	**Op**	**~**	**#**	**Op**	**~**	**#**	**Op**	**~**	**#**	**Op**	**~**	**#**	**Op**	**~**	**#**	**Description**	**H**	**N**	**Z**	**V**	**C**
LSL	LSLA													48	2	1		•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1		•	0	↕	•	↕
	LSRB													54	2	1		•	0	↕	•	↕
	LSR				04	6	2	64	6+	2+	74	7	3					•	0	↕	•	↕
MUL														3D	11	1	A × B → D (Unsigned)	•	•	↕	•	9
NEG	NEGA													40	2	1	$\overline{A}$ + 1 → A	8	↕	↕	↕	↕
	NEGB													50	2	1	$\overline{B}$ + 1 → B	8	↕	↕	↕	↕
	NEG				00	6	2	60	6+	2+	70	7	3				$\overline{M}$ + 1 → M	8	↕	↕	↕	↕
NOP														12	2	1	No Operation	•	•	•	•	•
OR	ORA	8A	2	2	9A	4	2	AA	4+	2+	BA	5	3				A V M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B V M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC V IMM → CC				7	
PSH	PSHS	34	5+[4]	2													Push Registers on S Stack	•	•	•	•	•
	PSHU	36	5+[4]	2													Push Registers on U Stack	•	•	•	•	•
PUL	PULS	35	5+[4]	2													Pull Registers from S Stack	•	•	•	•	•
	PULU	37	5+[4]	2													Pull Registers from U Stack	•	•	•	•	•
ROL	ROLA													49	2	1		•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1		•	↕	↕	•	↕
	RORB													56	2	1		•	↕	↕	•	↕
	ROR				06	6	2	66	6+	2+	76	7	3					•	↕	↕	•	↕
RTI														3B	6/15	1	Return From Interrupt					7
RTS														39	5	1	Return from Subroutine	•	•	•	•	•
SBC	SBCA	82	2	2	92	4	2	A2	4+	2+	B2	5	3				A − M − C → A	8	↕	↕	↕	↕
	SBCB	C2	2	2	D2	4	2	E2	4+	2+	F2	5	3				B − M − C → B	8	↕	↕	↕	↕
SEX														1D	2	1	Sign Extend B into A	•	↕	↕	0	•
ST	STA				97	4	2	A7	4+	2+	B7	5	3				A → M	•	↕	↕	0	•
	STB				D7	4	2	E7	4+	2+	F7	5	3				B → M	•	↕	↕	0	•
	STD				DD	5	2	ED	5+	2+	FD	6	3				D → M:M + 1	•	↕	↕	0	•
	STS				10 DF	6	3	10 EF	6+	3+	10 FF	7	4				S → M:M + 1	•	↕	↕	0	•
	STU				DF	5	2	EF	5+	2+	FF	6	3				U → M:M + 1	•	↕	↕	0	•
	STX				9F	5	2	AF	5+	2+	BF	6	3				X → M:M + 1	•	↕	↕	0	•
	STY				10 9F	6	3	10 AF	6+	3+	10 BF	7	4				Y → M:M + 1	•	↕	↕	0	•
SUB	SUBA	80	2	2	90	4	2	A0	4+	2+	B0	5	3				A − M → A	8	↕	↕	↕	↕
	SUBB	C0	2	2	D0	4	2	E0	4+	2+	F0	5	3				B − M → B	8	↕	↕	↕	↕
	SUBD	83	4	3	93	6	2	A3	6+	2+	B3	7	3				D − M:M + 1 → D	•	↕	↕	↕	↕
SWI	SWI[6]													3F	19	1	Software Interrupt 1	•	•	•	•	•
	SWI[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI[6]													11 3F	20	1	Software Interrupt 3	•	•	•	•	•
SYNC														13	≥4	1	Synchronize to Interrupt	•	•	•	•	•
TFR	R1, R2													1F	6	2	R1 → R2[2]	•	•	•	•	•
TST	TSTA													4D	2	1	Test A	•	↕	↕	0	•
	TSTB													5D	2	1	Test B	•	↕	↕	0	•
	TST				0D	6	2	6D	6+	2+	7D	7	3				Test M	•	↕	↕	0	•

Notes:

1. This column gives a base cycle and byte count. To obtain total count, add the values obtained from the INDEXED ADDRESSING MODE table, Table 2.
2. R1 and R2 may be any pair of 8 bit or any pair of 16 bit registers.
   The 8 bit registers are: A, B, CC, DP
   The 16 bit registers are: X, Y, U, S, D, PC
3. EA is the effective address.
4. The PSH and PUL instructions require 5 cycles plus 1 cycle for each **byte** pushed or pulled.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken (Branch instructions).
6. SWI sets I and F bits. SWI2 and SWI3 do not affect I and F.
7. Conditions Codes set as a direct result of the instruction.
8. Vaue of half-carry flag is undefined.
9. Special Case — Carry set if b7 is SET.

FIGURE 20 — PROGRAMMING AID (CONTINUED)

Branch Instructions

		Addressing Mode Relative				5	3	2	1	0
Instruction	Forms	OP	~ 5	#	Description	H	N	Z	V	C
BCC	BCC	24	3	2	Branch C = 0	•	•	•	•	•
	LBCC	10 24	5(6)	4	Long Branch C = 0	•	•	•	•	•
BCS	BCS	25	3	2	Branch C = 1	•	•	•	•	•
	LBCS	10 25	5(6)	4	Long Branch C = 1	•	•	•	•	•
BEQ	BEQ	27	3	2	Branch Z = 1	•	•	•	•	•
	LBEQ	10 27	5(6)	4	Long Branch Z = 0	•	•	•	•	•
BGE	BGE	2C	3	2	Branch ≥ Zero	•	•	•	•	•
	LBGE	10 2C	5(6)	4	Long Branch ≥ Zero	•	•	•	•	•
BGT	BGT	2E	3	2	Branch > Zero	•	•	•	•	•
	LBGT	10 2E	5(6)	4	Long Branch > Zero	•	•	•	•	•
BHI	BHI	22	3	2	Branch Higher	•	•	•	•	•
	LBHI	10 22	5(6)	4	Long Branch Higher	•	•	•	•	•
BHS	BHS	24	3	2	Branch Higher or Same	•	•	•	•	•
	LBHS	10 24	5(6)	4	Long Branch Higher or Same	•	•	•	•	•
BLE	BLE	2F	3	2	Branch ≤ Zero	•	•	•	•	•
	LBLE	10 2F	5(6)	4	Long Branch ≤ Zero	•	•	•	•	•
BLO	BLO	25	3	2	Branch lower	•	•	•	•	•
	LBLO	10 25	5(6)	4	Long Branch Lower	•	•	•	•	•

		Addressing Mode Relative				5	3	2	1	0
Instruction	Forms	OP	~ 5	#	Description	H	N	Z	V	C
BLS	BLS	23	3	2	Branch Lower or Same	•	•	•	•	•
	LBLS	10 23	5(6)	4	Long Branch Lower or Same	•	•	•	•	•
BLT	BLT	2D	3	2	Branch < Zero	•	•	•	•	•
	LBLT	10 2D	5(6)	4	Long Branch < Zero	•	•	•	•	•
BMI	BMI	2B	3	2	Branch Minus	•	•	•	•	•
	LBMI	10 2B	5(6)	4	Long Branch Minus	•	•	•	•	•
BNE	BNE	26	3	2	Branch Z = 0	•	•	•	•	•
	LBNE	10 26	5(6)	4	Long Branch Z ≠ 0	•	•	•	•	•
BPL	BPL	2A	3	2	Branch Plus	•	•	•	•	•
	LBPL	10 2A	5(6)	4	Long Branch Plus	•	•	•	•	•
BRA	BRA	20	3	2	Branch Always	•	•	•	•	•
	LBRA	16	5	3	Long Branch Always	•	•	•	•	•
BRN	BRN	21	3	2	Branch Never	•	•	•	•	•
	LBRN	10 21	5	4	Long Branch Never	•	•	•	•	•
BSR	BSR	8D	7	2	Branch to Subroutine	•	•	•	•	•
	LBSR	17	9	3	Long Branch to Subroutine	•	•	•	•	•
BVC	BVC	28	3	2	Branch V = 0	•	•	•	•	•
	LBVC	10 28	5(6)	4	Long Branch V = 0	•	•	•	•	•
BVS	BVS	29	3	2	Branch V = 1	•	•	•	•	•
	LBVS	10 29	5(6)	4	Long Branch V = 1	•	•	•	•	•

SIMPLE BRANCHES

	OP	~	#
BRA	20	3	2
LBRA	16	5	3
BRN	21	3	2
LBRN	1021	5	4
BSR	8D	7	2
LBSR	17	9	3

SIMPLE CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
N = 1	BMI	2B	BPL	2A
Z = 1	BEQ	27	BNE	26
V = 1	BVS	29	BVC	28
C = 1	BCS	25	BCC	24

SIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BGT	2E	BLE	2F
r ≥ m	BGE	2C	BLT	2D
r = m	BEQ	27	BNE	26
r ≤ m	BLE	2F	BGT	2E
r < m	BLT	2D	BGE	2C

UNSIGNED CONDITIONAL BRANCHES (Notes 1-4)

Test	True	OP	False	OP
r > m	BHI	22	BLS	23
r ≥ m	BHS	24	BLO	25
r = m	BEQ	27	BNE	26
r ≤ m	BLS	23	BHI	22
r < m	BLO	25	BHS	24

Notes:

1. All conditional branches have both short and long variations.
2. All short branches are 2 bytes and require 3 cycles.
3. All conditional long branches are formed by prefixing the short branch opcode with $10 and using a 16-bit destination offset.
4. All conditional long branches require 4 bytes and 6 cycles if the branch is taken or 5 cycles if the branch is not taken.
5. 5(6) means: 5 cycles if branch not taken, 6 cycles if taken.

INDEXED ADDRESSING MODES

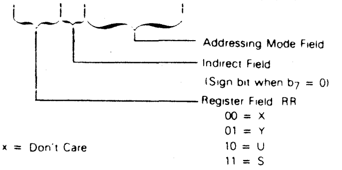
TYPE	FORMS	NON INDIRECT Assembler Form	Post-Byte OP Code	+	+ #	INDIRECT Assembler Form	Post-Byte OP Code	+	+ #
CONSTANT OFFSET FROM R	NO OFFSET	,R	1RR00100	0	0	[,R]	1RR10100	3	0
	5 BIT OFFSET	n, R	0RRnnnnn	1	0	defaults to 8-bit			
	8 BIT OFFSET	n, R	1RR01000	1	1	[n, R]	1RR11000	4	1
	16 BIT OFFSET	n, R	1RR01001	4	2	[n, R]	1RR11001	7	2
ACCUMULATOR OFFSET FROM R	A—REGISTER OFFSET	A, R	1RR00110	1	0	[A, R]	1RR10110	4	0
	B—REGISTER OFFSET	B, R	1RR00101	1	0	[B, R]	1RR10101	4	0
	D—REGISTER OFFSET	D, R	1RR01011	4	0	[D, R]	1RR11011	7	0
AUTO INCREMENT/DECREMENT R	INCREMENT BY 1	,R+	1RR00000	2	0	not allowed			
	INCREMENT BY 2	,R++	1RR00001	3	0	[,R++]	1RR10001	6	0
	DECREMENT BY 1	,-R	1RR00010	2	0	not allowed			
	DECREMENT BY 2	,--R	1RR00011	3	0	[,--R]	1RR10011	6	0
CONSTANT OFFSET FROM PC	8 BIT OFFSET	n, PCR	1XX01100	1	1	[n, PCR]	1XX11100	4	1
	16 BIT OFFSET	n, PCR	1XX01101	5	2	[n, PCR]	1XX11101	8	2
EXTENDED INDIRECT	16 BIT ADDRESS	—	—	-	-	[n]	10011111	5	2

R = X, Y, U, or S
X = DON'T CARE

RR: 00 = X 10 = U
01 = Y 11 = S

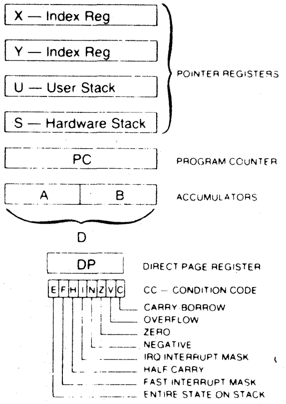
INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

POST-BYTE REGISTER BIT 7	6	5	4	3	2	1	0	INDEXED ADDRESSING MODE
0	R	R	x	x	x	x	x	EA = ,R + 5 Bit Offset
1	R	R	0	0	0	0	0	,R +
1	R	R	I	0	0	0	1	,R + +
1	R	R	0	0	0	1	0	,– R
1	R	R	I	0	0	1	1	,– – R
1	R	R	I	0	1	0	0	EA = ,R + 0 Offset
1	R	R	I	0	1	0	1	EA = ,R + ACCB Offset
1	R	R	I	0	1	1	0	EA = ,R + ACCA Offset
1	R	R	I	1	0	0	0	EA = ,R + 8 Bit Offset
1	R	R	I	1	0	0	1	EA = ,R + 16 Bit Offset
1	R	R	I	1	0	1	1	EA = ,R + D Offset
1	x	x	I	1	1	0	0	EA = ,PC + 8 Bit Offset
1	x	x	I	1	1	0	1	EA = ,PC + 16 Bit Offset
1	R	R	1	1	1	1	1	EA = [,Address]

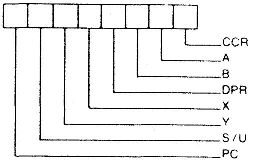


x = Don't Care

6809 PROGRAMMING MODEL



PUSH/PULL POST BYTE



6809 STACKING ORDER

PULL ORDER

CC
A
B
DP
X Hi
X Lo
Y Hi
Y Lo
U/S Hi
U/S Lo
PC Hi
PC Lo

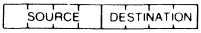
PUSH ORDER

INCREASING MEMORY

6809 VECTORS

FFFE	Restart
FFFC	NMI
FFFA	SWI
FFF8	IRQ
FFF6	FIRQ
FFF4	SWI2
FFF2	SWI3
FFF0	Reserved

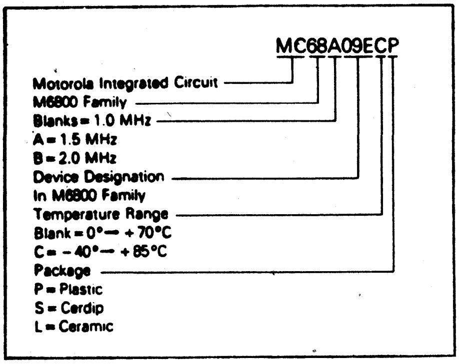
TRANSFER/EXCHANGE POST BYTE



REGISTER FIELD

0000	D (A B)	1000	A
0001	X	1001	B
0010	Y	1010	CCR
0011	U	1011	DPR
0100	S		
0101	PC		

ORDERING INFORMATION



Speed	Device	Temperature Range
1.0 MHz	MC6809EP,L,S	0 to 70°C
1.5 MHz	MC68A09EP,L,S	0 to +70°C
2.0 MHz	MC68B09EP,L,S	0 to +70°C

MOTOROLA

MC6821
(1.0 MHz)
MC68A21
(1.5 MHz)
MC68B21
(2.0 MHz)

PERIPHERAL INTERFACE ADAPTER (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

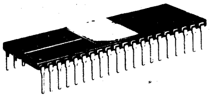
The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation

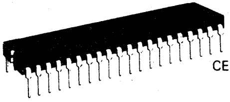
MOS

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

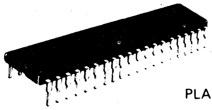
PERIPHERAL INTERFACE ADAPTER



L SUFFIX
CERAMIC PACKAGE
CASE 715

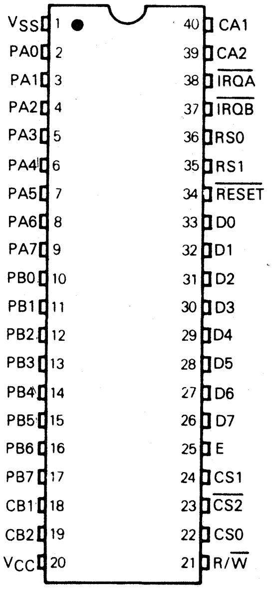


S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	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 MC6821, MC68A21, MC68B21 MC6821C, MC68A21C, MC68B21C	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 Ceramic Plastic Cerdip	θ_{JA}	 50 100 60	°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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (i.e., 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.

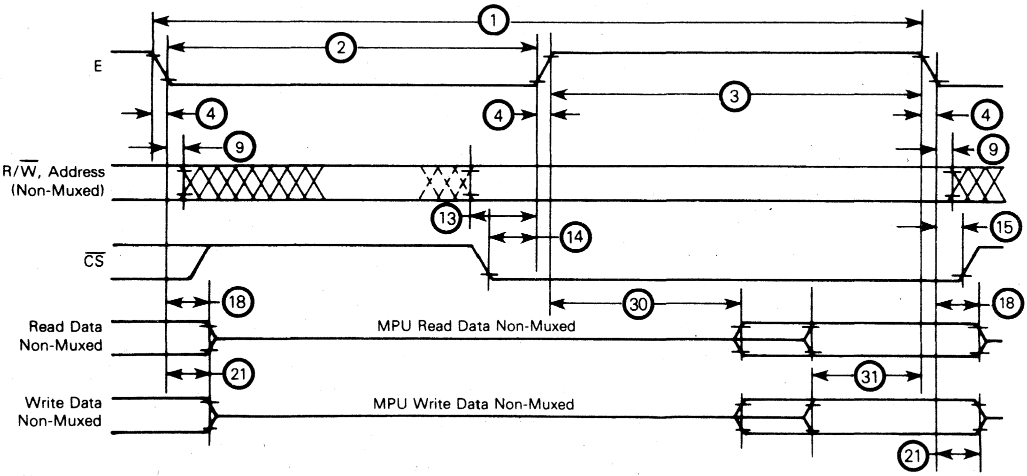
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted).

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R/$\overline{W}$, Enable, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$)					
Input High Voltage	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	I_{in}	–	1.0	2.5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	7.5	pF
INTERRUPT OUTPUTS ($\overline{IRQA}$, $\overline{IRQB}$)					
Output Low Voltage (I_{Load} = 3.2 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Three-State Output Leakage Current	I_{OZ}	–	1.0	10	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{out}	–	–	5.0	pF
DATA BUS (D0-D7)					
Input High Voltage	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Three-State Input Leakage Current (V_{in} = 0.4 to 2.4 V)	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = – 205 μA)	V_{OH}	$V_{SS}+2.4$	–	–	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	12.5	pF
PERIPHERAL BUS (PA0-PA7, PB0-PB7, CA1, CA2, CB1, CB2)					
Input Leakage Current (V_{in} = 0 to 5.25 V) — R/$\overline{W}$, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$, CA1, CB1, Enable	I_{in}	–	1.0	2.5	μA
Three-State Input Leakage Current (V_{in} = 0.4 to 2.4 V) — PB0-PB7, CB2	I_{IZ}	–	2.0	10	μA
Input High Current (V_{IH} = 2.4 V) — PA0-PA7, CA2	I_{IH}	– 200	– 400	–	μA
Darlington Drive Current (V_O = 1.5 V) — PB0-PB7, CB2	I_{OH}	– 1.0	–	– 10	mA
Input Low Current (V_{IL} = 0.4 V) — PA0-PA7, CA2	I_{IL}	–	– 1.3	– 2.4	mA
Output High Voltage (I_{Load} = – 200 μA) — PA0-PA7, PB0-PB7, CA2, CB2 (I_{Load} = – 10 μA) — PA0-PA7, CA2	V_{OH}	$V_{SS}+2.4$ $V_{CC}-1.0$	– –	– –	V
Output Low Voltage (I_{Load} = 3.2 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	10	pF
POWER REQUIREMENTS					
Internal Power Dissipation (Measured at $T_A = T_L$)	P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ms
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 1 — BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

PERIPHERAL TIMING CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 V, $T_A = T_L$ to T_H unless otherwise specified)

Characteristic	Symbol	MC6821		MC68A21		MC68B21		Unit	Reference Fig. No.
		Min	Max	Min	Max	Min	Max		
Data Setup Time	t_{PDS}	200	–	135	–	100	–	ns	6
Data Hold Time	t_{PDH}	0	–	0	–	0	–	ns	6
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	–	1.0	–	0.670	–	0.500	μs	3, 7, 8
Delay Time, Enable Negative Transition to CA2 Positive Transition	T_{RS1}	–	1.0	–	0.670	–	0.500	μs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	μs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	μs	3, 8
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	–	1.0	–	0.670	–	0.5	μs	3, 9, 10
Delay Time, Enable Negative Transition to CMOS Data Valid PA0-PA7, CA2	t_{CMOS}	–	2.0	–	1.35	–	1.0	μs	4, 9
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	–	1.0	–	0.670	–	0.5	μs	3, 11, 12
Delay Time, Data Valid to CB2 Negative Transition	t_{DC}	20	–	20	–	20	–	ns	3, 10
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	–	1.0	–	0.670	–	0.5	μs	3, 11
Control Output Pulse Width, CA2/CB2	PW_{CT}	500	–	375	–	250	–	ns	3, 11
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	μ	12
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	μs	3, 12
Interrupt Release Time, $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$	t_{IR}	–	1.60	–	1.10	–	0.85	μs	5, 14
Interrupt Response Time	t_{RS3}	–	1.0	–	1.0	–	1.0	μs	5, 13
Interrupt Input Pulse Time	PW_I	500	–	500	–	500	–	ns	13
$\overline{\text{RESET}}$ Low Time*	t_{RL}	1.0	–	0.66	–	0.5	–	μs	15

*The $\overline{\text{RESET}}$ line must be high a minimum of 1.0 μs before addressing the PIA.

FIGURE 2 — BUS TIMING TEST LOADS

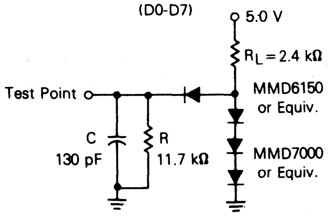
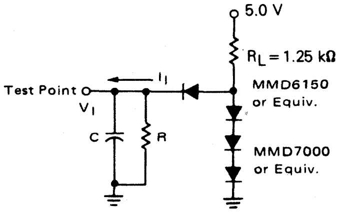


FIGURE 3 — TTL EQUIVALENT TEST LOAD

(PA0–PA7, PB0–PB7, CA2, CB2)



C = 30 pF, R = 12 k

FIGURE 4 — CMOS EQUIVALENT TEST LOAD

(PA0-PA7, CA2)

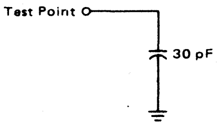


FIGURE 5 — NMOS EQUIVALENT TEST LOAD

($\overline{\text{IRQ}}$ Only)

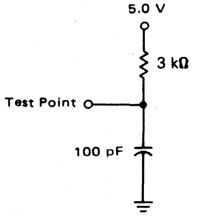


FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES (Read Mode)

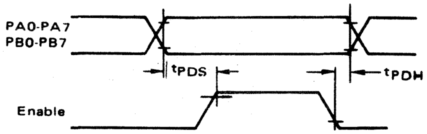
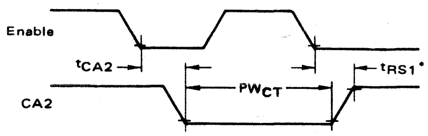


FIGURE 7 — CA2 DELAY TIME (Read Mode; CRA-5=CRA3=1, CRA-4=0)



*Assumes part was deselected during the previous E pulse.

FIGURE 8 — CA2 DELAY TIME (Read Mode; CRA-5=1, CRA-3=CRA-4=0)

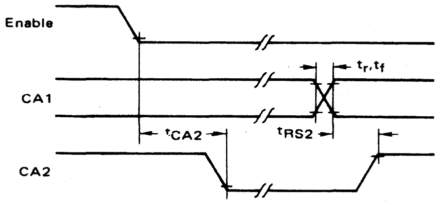


FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES (Write Mode; CRA-5=CRA-3=1, CRA-4=0)

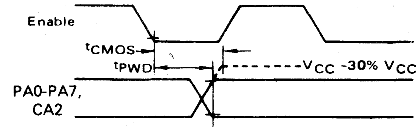
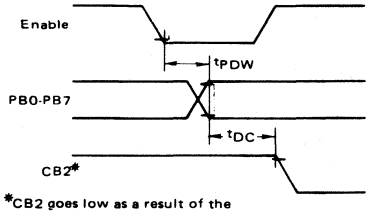
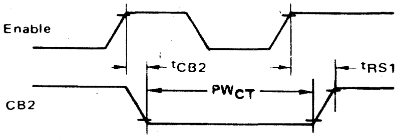


FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES (Write Mode; CRB-5=CRB-3=1, CRB-4=0)



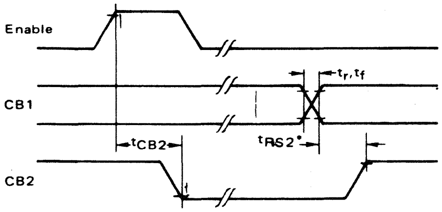
*CB2 goes low as a result of the positive transition of Enable.

FIGURE 11 — CB2 DELAY TIME (Write Mode; CRB-5=CRB-3=1, CRB-4=0)



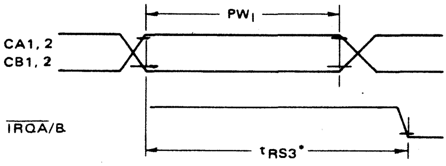
*Assumes part was deselected during the previous E pulse.

FIGURE 12 — CB2 DELAY TIME (Write Mode; CRB-5=1, CRB-3=CRB-4=0)



*Assumes part was deselected during any previous E pulse.

FIGURE 13 — INTERRUPT PULSE WIDTH AND $\overline{IRQ}$ RESPONSE



*Assumes Interrupt Enable Bits are set.

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 14 — $\overline{IRQ}$ RELEASE TIME

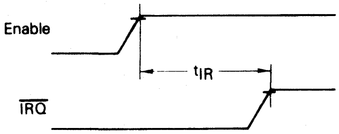
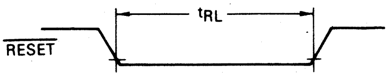


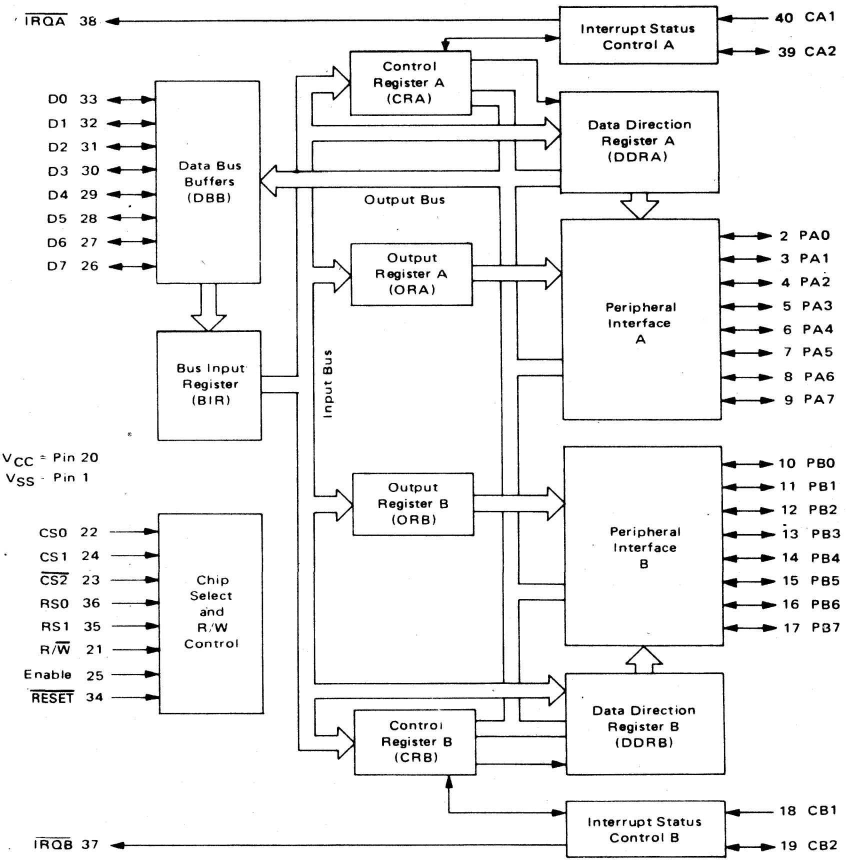
FIGURE 15 — $\overline{RESET}$ LOW TIME



*The $\overline{RESET}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the PIA.

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 16 — EXPANDED BLOCK DIAGRAM



PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the M6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802, or MC6808 microprocessors, VMA should be used as an active part of the address decoding.

Bidirectional Data (D0-D7) – The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when the PIA is selected for a read operation.

Enable (E) – The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

Read/Write (R/$\overline{\text{W}}$) – This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the PIA read/write line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

$\overline{\text{RESET}}$ – The active low $\overline{\text{RESET}}$ line is used to reset all register bits in the PIA to a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and $\overline{\text{CS2}}$) – These three input signals are used to select the PIA. CS0 and CS1 must be high and $\overline{\text{CS2}}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) – The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) – The active low Interrupt Request lines ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bidirectional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) – Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low." Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read property if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PB0-PB7) – The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7. They have three-state capabiity, allowing them to enter a high-impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines

PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low". As outputs, these lines are compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch.

Interrupt Input (CA1 and CB1) — Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) — The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL; as an input the internal pullup resistor on this line represents 1.5 standard TTL loads. The function of this signal line is programmed with Control Register A.

Peripheral Control (CB2) — Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{RESET}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 – INTERNAL ADDRESSING

		Control Register Bit		
RS1	RS0	CRA-2	CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 17, the MC6821 has a pair of I/O ports whose characteristics differ greatly. The A side is designed to drive CMOS logic to normal 30% to 70% levels, and incorporates an internal pullup device that remains connected even in the input mode. Because of this, the A side requires more drive current in the input mode than Port B. In contrast, the B side uses a normal three-state NMOS buffer which cannot pullup to CMOS levels without external resistors. The B side can drive extra loads such as Darlingtons without problem. When the PIA comes out of reset, the A port represents inputs with pullup resistors, whereas the B side (input mode also) will float high or low, depending upon the load connected to it.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) — The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) — Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-1, CRA-1, and CRB-1) — The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 17 — PORT A AND PORT B EQUIVALENT CIRCUITS

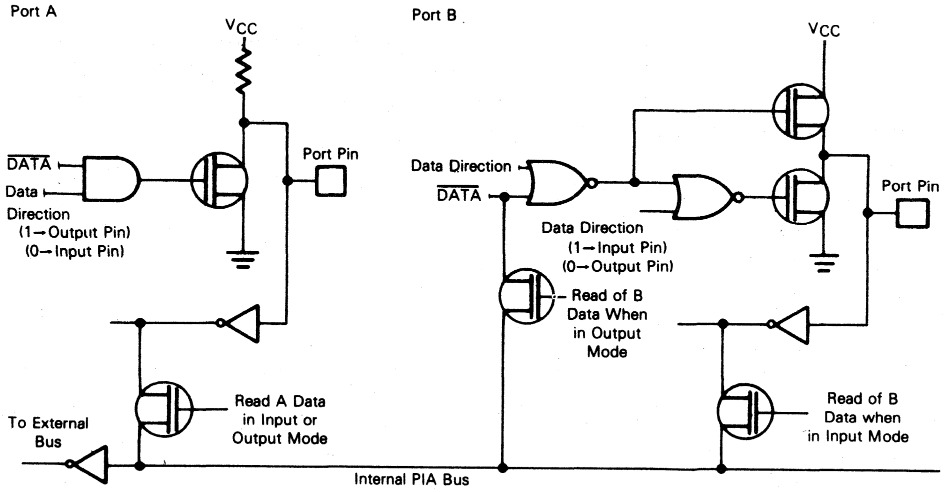


FIGURE 18 — CONTROL WORD FORMAT

Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 — (bit 7)

b1=0: IRQA(B)1 set by high-to-low transition on CA1 (CB1)

b1=1: IRQA(B)1 set by low-to-high transition on CA1 (CB1).

CA1 (CB1) Interrupt Request Enable/Disable

b0=0: Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition.[1]

b0=1: Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition.

1. IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled.

IRQA(B) 1 Interrupt Flag (bit 7)

Goes high on active transition of CA1 (CB1); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

Control Register

b7	b6	b5	b4	b3	b2	b1	b0
IRQA(B)1 Flag	IRQA(B)2 Flag	CA2 (CB2) Control			DDR Access	CA1 (CB1) Control	

IRQA(B)2 Interrupt Flag (bit 6)

When CA2 (CB2) is an input, IRQA(B) goes high on active transition CA2 (CB2); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA2 (CB2) Established as Output (b5=1): IRQA(B) 2=0, not affected by CA2 (CB2) transitions.

Determines Whether Data Direction Register Or Output Register is Addressed

b2=0: Data Direction Register selected.

b2=1: Output Register selected.

CA2 (CB2) Established as Output by b5=1

(Note that operation of CA2 and CB2 output functions are not identical)

b5 b4 b3

1 0 → CA2

b3=0: **Read Strobe with CA1 Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next active CA1 transition, as specified by bit 1.

b3=1: **Read Strobe with E Restore**
CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next high-to-low E transition during a deselect.

→ CB2

b3=0: **Write Strobe with CB1 Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next active CB1 transition as specified by bit 1. CRB-b7 must first be cleared by a read of data.

b3=1: **Write Strobe with E Restore**
CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next low-to-high E transition following an E pulse which occurred while the part was deselected.

b5 b4 b3

1 1 → **Set/Reset CA2 (CB2)**

CA2 (CB2) goes low as MPU writes b3=0 into Control Register.

CA2 (CB2) goes high as MPU writes b3=1 into Control Register.

CA2 (CB2) Established as Input by b5=0

b5 b4 b3

0

→ **CA2 (CB2) Interrupt Request Enable/Disable**

b3=0: Disables IRQA(A) MPU Interrupt by CA2 (CB2) active transition.*

b3=1: Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.

*IRQA(B) will occur on next (MPU generated) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled.

→ **Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 — (Bit b6)**

b4=0: IRQA(B)2 set by high-to-low transition on CA2 (CB2).

b4=1: IRQA(B)2 set by low-to-high transition on CA2 (CB2).

MOTOROLA

MC6822
(1.0 MHz)
MC68A22
(1.5 MHz)
MC68B22
(2.0 MHz)

INDUSTRIAL INTERFACE ADAPTER (IIA)

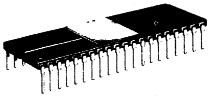
The MC6822 Industrial Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

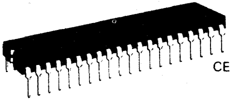
The functional configuration of the IIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

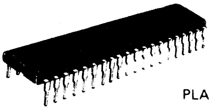
- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- **Open Drain Port Outputs**
- **High Voltage Capability up to 18V**
- **Program Controlled Interrupt and Interrupt Disable Capability**
- **Ports Output Compatible with CMOS @15V**
- TTL-Compatible
- Static Operation

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

INDUSTRIAL INTERFACE ADAPTER


L SUFFIX
CERAMIC PACKAGE
CASE 715


S SUFFIX
CERDIP PACKAGE
CASE 734


P SUFFIX
PLASTIC PACKAGE
CASE 711

MAXIMUM RATINGS

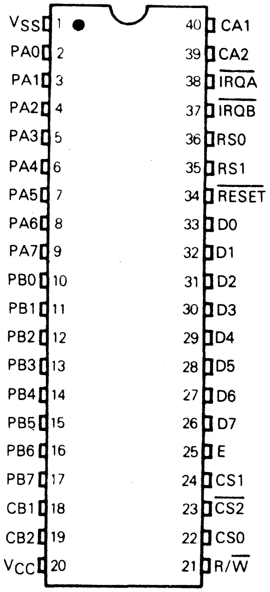
Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage Except Port Pins	V_{in}	−0.3 to +7.0	V
Operating Temperature Range **MC6822, MC68A22, MC68B22** **MC6822C, MC68A22C, MC68B22C**	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Output Port current (Sink)	I_{in}	**10**	**mA**
Input Voltage PA0-7 CA2CA1 **PB0-8 CB2CB1**	V_{pin}	**18**	V

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Cerdip	θ_{JA}	 50 100 60	°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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (i.e., either V_{SS} or V_{CC}).

PIN ASSIGNMENT



DS9435-R3

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic	Symbol	Min	Typ	Max	Unit
BUS CONTROL INPUTS (R/$\overline{W}$, Enable, $\overline{RESET}$, RS0, RS1, CS0, CS1, CS2)					
Input High Voltage	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	I_{in}	–	1.0	2.5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	7.5	pF
INTERRUPT OUTPUTS ($\overline{IRQA}$, $\overline{IRQB}$)					
Output Low Voltage (I_{Load} = 3.2 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Three-State Output Leakage Current	I_{OZ}	–	1.0	10	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{out}	–	–	5.0	pF
DATA BUS (D0-D7)					
Input High Voltage	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Three-State Input Leakage Current (V_{in} = 0.4 to 2.4 V)	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = –205 μA)	V_{OH}	$V_{SS}+2.4$	–	–	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}	–	–	12.5	pF

PERIPHERAL BUS (PA0 - 7, PB0 - 7, CA1, CA2, CB1, CB2)

Port Leakage HI (V = 16 Volts)	I_{PLKH}			10	μA
Port Leakage LO (V = 10 V)	I_{PLKL}			2.5	μA
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}			10	pF

PERIPHERAL BUS (PA0 - 7, PB0 - 7, CA2, CB2

Output Low Voltage (I_{Load} = 1mA)	V_{OL}			0.4	V
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	C_{in}			12.5	pF

POWER REQUIREMENTS

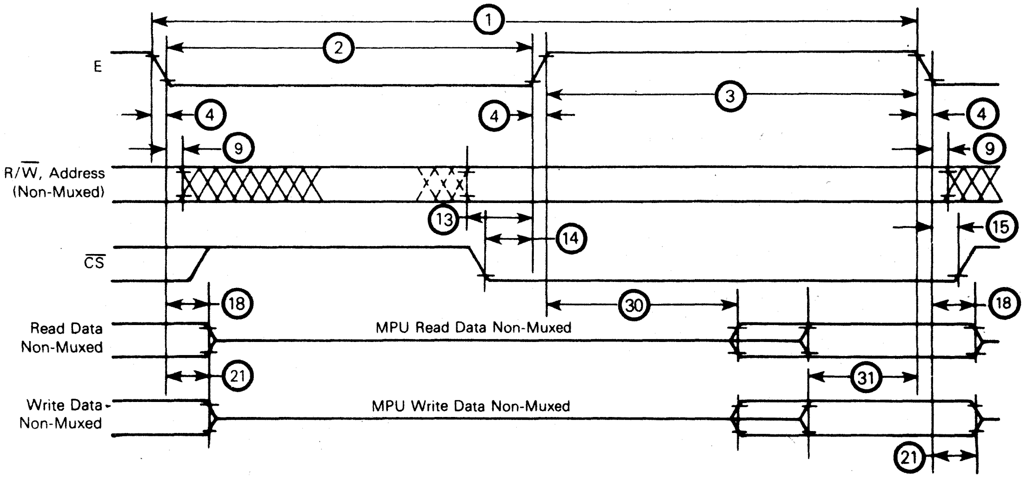
Internal Power Dissipation (Measured at $T_A = T_L$)	P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6822		MC68A22		MC68B22		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ms
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 1 — BUS TIMING



Notes:

1. Voltage levels shown are $V_L \leq 0.4$ 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.

PERIPHERAL TIMING CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0 V, $T_A = T_L$ to T_H unless otherwise specified)

Characteristic	Symbol	MC6822		MC68A22		MC68B22		Unit	Reference Fig. No.
		Min	Max	Min	Max	Min	Max		
Data Setup Time	t_{PDS}	200	–	135	–	100	–	ns	6
Data Hold Time	t_{PDH}	0	–	0	–	0	–	ns	6
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	–	1.0	–	0.670	–	0.500	µs	3, 7, 8
Delay Time, Enable Negative Transition to CA2 Positive Transition	T_{RS1}	–	1.0	–	0.670	–	0.500	µs	3, 7
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µs	8
Delay Time from CA1 Active Transition to CA2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	3, 8
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	–	1.0	–	0.670	–	0.5	µs	3, 9, 10
Delay Time, Enable Negative Transition to CMOS Data Valid PA0-PA7, CA2	t_{CMOS}	–	2.0	–	1.35	–	1.0	µs	4, 9
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	–	1.0	–	0.670	–	0.5	µs	3, 11, 12
Delay Time, Data Valid to CB2 Negative Transition	t_{DC}	20	–	20	–	20	–	ns	3, 10
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	–	1.0	–	0.670	–	0.5	µs	3, 11
Control Output Pulse Width, CA2/CB2	PW_{CT}	500	–	375	–	250	–	ns	3, 11
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µ	12
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	3, 12
Interrupt Release Time, $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$	t_{IR}	–	1.60	–	1.10	–	0.85	µs	5, 14
Interrupt Response Time	t_{RS3}	–	1.0	–	1.0	–	1.0	µs	5, 13
Interrupt Input Pulse Time	PW_I	500	–	500	–	500	–	ns	13
$\overline{\text{RESET}}$ Low Time*	t_{RL}	1.0	–	0.66	–	0.5	–	µs	15

*The $\overline{\text{RESET}}$ line must be high a minimum of 1.0 µs before addressing the PIA.

FIGURE 2 – BUS TIMING TEST LOADS

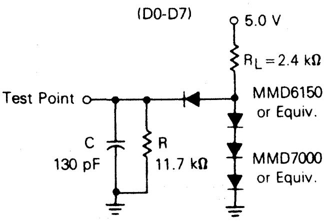
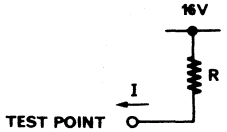


FIGURE 3 – TEST LOAD FOR

(PA0 - 7, PB0 - 7, CA2, CB2)



R IS SUCH THAT
I = 1mA WITH $V_{TEST\ POINT}$ = 0,4V

FIGURE 4 – NMOS EQUIVALENT TEST LOAD

($\overline{\text{IRQ}}$ Only)

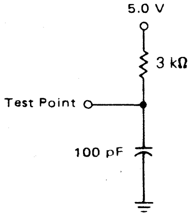


FIGURE 5 – PERIPHERAL DATA SETUP AND HOLD TIMES
(Read Mode)

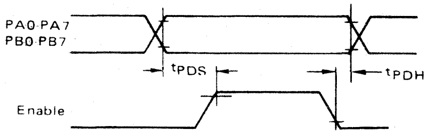
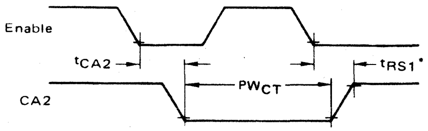


FIGURE 6 – CA2 DELAY TIME
(Read Mode; CRA-5 = CRA3 = 1, CRA-4 = 0)



*Assumes part was deselected during the previous E pulse.

FIGURE 7 – CA2 DELAY TIME
(Read Mode; CRA-5 = 1, CRA-3 = CRA-4 = 0)

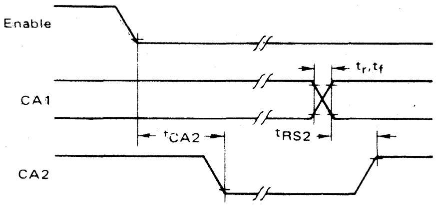


FIGURE 8 – PERIPHERAL CMOS DATA DELAY TIMES
(Write Mode; CRA-5 = CRA-3 = 1, CRA-4 = 0)

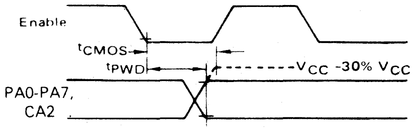
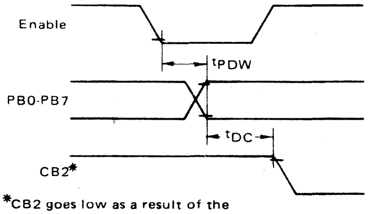
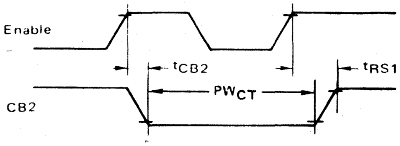


FIGURE 9 – PERIPHERAL DATA AND CB2 DELAY TIMES
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



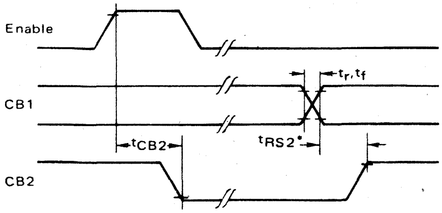
*CB2 goes low as a result of the positive transition of Enable.

FIGURE 10 – CB2 DELAY TIME
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



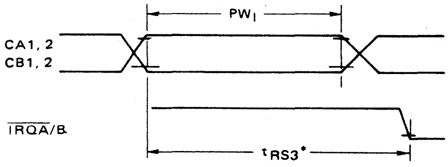
*Assumes part was deselected during the previous E pulse.

FIGURE 11 – CB2 DELAY TIME
(Write Mode; CRB-5 = 1, CRB-3 = CRB-4 = 0)



*Assumes part was deselected during any previous E pulse.

FIGURE 12 – INTERRUPT PULSE WIDTH AND $\overline{IRQ}$ RESPONSE



*Assumes Interrupt Enable Bits are set.

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 13 — $\overline{\text{IRQ}}$ RELEASE TIME

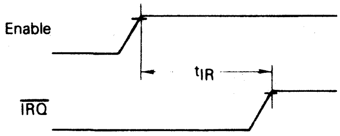
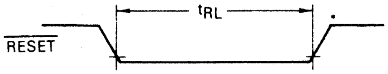


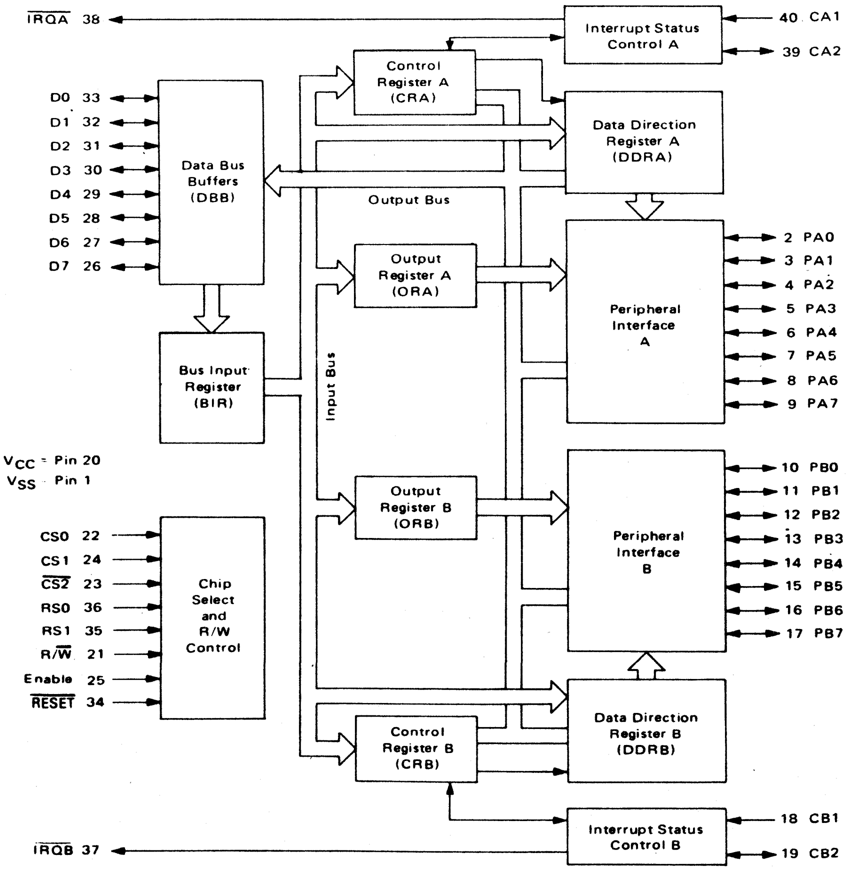
FIGURE 14 — $\overline{\text{RESET}}$ LOW TIME



*The $\overline{\text{RESET}}$ line must be a V_{IH} for a minimum of 1.0 µs before addressing the PIA.

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 15 — EXPANDED BLOCK DIAGRAM



PIA INTERFACE SIGNALS FOR MPU

The IIA interfaces to the M6800 bus with an 8-bit bidirectional data bus, three chip select lines, two register select lines, two interrupt request lines, a read/write line, an enable line and a reset line. To ensure proper operation with the MC6800, MC6802, or MC6808 microprocessors, VMA should be used as an active part of the address decoding.

Bidirectional Data (D0-D7) – The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the PIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a PIA read operation. The read/write line is in the read (high) state when **the IIA** is selected for a read operation.

Enable (E) – The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse.

Read/Write (R/$\overline{W}$) – This signal is generated by the MPU to control the direction of data transfers on the data bus. A low state on the **IIA read/write line enables the input** buffers and data is transferred from the MPU to **the IIA on** the E signal if the device has been selected. A high on the read/write line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

$\overline{\text{RESET}}$ – The active low $\overline{\text{RESET}}$ line is used to reset all register bits in **the IIA to** a logical zero (low). This line can be used as a power-on reset and as a master reset during system operation.

Chip Selects (CS0, CS1, and $\overline{\text{CS2}}$) – These three input signals are used to select the **IIA**. CS0 and CS1 must be high and $\overline{\text{CS2}}$ must be low for selection of the device. Data transfers are then performed under the control of the enable and read/write signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

Register Selects (RS0 and RS1) – The two register select lines are used to select the various registers inside the **IIA**. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

Interrupt Request ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) – The active low Interrupt Request lines ($\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each Interrupt Request line has two internal interrupt flag bits that can cause the Interrupt Request line to go low. Each flag bit is associated with a particular peripheral interrupt line. Also, four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each **IIA** for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the **IIA** is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA1, CA2, CB1, CB2). When these lines are used as interrupt inputs, at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

PIA PERIPHERAL INTERFACE LINES

The **IIA** provides two 8-bit bidirectional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) – Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode, the internal pullup resistor on these lines represents a maximum of 1.5 standard TTL loads.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low." Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read property if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

Section B Peripheral Data (PB0-PB7) – The peripheral data lines in the B Section of the IIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7, data on the peripheral data lines PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high" or above 0.8 V for a "low".

Interrupt Input (CA1 and CB1) – Peripheral input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) – The peripheral control line CA2 can be programmed to act as an interrupt input or as a an Open drain output. The action of this signal line is programmed with Control Register A.

Peripheral Control (CB2) – Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. This line is programmed by Control Register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all PIA registers. This will set PA0-PA7, PB0-PB7, CA2 and CB2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

There are six locations within the IIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS0 and RS1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Details of possible configurations of the Data Direction and Control Register are as follows:

TABLE 1 – INTERNAL ADDRESSING

RS1	RS0	Control Register Bit CRA-2	Control Register Bit CRB-2	Location Selected
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 16, the MC6822 has a pair of I/O ports whose characteristics differ greatly.

Notice the differences between a Port A and Port B read operation when in the output mode. When reading Port A, the actual pin is read, whereas the B side read comes from an output latch, ahead of the actual pin.

CONTROL REGISTERS (CRA and CRB)

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA1, CA2, CB1, and CB2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA1, CA2, CB1, or CB2. The format of the control words is shown in Figure 18.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 and CRB-2)

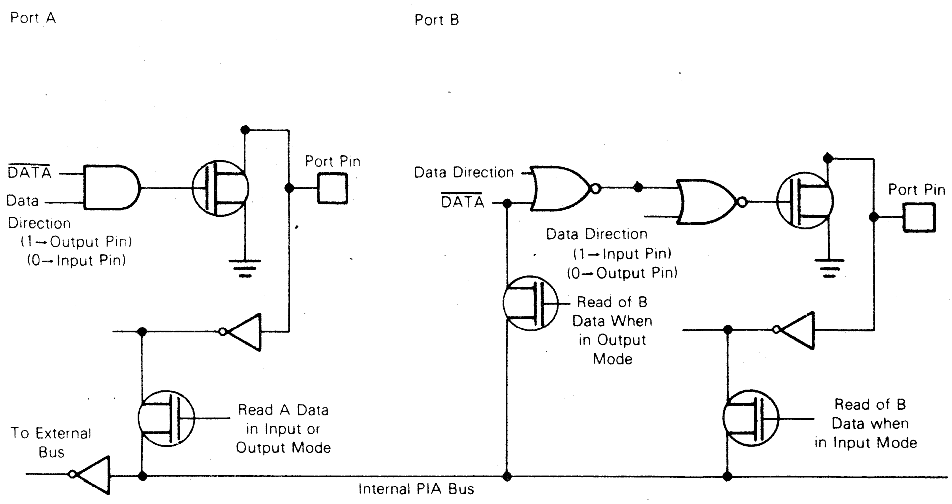
Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" causes the Data Direction Register to be addressed.

Interrupt Flags (CRA-6, CRA-7, CRB-6, and CRB-7) – The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA2 and CB2 Peripheral Control Lines (CRA-3, CRA-4, CRA-5, CRB-3, CRB-4, and CRB-5) – Bits 3, 4, and 5 of the two control registers are used to control the CA2 and CB2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA-5 (CRB-5) is low, CA2 (CB2) is an interrupt input line similar to CA1 (CB1). When CRA-5 (CRB-5) is high, CA2 (CB2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA2 and CB2 have slightly different loading characteristics.

Control of CA1 and CB1 Interrupt Input Lines (CRA-0, CRB-1, CRA-1, and CRB-1) – The two lowest-order bits of the control registers are used to control the interrupt input lines CA1 and CB1. Bits CRA-0 and CRB-0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA-1 and CRB-1 determine the active transition of the interrupt input signals CA1 and CB1.

FIGURE 16 – PORT A AND PORT B EQUIVALENT CIRCUITS



ORDERING INFORMATION

MC68A22CP

Motorola Integrated Circuit
M6800 Family
Blanks = 1.0 MHz
A = 1.5 MHz
B = 2.0 MHz
Device Designation
In M6800 Family
Temperature Range
Blank = 0°→ +70°C
C = −40°→ +85°C
Package
P = Plastic
S = Cerdip
L = Ceramic

BETTER PROGRAM

Better program processing is available on all types listed. Add suffix letters to part number.

Level 1 add "S" Level 2 add "D" Level 3 add "DS"

Level 1 "S" = 10 Temp Cycles – (−25 to 150°C),
Hi Temp testing at T_A max
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2

Speed	Device	Temperature Range
1.0 MHz	MC6822P,L,S	0 to 70°C
	MC6822CP,CL,CS	−40 to +85°C
1.5 MHz	MC68A22P,L,S MC68A22CP,CL,CS	0 to +70°C −40 to +85°C
2.0 MHz	MC68B22P,L,S	0 to +70°C

FIGURE 17 — CONTROL WORD FORMAT

Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 — (bit 7)

b1 = 0: IRQA(B)1 set by high-to-low transition on CA1 (CB1)

b1 = 1: IRQA(B)1 set by low-to-high transition on CA1 (CB1).

CA1 (CB1) Interrupt Request Enable/Disable

b0 = 0: Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition.[1]

b0 = 1: Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition.

1. IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled.

IRQA(B) 1 Interrupt Flag (bit 7)

Goes high on active transition of CA1 (CB1); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

Control Register

b7	b6	b5	b4	b3	b2	b1	b0
IRQA(B)1 Flag	IRQA(B)2 Flag	CA2 (CB2) Control			DDR Access	CA1 (CB1) Control	

IRQA(B)2 Interrupt Flag (bit 6)

When CA2 (CB2) is an input, IRQA(B) goes high on active transition CA2 (CB2); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA2 (CB2) Established as Output (b5 = 1): IRQA(B) 2 = 0, not affected by CA2 (CB2) transitions.

Determines Whether Data Direction Register Or Output Register is Addressed

b2 = 0: Data Direction Register selected.

b2 = 1: Output Register selected.

CA2 (CB2) Established as Output by b5 = 1

(Note that operation of CA2 and CB2 output functions are not identical)

b5 b4 b3

1 0

→ CA2

b3 = 0: **Read Strobe with CA1 Restore** CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next active CA1 transition, as specified by bit 1.

b3 = 1: **Read Strobe with E Restore** CA2 goes low on first high-to-low E transition following an MPU read of Output Register A; returned high by next high-to-low E transition during a deselect.

→ CB2

b3 = 0: **Write Strobe with CB1 Restore** CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next active CB1 transition as specified by bit 1. CRB-b7 must first be cleared by a read of data.

b3 = 1: **Write Strobe with E Restore** CB2 goes low on first low-to-high E transition following an MPU write into Output Register B; returned high by the next low-to-high E transition following an E pulse which occurred while the part was deselected.

b5 b4 b3

1 1 → **Set/Reset CA2 (CB2)**

CA2 (CB2) goes low as MPU writes b3 = 0 into Control Register.

CA2 (CB2) goes high as MPU writes b3 = 1 into Control Register.

CA2 (CB2) Established as Input by b5 = 0

b5 b4 b3

0

→ **CA2 (CB2) Interrupt Request Enable/Disable**

b3 = 0: Disables IRQA(A) MPU Interrupt by CA2 (CB2) active transition.*

b3 = 1: Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.

*IRQA(B) will occur on next (MPU generatted) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled.

→ **Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 — (Bit b6)**

b4 = 0: IRQA(B)2 set by high-to-low transition on CA2 (CB2).

b4 = 1: IRQA(B)2 set by low-to-high transition on CA2 (CB2).

MOTOROLA

MC6828
MC8507

Note: The dual numbering system emphasis that this device is a bipolar LSI service and directly compatible with the M6800X Microprocessor Family. The Priority Interrupt Controller may be ordered by using either part number.

PRIORITY INTERRUPT CONTROLLER

The MC6828/8507 Priority Interrupt Controller (PIC) is used to add prioritized responses to inputs to microprocessor systems. The performance has been optimized for the M6800X system, but will serve to eliminate input polling routines from any processor system.

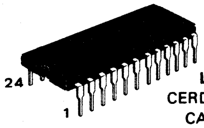
The MC6828/8507 (PIC) modifies the vector ROM addresses that the microprocessor uses to jump to an interrupt routine. The MC6828 provides the user with an additional eight latched interrupt inputs, and it can be cascaded to provide more interrupts.

An interrupt mask prevents any latched interrupt input of lower priority than the mask level from generating an $\overline{\text{IRQ}}$ output.

The (PIC) allows for any added decode time by generating a $\overline{\text{Stretch}}$ signal which can be used to slow the processor clock while fetching interrupt routine starting addresses. The $\overline{\text{Stretch}}$ signal allows the interrupt structure to be designed without concern for faster operation due to improvements in processor speeds.

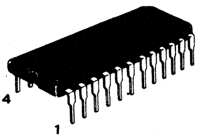
MEGALOGIC

PRIORITY INTERRUPT CONTROLLER

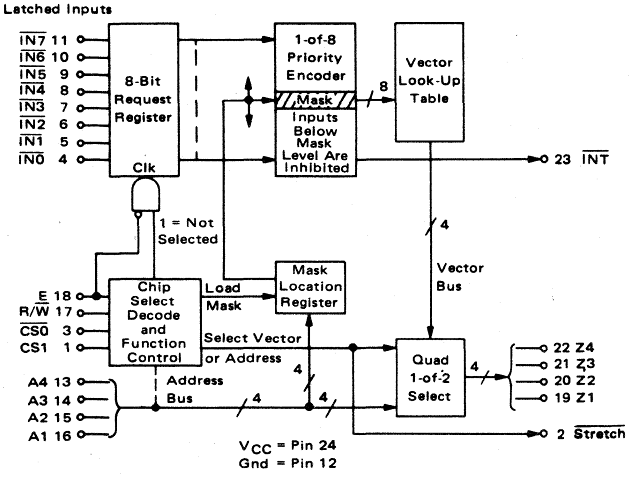


L SUFFIX
CERDIP PACKAGE
CASE 623-03

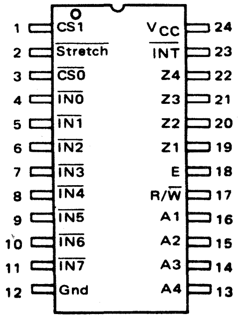
P SUFFIX
PLASTIC PACKAGE
CASE 649-03



BLOCK DIAGRAM



PIN ASSIGNMENT



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.5 to + 7.0	Vdc
Input Voltage	V_{in}	-1.0 to +5.5	Vdc
Output Voltage	V_{OH}	-0.4 to +7.0	Vdc

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance Cerdip Plastic	$R_{\theta JA}$	65 120	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

$$K = P_D [T_A + 273°C + (P_D \cdot \theta_{JA})] \qquad (3)$$

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} = 5.0 Vdc ±5%, T_A = 0 to 75°C unless otherwise noted)

Characteristic		Symbol	Min	Max	Unit
Input Forward Current		I_{IL}			μAdc
(V_{IL} = 0, V_{CC} = 5.25 Vdc)	CS1, E		—	-75	
	$\overline{CS0}$, R/$\overline{W}$		—	-150	
	A1 thru A4		—	-225	
	$\overline{IN0}$ thru $\overline{IN7}$		—	-1300	
Input Leakage Current		I_{IH}			μAdc
(V_{IH} = 2.4 Vdc, V_{CC} = 5.25 Vdc)	CS1		—	120	
	$\overline{CS0}$		—	240	
	A1 thru A4		—	360	
	$\overline{1N0}$ thru $\overline{IN7}$		—	-560	
DC Logic "0" Output Voltage		V_{OL}			Vdc
(I_{OL} = 1.6 mAdc, V_{ILT} = 0.8 Vdc, V_{IHT} = 2.0 Vdc, V_{CC} = 4.75 Vdc)	Z1 thru Z4, $\overline{Stretch}$		—	0.5	
(I_{OL} = 3.2 mAdc, V_{CC} = 4.75 Vdc)	$\overline{IRQ}$ — Open Collector		—	0.5	
DC Logic "1" Output Voltage (I_{OH} = -0.3 mAdc, V_{ILT} = 0.8 Vdc, V_{IHT} = 2.0 Vdc, V_{CC} = 4.75 Vdc)	Z1 thru Z4, $\overline{Stretch}$	V_{OH}	2.4	—	Vdc
Output Leakage Current (V_{CC} = V_{CEX} = 5.25 Vdc)	$\overline{INT}$	I_{CEX}	—	200	μAdc
Power Supply Drain Current (V_{CC} = 5.0 Vdc, All Inputs Open)		I_{CC}	—	125	mAdc

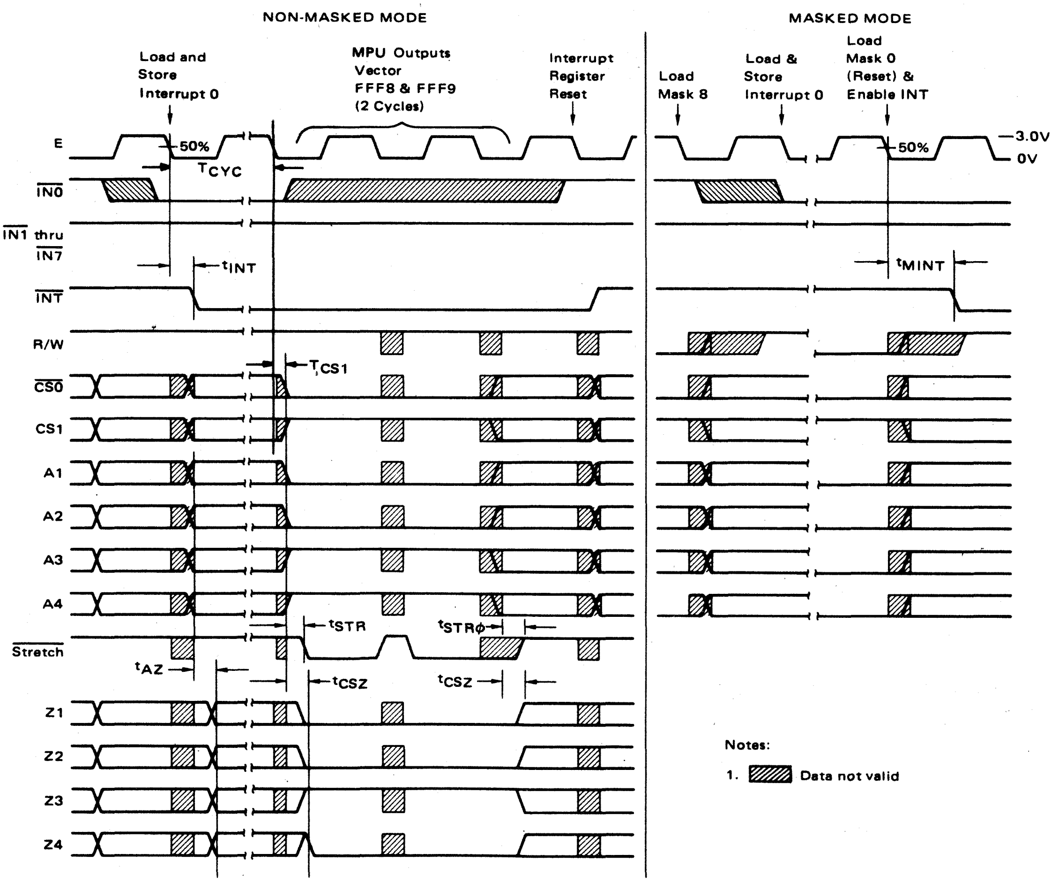
SWITCHING TIMES (V_{CC} = 5.0 Vdc, T_A = 25°C)

Characteristic	Symbol	Min	Max	Unit
A_i to Z_i Delay Time (Not Selected)	t_{AZ}	—	75	ns
A_i to Z_i Delay Time (Selected)	t_{AZ}	—	60	ns
Select* to Z_i Delay Time ($\overline{A1}$ · $\overline{A2}$ · A3 · A4 · CS0 · CS1 to Z_i)	t_{CSZ}	—	125	ns
Enable Pulse Width	T_{CYC}	100	—	ns
Enable Low to CS1	T_{CS1}	125	—	ns
Deselect to $\overline{\text{Stretch}}$ High	$t_{STR\phi}$	—	125	ns
Select* to $\overline{\text{Stretch}}$ Delay Time ($\overline{A1}$ · $\overline{A2}$ · A3 · A4 · $\overline{CS0}$ · CS1 to $\overline{\text{Stretch}}$)	t_{STR}	—	140	ns
Enable to $\overline{INT}$ Delay Time, Non-Masked Mode	t_{INT}	—	240	ns
Enable to $\overline{INT}$ Delay Time, Masked Mode	$t_{MINT(IN1)}$ $t_{MINT(INJ)}$	— —	360** 200**	ns

*Select = ($\overline{A1}$ · $\overline{A2}$ · A3 · A4 · $\overline{CS0}$ · CS1 · R/$\overline{W}$) which corresponds to FFF8 or FFF9 interrupt response in the M6800 system.

**Value depends on mask level and stored priority input. Maximum value occurs with mask level 7 and stored interrupt IN0. Minimum value occurs with mask level J and stored interrupt IN(J).

FIGURE 1 — FUNCTIONAL WAVEFORMS



OPERATING CHARACTERISTICS

The primary purpose of the Priority Interrupt Controller (PIC) is to generate a modified address to ROM in response to prioritized inputs. With the PIC, each interrupting device is assigned a unique ROM location which contains the starting address of the appropriate service routine. After the MPU detects and responds to an interrupt, the PIC directs the MPU to the proper memory location.

The basic functions of the PIC are shown in the block diagram. The 8-bit request register is an edge clocked D-type register with internal 6 kΩ pullup resistors on the interrupt inputs (IN0 thru IN7). Note the inputs are active low. The interrupt register is loaded on the falling edge of the enable when the PIC is not selected.

The 1-of-8 priority encoder enables a vector corresponding to the stored interrupt with the highest priority and places it on the vector input port of a data selector. In addition an interrupt request signal $\overline{\text{INT}}$ is generated to signal the MPU that an interrupt has been detected. The mask location register overrides and inhibits all interrupts with priority below the mask level. The mask can be thought of as a movable partition allowing responses to inputs equal to or greater than the mask value. For example if the stored mask level was 4, inputs $\overline{\text{IN0}}$, $\overline{\text{IN1}}$, $\overline{\text{IN2}}$, and $\overline{\text{IN3}}$ would not generate an interrupt to the MPU system. The input request register is not affected by the mask, and if the mask is cleared (by loading it with zeros), any previously stored inputs will generate an IRQ signal.

FIGURE 2 — MC6828 TRUTH TABLE FOR M6800 MICROPROCESSOR SYSTEMS

Active Input		Output When Selected				Equivalent to Bits 1-4 of B0, B1 . . . , B15 Hex Address	Address ROM Bytes Contain Address of:
		Z4	Z3	Z2	Z1		
Highest	$\overline{\text{IN7}}$	1	0	1	1	F F F 6 or 7	Priority 7 Routine
	$\overline{\text{IN6}}$	1	0	1	0	F F F 4 or 5	Priority 6 Routine
	$\overline{\text{IN5}}$	1	0	0	1	F F F 2 or 3	Priority 5 Routine
	$\overline{\text{IN4}}$	1	0	0	0	F F F 0 or 1	Priority 4 Routine
	$\overline{\text{IN3}}$	0	1	1	1	F F E E or F	Priority 3 Routine
	$\overline{\text{IN2}}$	0	1	1	0	F F E C or D	Priority 2 Routine
	$\overline{\text{IN1}}$	0	1	0	1	F F E A or B	Priority 1 Routine
Lowest	$\overline{\text{IN0}}$	0	1	0	0	F F E 8 or 9	Priority 0 Routine
	None	1	1	0	0	F F F 8 or 9	Default Routine*

*Default routine is the response to interrupt requests not generated by a prioritized input. The default routine may contain polling routines or may be an address in a loop for an interrupt driven system.

FIGURE 3 — MC6828 TRUTH TABLE FOR M6809 MICROPROCESSOR SYSTEMS

Active Input		Output When Selected				Equivalent Hex Address	Address ROM Bytes Contain Address of:
		Z4	Z3	Z2	Z1		
Highest	IN7	1	0	1	1	F F D 6 — F F D 7	Priority 7 Routine
	IN6	1	0	1	0	F F D 4 — F F D 5	Priority 6 Routine
	IN5	1	0	0	1	F F D 2 — F F D 3	Priority 5 Routine
	IN4	1	0	0	0	F F D 0 — F F D 1	Priority 4 Routine
	IN3	0	1	1	1	F F C E — F F C F	Priority 3 Routine
	IN2	0	1	1	0	F F C C — F F C D	Priority 2 Routine
	IN1	0	1	0	1	F F C A — F F C B	Priority 1 Routine
	IN0	0	1	0	0	F F C 8 — F F C 9	Priority 0 Routine
Lowest	None	1	1	0	0	F F F 8 — F F F 9	Default Routine ($\overline{\text{IRQ}}$)

Chip Select and Stretch

The chip select and decode circuitry controls all internal functions of the PIC. The selected mode is defined as the logical AND function $\overline{A1} \cdot \overline{A2} \cdot A3 \cdot A4 \cdot \overline{CS0} \cdot CS1 \cdot R/\overline{W}$. When the device is not in the selected mode the request register clock is enabled and the address inputs A_i passes directly through the data selector to the Z_i outputs. When the MPU responds to the interrupt request and the PIC decodes the select address, the request register is inhibited and the data selector places the vector on the Z outputs. The address delay added to the MPU system is shown in Figure 4. This delay may be critical in some systems. A stretch signal, which indicates the selected mode, is provided for use with special MPU clock drivers to stretch the clock cycle when accessing slow ROM. This stretch signal is applicable only to those microprocessors which incorporate external clock generators, i.e., 6800, 6809E. The user cannot directly connect stretch to MRDY on the MC6809 or MC6802, as stretching the clock circuit with a signal which is derived from the clock will latch up the MPU. An alternative to this problem is to use a one-shot which would provide the required amount of access time. Figure 8 illustrates a typical such circuit. The $\overline{CS0}$ output has one less gating level than the remainder of the select decode logic. This allows an external NAND gate to be used for the full address decode without any increase in delay times.

Programming the PIC

Changing the priority level, or mask, in the PIC is done by writing to the device. Unlike normal programming of a peripheral where a specific data pattern is written into a selected register, the PIC is programmed by accessing a location determined by A1 through A4 while $R/\overline{W}$ is low.

The decode logic also controls the loading of the mask location register. This register will be loaded on the falling edge of the enable pulse when enabled by the logical AND function $\overline{CS0} \cdot CS1 \cdot \overline{R/W}$ (Note 1). This means that in the load mask mode the data on the data bus is a don't care. However, in this mode the ROM will also be accessed and both the ROM and MPU will be driving the data bus. Therefore the read/write line should be used as an active high chip select or enable signal for ROM decoding.

Figure 5 shows the typical operation flow diagram for the PIC in an M6800 system. The functional timing for this flow is as shown in the first part of the waveforms in Figure 1. The second half of Figure 1 shows the operation of the mask. Interrupts will be stored even if they are masked. When the mask is released the $\overline{INT}$ signal will then be generated.

The influence of the mask register on the priority encoder is shown in the truth table of Figure 6. The actual use of the mask register will vary with the system needs and the imaginative software programmer.

Special Cases

As originally conceived, the PIC was only meant to be used with the MC6800 MPU. With the advent of higher performance/function microprocessors such as the MC6809/MC6802/MC6808, prioritized interrupts are still required. The interrupt vector map for the M6800 is located from FFF8 to FFFF. With the MC6809, this vector map extends downward to FFF0. As can be seen in Figure 2, the normal configuration of the PIC places priority interrupt vectors from FFE8 to FFF7 which conflict with the existing MC6809 interrupt vectors. Figure 1 shows appropriate circuitry required to interface with the MC6809. Figure 3 gives the "modified priority vectors" associated with this hardware modification.

Note 1. Since during normal operation of the MPU the address lines and the $R/\overline{W}$ line can be in an indeterminate state, VMA should be logically ANDed with one of the chip select inputs of the PIC to prevent erroneous writes into the mask register (non-6800 systems).

FIGURE 4 — HIGH ROM ADDRESS DELAY ADDED TO M6800X SYSTEMS

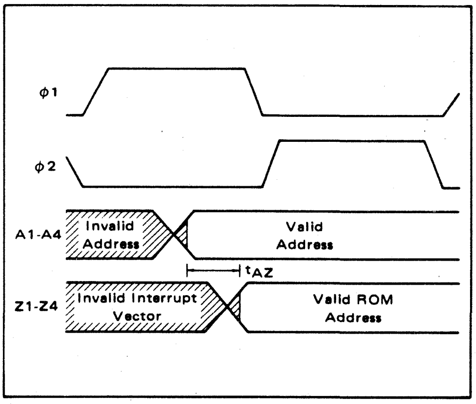


FIGURE 5 — BASIC FUNCTIONAL FLOW CHART

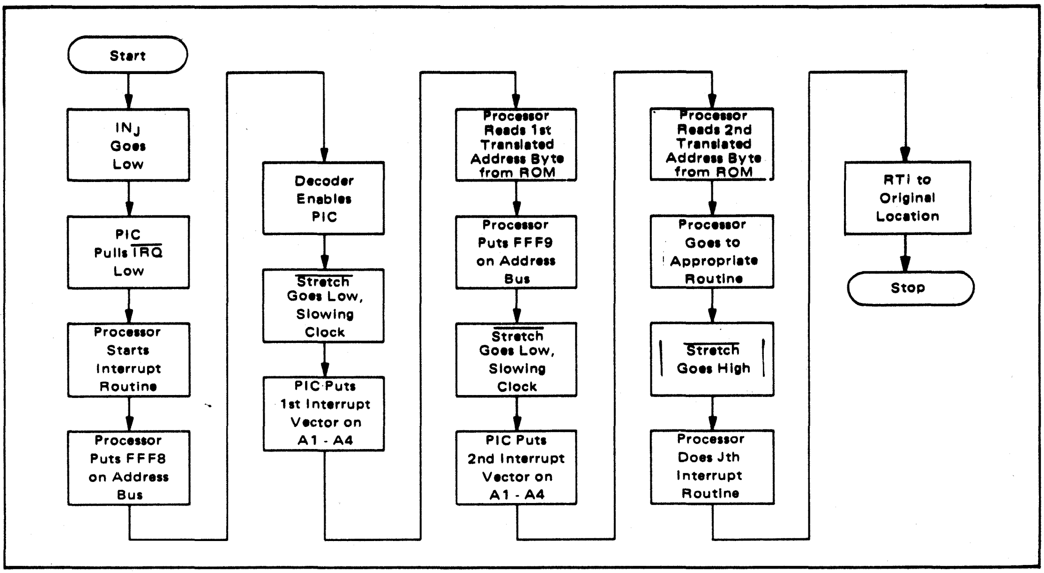
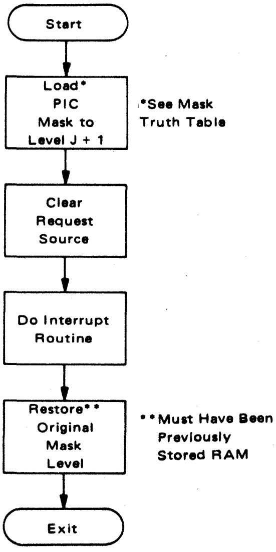


FIGURE 6 — MASK OPERATION

a – MASK FLOW CHART



b – MASK TRUTH TABLE

Mask Register Contents				Response to Priority Inputs 1 = Response to Input, 0 = No Response							
M4	M3	M2	M1	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0	0
0	1	0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	0
0	0	0	0	1	1	1	1	1	1	1	1

FIGURE 7 — TYPICAL SYSTEM CONFIGURATION

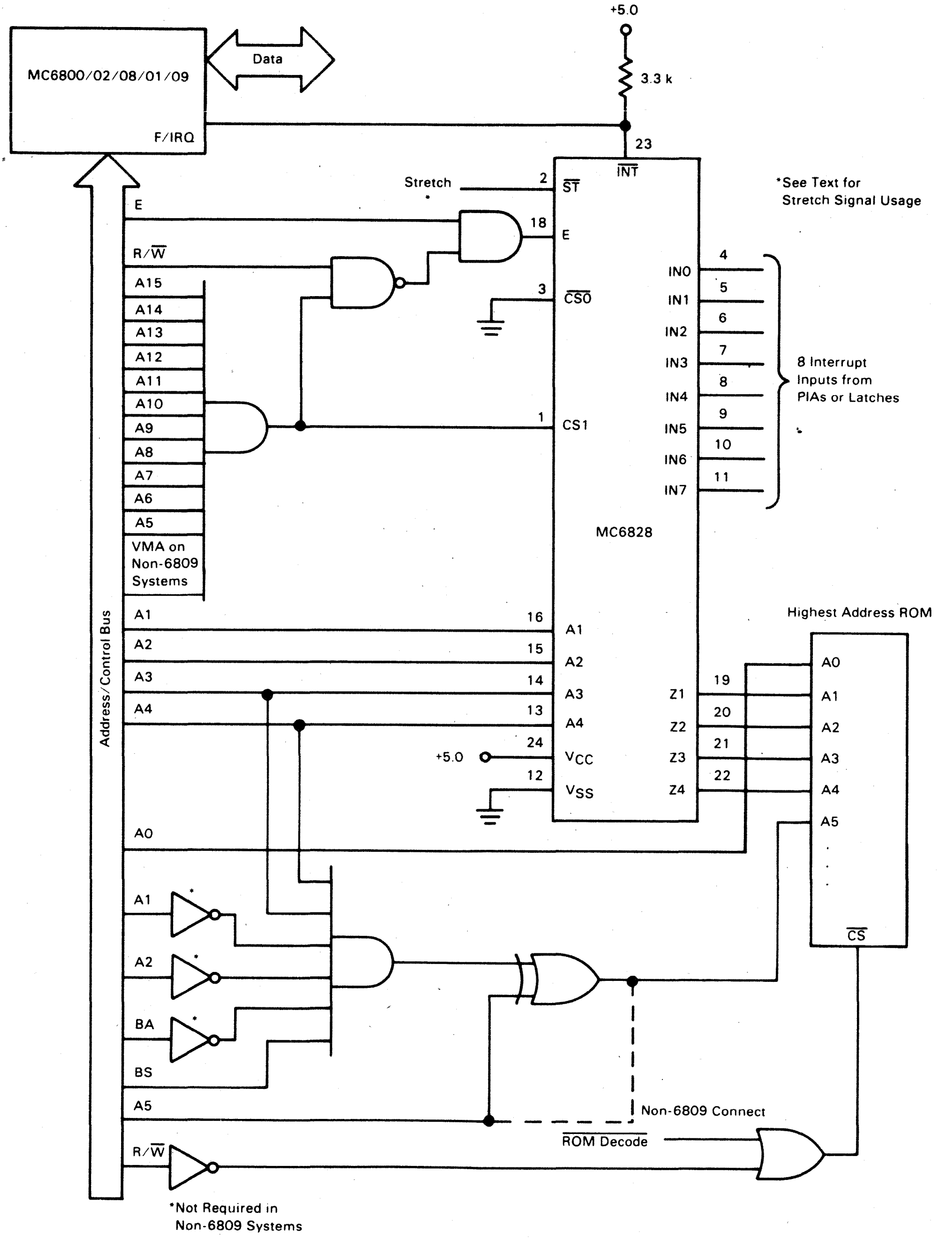
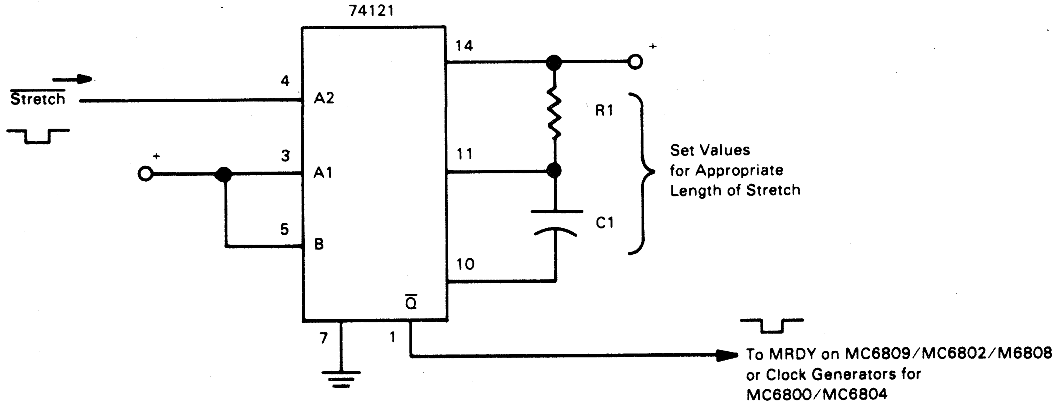


FIGURE 8 — ACCESS TIME EXTENSION USING $\overline{\text{STRETCH}}$ AND MRDY



Ordering Information

MC 68 28 P

Motorola Integrated Circuit — MC
M6800 Family — 68
Device Designations — 28
Package Designation — P
P = Plastic
L = Ceramic

MOTOROLA

MC6829
(1.0 MHz)
MC68A29
(1.5 MHz)
MC68B29
(2.0 MHz)

Advance Information

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

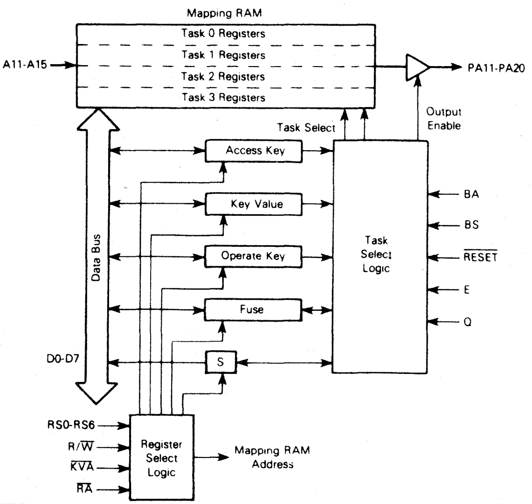
MEMORY MANAGEMENT UNIT
(MMU)

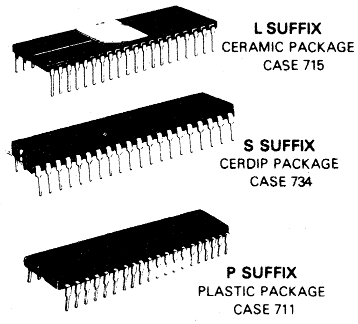
MEMORY MANAGEMENT UNIT

The principle function of the MC6829 Memory Management Unit (MMU) is to expand the address space of the MC6809 from 64K bytes to a maximum of 2 Megabytes. Each MMU is capable of handling four different concurrent tasks including DMA. The MMU can also protect the address space of one task from modification by another task. Memory address space expansion is accomplished by applying the upper five address lines of the processor (A11-A15) along with the contents of a 5-bit task register to an internal high-speed mapping RAM. The MMU output consists of ten physical address lines (PA11-PA20) which, when combined with the eleven lower address lines of the processor (A0-A10), forms a physical address space of 2 Megabytes. Each task is assigned memory in increments of 2K bytes up to a total of 64K bytes. In this manner, the address spaces of different tasks can be kept separate from one another. The resulting simplification of the address space programming model will increase the software reliability of a complex multi-process system.

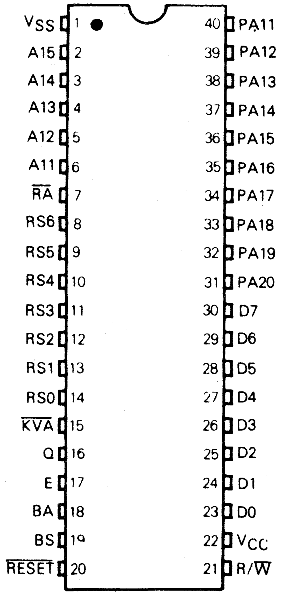
- Expands Memory Address Space from 64K to 2 Megabytes
- Each MMU is Capable of Handling Four Separate Tasks
- Up to Eight MMUs can be Used in a System
- Provides Task Isolation and Write Protection
- Provides Efficient Memory Allocation; 1024 Pages of 2K Bytes Each
- Designed for Efficient Use with DMA
- Fast, Automatic On-Chip Task Switching
- Allows Inter-Process Communication Through Shared Resources
- Simplifies Programming Model of Address Space
- Increases System Software Reliability
- MC6809/MC6800 Bus Compatible
- Single 5-Volt Power Supply

BLOCK DIAGRAM





PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	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 MC6829, MC68A29, MC68B29 MC6829C, MC68A29C, MC68B29C	T_A	T_L to T_H 0 to 70 − 40 to + 85	°C
Storage Temperature Range	T_{stg}	− 55 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. 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

	Symbol	Value	Rating
Thermal Resistance Plastic Cerdip Ceramic	θ_{JA}	 100 60 50	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

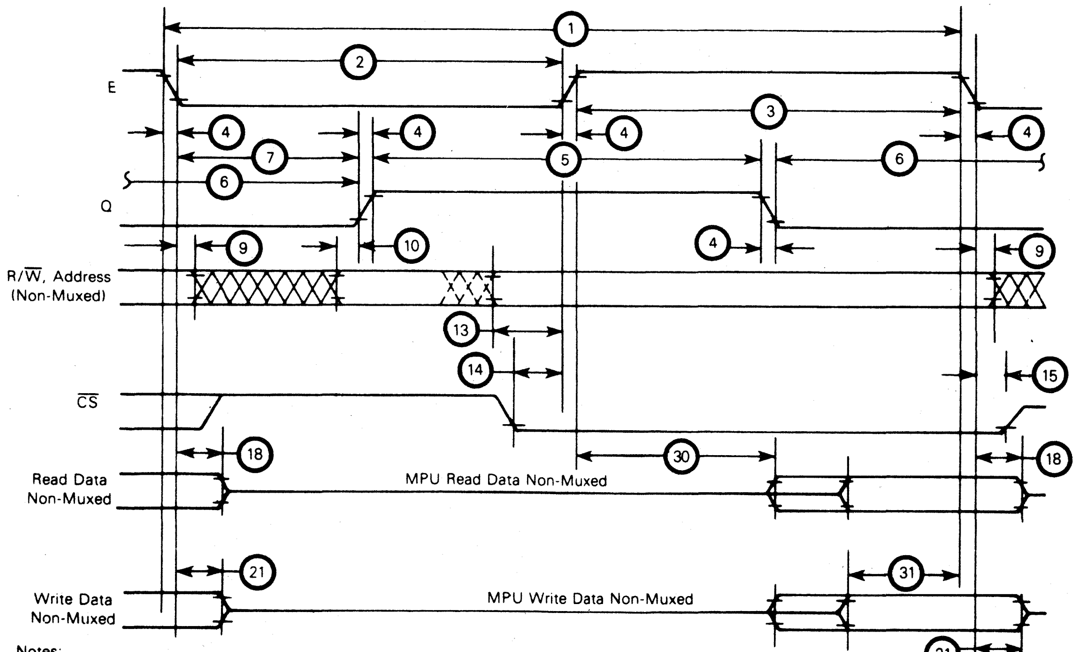
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage	All Inputs	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	V_{CC} = Max	I_{in}	–	1.0	2.5	µA
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{IZ}	–	2.0	10	µA
Output High Voltage (I_{load} = − 145 µA) V_{CC} = min	 D0-D7 PA11-PA20	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4	 – –	 – –	V
Output Low Voltage (I_{load} = 2.0 mA) V_{CC} = max	 D0-D7 PA11-PA20	V_{OL}	 – –	 – –	 V_{SS} + 0.5 V_{SS} + 0.5	V
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	–	800	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.5 MHz)	All Inputs	C_{in}	–	10.0	12.0	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.5 MHz)	All Outputs	C_{out}	–	–	12.0	pF

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6829		MC68A29		MC68B29		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Witdth, E Low	PW_{EL}	430	9500	280	9500	210	9700	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9700	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
5	Pulse Width, Q High	PW_{QH}	430	5000	280	5000	210	5000	ns
6	Pulse Width, Q Low	PW_{QL}	450	9500	280	9500	220	9500	ns
7	E to Q Rise Delay Time*	t_{AVQ}	–	250	–	165	–	125	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns
See Figures 2 and 3	Three-State Address Delay	t_{TAD}	–	90	–	80	–	60	ns
See Figure 2	Mapped Address Delay	t_{MAD}	–	200	–	145	–	110	ns

*At specified cycle time.

FIGURE 1 — BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

FIGURE 2 — MAP SWITCHING, ADDRESS MAPPING

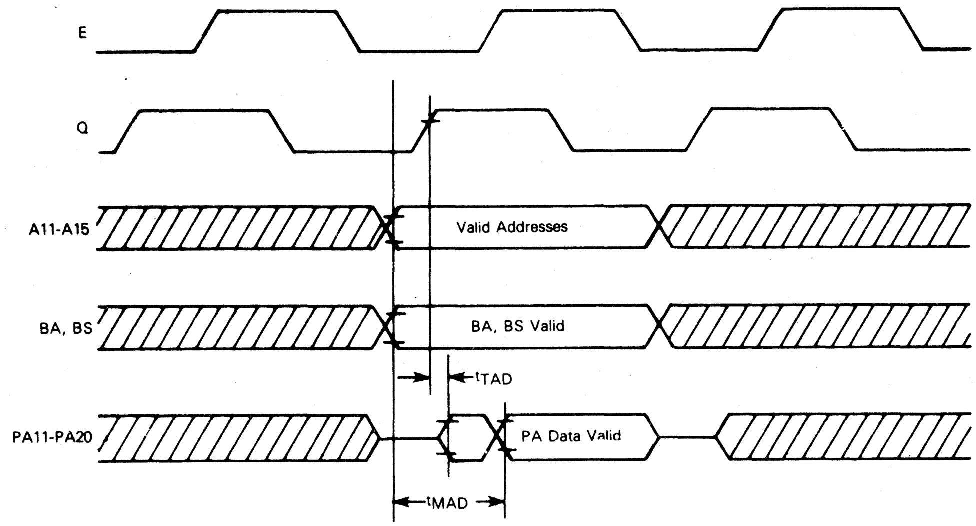
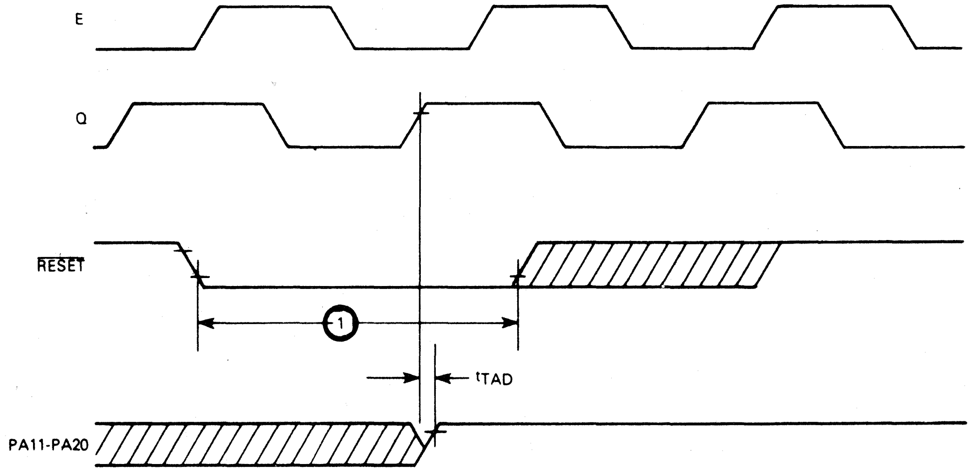


FIGURE 3 — $\overline{\text{RESET}}$ TIMING



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.

PIN DESCRIPTION

The following section describes each pin of the MMU in detail.

V_{CC}, V_{SS} – Supplies power to the MC6829. V_{CC} is +5 volts and V_{SS} is ground.

E – Input E clock (from MC6809).

Q – Input Q clock (from MC6809).

$R/\overline{W}$ – Read/Write Line Input; 1 = Read, 0 = Write.

D0-D7 – Bi-directional Data Bus. The data bus is used when the MMU registers are to be read or written.

PA11-PA20 – Physical Address Lines (Output from MMU). The physical address lines are generated by the MMU for every bus cycle. When multiple MMUs are present in a system, only one MMU will output a physical address. Each physical address line will drive one Schottky TTL load or four LS TTL loads and a maximum of 90 pF.

RS0-RS6 – Register Select Lines (Access to MMU Registers). When accessing the MMU registers, the register select lines determine which byte of information is being referenced within the MMU. Valid addresses are detailed in the Register Select Truth Table.

BA, BS – Bus Available and Bus State (Inputs). These inputs are directly connected from the BA, BS lines of the MC6809. They provide the MMU with information about the class of bus operation for each cycle. Note that when coming out of a DMA cycle, the MC6809 BA, BS pins change back from DMA acknowledge (BA = 1, BS = 1) to running (BA = 0, BS = 0) one cycle before the end of the DMA.

$\overline{RA}$ – Register Access (Chip Select for MMU Registers). This active low input determines the location of the MMU registers. Since the MMU registers are only accessible from the last page of task #0 ($F800-$FFFF), this signal can be derived from address lines A10-A7 of the processor. When RA is asserted low, the MMU registers are selected if the current task number is zero and A15-A11 are all 1's.

$\overline{KVA}$ – Key Value Access select line (Input). This active low input enables access to the 3-bit Key Value register on the MMU. Reading the Key Value Register is allowed only when the current task is zero, address lines A11-A15 are all ones, $\overline{RA}$ = 0 (asserted), RS6-RS0 are within the range $40-$47 and $\overline{KVA}$ = 0 (also asserted). Writing the Key Value Register has the additional requirement of having the S-bit set.

$\overline{RESET}$ – $\overline{RESET}$ (Input). A low level on this input causes the MMU to initialize its registers to a known state. An internal flag is also set which forces $3FF onto the physical address lines until the Key Value Register is written. $\overline{RESET}$ must be low for at least one cycle.

MMU OPERATION

For every processor cycle, the MMU supplies a mapped address based on the processor address and the current task number (refer to Figure 4). The current task number is kept in an on-chip register called the OPERATE KEY. Changing the value of the operate key causes a new map to be selected.[*] The MMU also contains automatic task switching logic to cause pre-defined task numbers to override the task number in the operate key for certain events (Interrupts, Direct Memory Access, Reset).

The MMU registers always appear as a block of 64 byes located on the last page of task #0 (refer to Figure 5). When the registers are accessed, the MMU outputs a physical address of $3FF (PA11-PA20 all high). This is necessary since the mapping RAM of the MMU cannot map an address and be modified at the same time.

The exact location of the MMU registers within the last page of physical memory is determined by the REGISTER ACCESS ($\overline{RA}$) signal which is similar to a chip select line. The $\overline{RA}$ signal will normally be derived from processor address lines A7-A10 using a simple 4-input gate. For example, a 4-input OR gate would place the MMU registers at $F800 to $F87F. In systems using DMA, the $\overline{RA}$ input must include the externally derived DMA/VMA signal to prevent dead bus cycles from affecting the MMU. Refer to Programming Considerations.

Inputs RS0-RS6 to the MMU are the register select lines. These lines are normally connected to the low order address lines A0-A6 from the processor. The MMU registers are only accessible if:

1. the current task number is zero;
2. processor address lines A11-A15 are all 1's;
3. the Register Access line ($\overline{RA}$) is asserted low;
4. Register Select lines (RS0-RS6) contain a defined register address; and
5. the System Bit (S-bit) is set (for a write operation only).

As a result of the above restrictions on accessing the MMU registers, the portion of the software that sets up and maintains the memory maps for all tasks must run as task zero.

The first 64 bytes of the MMU's register area comprise a "window" through which any one of the 4 maps may be viewed or changed. The task number to be viewed through this "window" is written into a read/write register called the ACCESS KEY. Thus, to examine or change the map for any task, the processor must first write the task number into the Access Key. Once set, the Access Key will retain its value until explicitly changed.

---

[*] Refer to Register Select Truth Table for exact procedure to change this register.

FIGURE 4 — LOGIC-TO-PHYSICAL ADDRESS TRANSLATION DIAGRAM

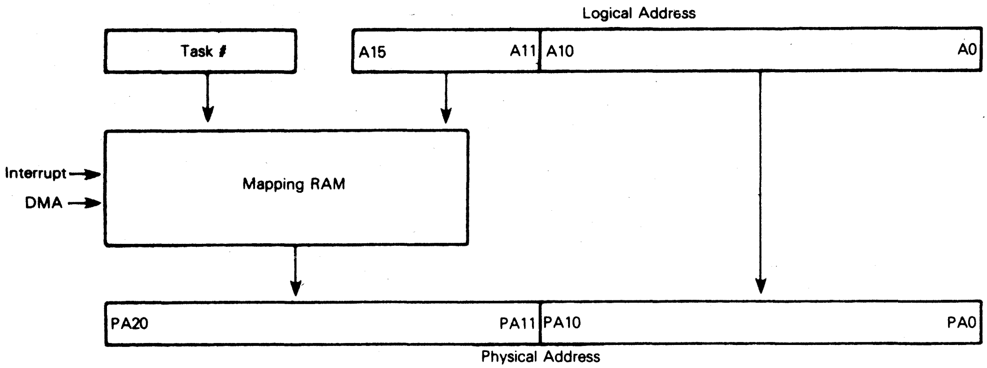


FIGURE 5 — MMU REGISTER MODEL

	Register	D7	D6	D5	D4	D3	D2	D1	D0	Logical Address
Access Key "Window"	00							PA20	PA19	$0000-07FF
	01	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	
	02							PA20	PA19	$0800-$0FFF
	03	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	
	04							PA20	PA19	$1000-$17FF
	05	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	
					o o o o o					
	3E							PA20	PA19	$F800-$FFFF
	3F	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	
	40						KV MMU0			Only one Key Value Register for each MMU, but all Key Value Registers fall in this range
	41						KV MMU1			
	42						KV MMU2			
	43						KV MMU3			
	44						KV MMU4			
	45						KV MMU5			
	46						KV MMU6			
	47						KV MMU7			
	48								S	System/User flag bit
	49						Fuse			Map Switch Fuse
	4A				Access Key					Task Currently Accessed Through Register #$0-$3F
	4B				Operate Key					Current Task
	4C				o o o o					Undefined
	7F									

Notes:

1. The contents of bytes $4C through $7F are undefined and do not respond to any reads or writes.
2. The Access, Operate and Key Value Registers are cleared on reset. The S-bit is set.
3. Unused bits of defined registers always read zeros.
4. Locations $40-$47 are accessible only when $\overline{KVA} = 0$.
5. In multiple MMU configurations, the MMU whose Key Value Register matches the upper three bits of the access key will respond to a processor read of locations $48-$4B. Processor writes to these registers will cause the data to be written to all MMUs simultaneously.

Pages in physical memory require 10 bits to define their location (refer to Figure 5). These 10 bits are arranged as a pair of bytes in the MMU in order to allow the use of double byte instructions (e.g., LDD) in manipulating the MMU registers. These first 64 bytes of the register area are then accessed as 32 pairs of bytes with each pair describing the logical-to-physical mapping for one 2K page. Registers 0 and 1 contain the page number for Igoical addresses \$0000-\$07FF, register 2 and 3 control logical addresses \$0800-\$0FFF, etc.

Each MMU has a 3-bit register called the KEY VALUE REGISTER. This register determines the range of task numbers an MMU controls. The top three bits of the Operate Key must match the Key Value Register for that task to be active. Similarly, the Key Value Register must match the top three bits of the Access Key to change or view registers #0 through #\$3F. Each MMU must receive a unique key value when the system is initialized to guarantee that no two MMUs control the same range of tasks. To be able to write to each MMU's Key Value Register separately, an external decoder must be provided. This decode function can be derived from address lines A0, A1 and A2 using a 3-to-8 line decoder. Writing to locations \$40-\$47 will cause the Key Value of the MMU to be updated only if the $\overline{KVA}$ input is low. In systems using a single MMU, the $\overline{KVA}$ input may be wired low.

BUILDING AN MMU SYSTEM

Up to 8 chips may be connected in parallel to create a maximum of 32 tasks. All MMU pins except one ($\overline{KVA}$) may be wired in parallel. Each MMU chip contains 1280 bits of fast on-chip lookup RAM. This RAM is accessible 10 bits at a time for mapping purposes, and as 2 and 8 bits at a time when the Operating System OS is changing the contents of the RAM. In addition to the lookup RAM, each MMU contains a separate copy of the Access Key, Operate Key, Fuse Register, Key Value Register, and S-bit. A CPU write to the Access, Operate, or Fuse Register causes all registers on all MMUs to be updated. In contrast, the lookup RAM for each chip is updated only when the top three bits of the Access Key match the Key Value Register for that chip. During mapping operations, each MMU compares the value in its Operate Key (top three bits) with its Key Value Register and responds only if a match is found. Similarly, when the processor reads the RAM, each MMU compares its Key value with the Access Key (Figure 6).

REGISTER SELECT TRUTH TABLE

Table 1 shows how the MMU registers are accessed by the processor. It is assumed that the current task is zero and that the processor address lines A11-A15 are all ones. If the S-bit is not set, the registers are still readable, but cannot be modified.

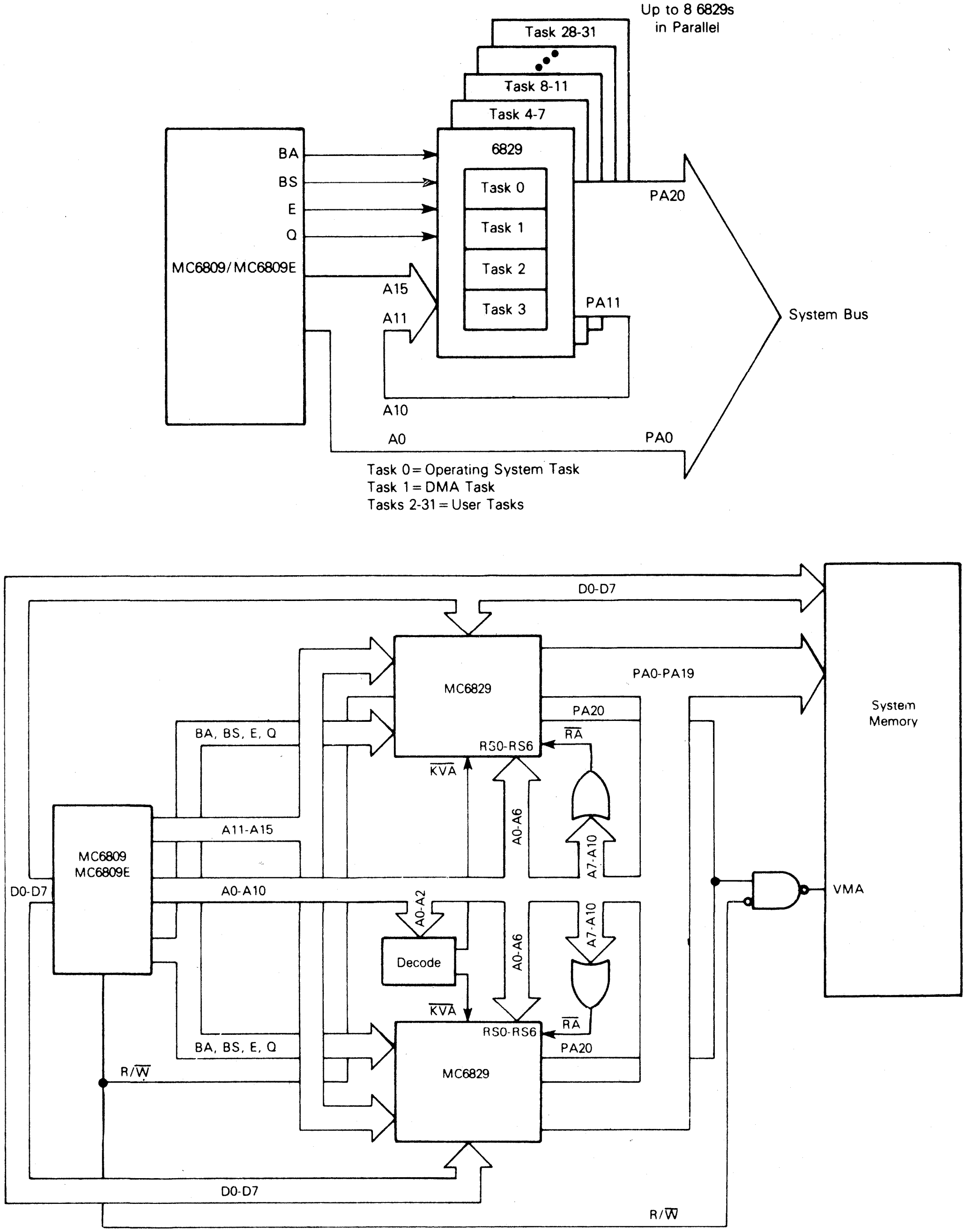
TABLE 1 — REGISTER SELECT TRUTH TABLE

$\overline{RA}$	R/$\overline{W}$	$\overline{KVA}$	RS6	RS5	RS4	RS3	RS2	RS1	RS0	register addressed
1	X	X	X	X	X	X	X	X	X	none
0	X	1	1	0	0	0	X	X	X	none
0	1	0	1	0	0	0	X	X	X	read Key Value Register
0	0	0	1	0	0	0	X	X	X	write Key Value Register
0	X	X	0	n	n	n	n	n	n	byte nnnnnn of MMU RAM (Note 1)
0	0	X	1	0	0	1	0	0	0	none (Note 2)
0	0	X	1	0	0	1	0	0	1	write Fuse Register
0	0	X	1	0	0	1	0	1	0	write Access Key
0	0	X	1	0	0	1	0	1	1	write Operate Key
0	1	X	1	0	0	1	0	0	0	read S-bit (Note 3)
0	1	X	1	0	0	1	0	0	1	read Fuse Register (Note 3)
0	1	X	1	0	0	1	0	1	0	read Access Key (Note 3)
0	1	X	1	0	0	1	0	1	1	read Operate Key (Note 3)
0	X	X	1	0	0	1	1	X	X	none
0	X	X	1	0	1	X	X	X	X	none
0	X	X	1	1	X	X	X	X	X	none

Notes:

1. The MMU RAM is accessible only if the Key Value Register is equal to the top 3 bits of the Access Key Register The lower two bits of the Access Key Register then determines which task is to be accessed (R/$\overline{W}$).
2. The S-bit is read-only.
3. The S-bit, Fuse, Access or Operate registers are readable only if the Key Value Register is equal to the top 3 bits of the Access Key Register. This insures that only one MMU will respond to a read request of these locations.

FIGURE 6 — MMU SYSTEMS CONFIGURATION



Task 0 = Operating System Task
Task 1 = DMA Task
Tasks 2-31 = User Tasks

REGISTER DESCRIPTION

System Bit (S-bit) — Read-only bit that must be set (S = 1) to write MMU registers. Reset and Interrupts set the S-bit. Refer to Fuse Register for clearing the S-bit.

Operate Key — 5-bit R/$\overline{W}$ register that contains the current task number. The operate key retains its value until explicitly changed. During DMA transfers, the MMU overrides the value in the operate key and forces task #1 to be the active task. When the S-bit is set, the operate key is also overridden, and task #0 is forced to be the active key.

Key Value — 3-bit R/$\overline{W}$ register that contains the range of tasks an MMU controls. The Key Value Register must match the top three bits of the Operate Key for a task to be active. The $\overline{KVA}$ signal must be low for an access of this register.

Access Key — 5-bit R/$\overline{W}$ register that contains the task number of a task to be viewed or changed. This register retains its value until explicitly changed.

Register #0 to #3F — 64 bytes accessed as 32 pairs of bytes with each pair describing the logical to physical mapping for one 2K page. Refer to Figure 5.

Fuse Register — 3-bit count down register used to change from task #0 to a user task. When a write to this register is detected, the value written is loaded into the counter and it begins to decrement by one for every processor cycle. When the counter underflows, the S-bit is cleared and the next processor cycle will be mapped using the task number in the operate key

RESET OPERATION

When reset, the MMU performs the following operations:

1. The Key Value Register is cleared;
2. The Fuse Register is disabled;
3. The System bit (S-bit) is set;
4. The Operate Key Register is cleared;
5. The Access Key Register is cleared;
6. An internal reset flag is set.

Reset causes the MC6829 to automatically switch the memory map to task #0. An internal flag is set causing all bus cycles to access physical addresses $1FF800-$1FFFFF (PA11 to PA20 all high, page $3FF). This flag is cleared when the Key Value Register is first written. While the internal reset flag is set, each MMU in the system will be actively driving the address bus. An orderly start up procedure must assign each MMU a key value before individual task allocations are made.

FUSE REGISTER OPERATION

The Fuse Register is a 3-bit register used to switch from task #0 to any other task. A write to this register causes an internal 3-bit counter to be loaded with the data. On each successive valid (non-DMA) processor cycle the internal counter is decremented once. When the counter reaches zero, the task number in the Operate Key will be the active task, mapping logical to physical address. The value written into the Fuse Register must be the number of cycles it takes to transfer program control from the store to Fuse Register instruction. It is the responsibility of the Operating System (task #0) to make sure the processor will execute code from the new task properly by changing the Program Counter the same cycle that the Fuse Register reaches zero (see following example).

Change from Task #0 to Task n

```
LDA #n
STA OPERATE
LDA #4
STA FUSE
JMP $XXXX
```

Cycle by Cycle Operation	Write #4 to Fuse Register	JMP	Address High	Address Low	$\overline{VMA}$	Task N Opcode
Fuse Register Contents	0	4	3	2	1	0

Refer to Section *MMU in a MC6809 System* for Fuse Register use in returning from an interrupt.

MMU INITIALIZATION PROCEDURE

The following steps should be followed to initialize a multiple MMU system. (Refer to Hardware/Programming Considerations; Programming Examples section.)

1. Out of Reset, all MMUs are driving the address lines, PA11 to PA20, high. This requires the initialization program to be located in this 2K byte page of physical memory. Each MMU must be deselected by writing a unique value to its Key Value Register except for the MMU that will run task #0 (MMU0). MMU0's Key Value Register must not be written to until task #0 registers $00 to $3F are programmed, specifying the logical to physical mapping of memory. In addition, if MMU0 Key Value Register is also initialized with a non-zero value at this time the entire memory space is deselected and the operating system (task #0) cannot be accessed (Example 1).
2. Only one MMU is now driving the address bus. Task #0 memory pages (2K per page) must be assigned by writing the corresponding values into registers $00 to $3F (Example 2).
3. The Key Value Register must be written to MMU0's key value to allow initialization of all other tasks by removal of automatic mapping of PA11 to PA20 high (Example 2).
4. At this time, each MMU has a unique key value, Task #0 has a specified memory map, and Task #0 is operating. Tasks can now be started by writing the task number to be specified in the Access Key Register, writing registers $00 to $3F to the memory map desired, loading the program into memory and causing a task switch by a correct use of the Fuse Register.

INTERRUPTS/MAP SWITCHING

The MC6829 monitors the Bus Available (BA) and Bus Status (BS) lines from the processor to determine what type of bus operation is occurring. When an interrupt is detected, the current task is overridden by Task #0. The map switch occurs during the processor vector fetch (BA = 0, BS = 1) so that Task #0 supplies the interrupt vector address. Detecting an interrupt also sets the S-bit within the MMU allowing Task #0 to be the operating task while the interrupt is serviced.

DMA OPERATION

For a DMA transfer, the memory map is switched to Task #1. This allows transfers of up to 64K bytes without processor intervention and without interfering with any other task. (An external DMA/$\overline{\text{VMA}}$ signal should be included in the decode circuitry for the $\overline{\text{RA}}$ input to prevent dead bus cycles from affecting the MMU). At the end of the DMA transfer, the MC6829 returns to the task being used before the transfer began (refer to Programming Considerations).

MMU IN A MC6809 SYSTEM

The MC6829 is designed to work directly with the MC6809 processor. Other 8-bit microcomputers may also use the MMU by generating the appropriate inputs to the MMU. The crucial area for interfacing the computer to the MMU is the design of the map switching hardware.

For the MC6809, the BA and BS signals are extremely useful for this function. Decoding these two signals provides the following information:

BA	BS	MC6809 State
0	0	Normal (running) mode
0	1	Interrupt Acknowledge (IACK)
1	0	SYNC Acknowledge
1	1	HALT or Bus Grant

The MMU uses these two signals directly from the processor to determine what action to take for every bus cycle.

The MMU, unlike other M6800 peripherals, introduces an additional delay (t_{MAD}) in the system configuration as it accepts address signals from the MPU and maps the MC6809 logical address to the system physical address. When a system is constructed this additional delay must be considered.

The system clock frequency is determined by these address timing delays. Figure 7 shows this data. The System Cycle time may be determined by adding:

1. the MPU E to Q rise delay t_{AVQ} (max)
2. the MPU address valid to Q rise to t_{AQ} (min)
3. the MMU mapping delay t_{MAD} (max)
4. the system decode and buffer time t_B (this is the delay due to bus buffers and decoding circuitry)
5. the address setup time required by peripherals t_{AS} (note the setup time is required for the peripheral to determine if it is selected as well as deselected during every bus cycle).
6. the MPU pulse width high t_{PWEH}.

NOTE

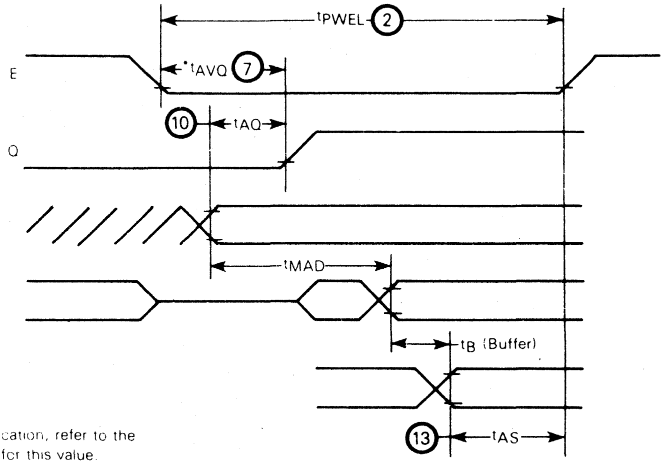
This equation must be satisfied:

$$t_{PWEL} \geq t_{AVQ} - t_{AQ} + t_{MAD} + t_B + t_{AS}$$

DMA OPERATION — By decoding the bus grant signal (BA = 1, BS = 1), the MMU will automatically switch to Task #1. Even when the MC6809 occasionally steals back a cycle to refresh its internal buses, this is reflected by a change in the bus grant signal which causes the map to temporarily switch back to the normal running mode.

Note that the bus grant status is identical to the Halt status and is thus indistinguishable from a HALT. This should not cause a problem since halting the processor will simply cause the MMU to switch to Task #1. When the MC6809 starts to run again, the status lines will change and cause the MMU to switch to the proper map.

FIGURE 7 — ADDRESS DELAY



t_{AQ} is a MPU specification, refer to the MC6809 Data Sheet for this value.

CHANGING TASK TO OPERATING SYSTEM (OS) – The OS map (Task #0) is automatically selected to service all interrupts. The Interrupt Acknowledge (IACK; BA=0, BS=1) signal is used to determine when an interrupt vector is being fetched. The map is switched at this time in order to supply the processor with an interrupt vector from the OS address space, not the user's. At the time IACK is asserted, all of the registers have been stacked for the interrupt in the user's address map. This means that the only information the OS needs to save concerning the running process is its stack pointer. All other information about the task is saved on the user's stack and in the MMU registers. The map switch is latched since IACK will only be present for two machine cycles, yet the OS must retain control until the interrupt is serviced. This latched information is kept in a flag register called the S-bit. This bit is set on any IACK and remains set until cleared by software. The first thing the OS must do is save the interrupted task's stack pointer in a table and load the stack pointer with the current top of stack in the OS map. This is a critical section of code and must not be interrupted. For this reason, an MMU system cannot accept two interrupts in a row. The first interrupt causes the map to switch to task zero. The second interrupt would stack the machine state at the wrong address in the operating system. As a consequence of this, Non-Maskable Interrupts (NMI) must be forbidden in multi-tasking systems since an NMI is possible at any time (even during another interrupt). Similarly, normal interrupts (IRQ) do not set the Fast Interrupt (FIRQ), bit F of the status register, in the processor and, thus, potentially allow another interrupt before the processor has a chance to switch stack pointers. Simple external hardware can be used to disable FIRQ when IRQ is pending. Unlike the NMI input, the FIRQ input is level sensitive and may be masked with external hardware during IRQ operations.

A typical interrupt service routine begins like this:

```
ORCC    #1+F
STS     SAVESP
LDS     OSSP
```

RETURNING FROM THE OS TO TASK N – The OS must execute an RTI instruction to get the processor to reload the user registers. The map switch must occur after the opcode for the RTI is fetched and before the first register is pulled from the stack. Prior to the RTI, the OS must reload the stack pointer from the one that corresponds to the task about to run. There must be no interrupts from the time the stack pointer is reloaded until the RTI is executed. The signal to the MMU that the map should be returned to the user task is noted by a write to a 3-bit down counter called the FUSE REGISTER. When a write to this register is detected, the value written is loaded into the counter and it begins to decrement by one for every processor cycle. When the counter under flows, the S-bit is cleared and the next processor cycle will be mapped using the task number in the Operate Key. For most systems, a 1 would be written to the Fuse Register immediately before the RTI opcode is executed. Note that DMA operations are still possible within this critical section. The Fuse Register counts only non-DMA cycles after the write to the Fuse Register in order to be sure of when to switch the map. Bus dead cycles are also excluded when clocking the Fuse Register. Thus, the Fuse Register is inhibited from counting whenever BA is high, and for the cycle after BA transitions from high to low. The common exit point for all OS functions looks something like this:

```
EXIT    LDA     TASK        GET NEXT TASK TO RUN
        STA     OPERAT      AND PLACE IT IN THE OPERATE KEY
        STS     OSSP        SAVE CURRENT STACK POINTER
        ORCC    #F+I        SET F AND I (ENTER CRITICAL SECTION)
        LDS     SAVESP      RESTORE USER'S STACK POINTER
        LDA     #1          CAUSE MAP SWITCH 1 CYCLE AFTER
        STA     FUSE        WRITE TO FUSE REGISTER
        RTI                 RETURN TO USER TASK
        •
        •
        •       MAP SWITCH OCCURS, USER TASK RESUMES
```

USING THE MC6800

When using a MC6800 processor external logic is required to determine when to switch maps. The MMU is controlled by its BA, BS inputs, the S-bit and the Operate Key. For example, decoding any references to the interrupt vectors and generating IACK as a result will work as long as each task references these locations only when the processor is fetching an interrupt vector. Another possibility is to monitor the processor R/W line. For the MC6800, the only time seven writes occur in a row is during an interrupt sequence. Thus the external logic that generates BA and BS must wait until it sees the seven writes and then assert IACK for the next two cycles.

A MC6800 processor interface to the MMU must also include logic to generate the Q bus signal.

HARDWARE/PROGRAMMING CONSIDERATIONS

The following sections contain examples and suggestions on how to apply the MMU in a system.

MEMORY PROTECTION – The MMU can provide memory protection on a per page basis by defining the high order physical address line (PA20) as a write access line. If write protection is desired, this signal can be gated with the read/write line, from the processor, to generate a disable signal. This can be used to inhibit the memory chip select logic or generate an interrupt to signal a violation of a write protected area. The write protect line can also be combined with the DMA/VMA logic that is necessary in systems using DMA. In this case, writes to protected memory would appear as dead cycles to the main memory. Note that the designtion of the write protect line is purely arbitrary. The MMU simply combines the incoming address with the current task number to determine a 10-bit result. If no write protection is needed, PA20 can be used as a 21st address line, giving a total addressing range of 2 Megabyte. This scheme can be reversed if desired and additional output lines from the MMU can be used to specify more attributes of the physical pages at the expense of reducing the number of pages in physical memory.

MANAGING INTERRUPTS – An interrupt causes the processor to suspend the current running task and perform a service routine for the interrupting device. User programs should not have to handle interrupts directly. Thus on interrupts, the MMU (the operating system OS) must switch from the current map to task 0 so that it can handle the interrupt. (The OS may of course elect to pass the work of handling a specific interrupt to a task that is expecting it.) The map switching is latched (indicated by the S-bit) so that the processor has as much time as it needs to service the interrupt. After the interrupt has been processed, the OS can then look at the current process priorities and determine the next process to run. If, after the interrupt service, the task that was running before the interrupt is to continue to run, the OS causes the map to switch back to that task. If, however, another task is to start running, the OS can simply write the new task number into the Operate Key Register and then cause the map switch. Returning to the normal map clears the S-bit and allows the user process to continue. By supplying a source of periodic interrupts, the OS can regain control of the processor and reschedule running processes.

Operating system requests for privileged operations by running tasks are ideally handled using the SWI instruction. This causes a map switch to task zero (IACK is asserted on SWI) which then processes the request and eventually returns control to the requesting task. Note that SWI sets the I and F bits during execution of the instruction so that when the OS is entered, the critical section of saving the user task pointer and reloading the OS stack pointer can be safely executed. Note that SWI2 and SWI3 do not have this property and therefore require special handling. To safely use SWI2 or SWI3, the programmer must explicitly mask hardware interrupts.

```
ORCC      #I+F        DISABLE INTERRUPTS
SWI2/3                CALL OS
```

MANAGING NON-EXISTENT MEMORY ACCESSES – Memory accesses to non-existent memory requires careful consideration. Once an instruction has begun execution, there is no way to stop it from completing. Thus, an instruction may reference a non-existent memory location, or an interrupt may cause the machine state to be stacked into non-existent memory. Once this has occurred, there is not always enough information available to backtrack the last instruction.

One solution to this problem is a hardware FIFO. When a task is initialized, a certain number of pages will be assigned from available memory. For example, a ROM program could be placed in a task's map along with RAM for stack and variable data areas. The remaining pages in the task's map are unassigned and references to these unassigned areas require special handling. These gaps in the memory map of a task may be filled by constructing a "FIFO page" that returns a known value when read (zero) and when written saves the (logical) address and the data written to it. If at any time the FIFO is not empty, the FIFO causes an interrupt at the end of the current instruction. The processor then examines the contents of the FIFO and allocates real pages where there were none before. The data in the FIFO is then placed in real memory and the task may resume execution. Thus, the program is stopped at the end of the instruction that causes a page fault, and all writes to non-existent memory are captured in the FIFO.

The maximum number of new pages that may be required after any page fault is four. Consider the following instruction sequence. A task has just started running and has only one page allocated to it ($0000-$1FFF). The program to be executed is as follows:

```
ORG       $0000       PROGRAM START ADDRESS
LDS       #$8000      INITIALIZE STACK
LDX       #$3FFF      POINT TO DATA AREA
LDD       #$1234
STD       ,X          INITIALIZE VARIABLE
```

Execution then proceeds as follows. Upon executing the fourth instruction, two bytes are written, one at location $3FFF and the other at $4000. Since neither of these two pages actually exist, the FIFO catches the address and data written and pulls the IRQ line to signal a page fault. At the end of the STD instruction, the processor will stack the machine registers which causes two further page faults since the stacking operation writes data to locations $7FF5-$8000. The FIFO must also catch these references since they contain the machine state at the time of the original interrupt. When task zero gains control, the FIFO data must be cleared before any attempt is made to reference the task's memory map. If there are no available pages, the task may be made inactive until sufficient space exists to allow the program to continue.

The maximum number of bytes that may be written to non-existent memory before task zero gains control is 24. This occurs when the task pushes all of its registers onto the stack when the stack points to an uninitialized page. Pushing all registers requires 12 bytes. At the end of the instruction, an interrupt will be generated which again pushes the entire machine state. Thus, the FIFO must be 24 bits wide (16 address + 8 data lines) and 24 words deep.

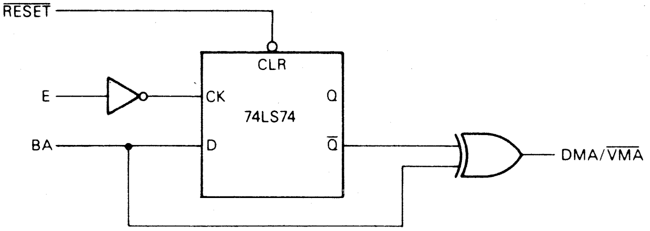
The primary benefit of this scheme is to allow the MC6809 stack to grow dynamically. When a task starts to run, the stack could be initialized to $FFFF with no real memory at that location. When the task did its first subroutine call or stack push, the FIFO interrupt would catch the information and the operating system would then allocate memory. If the task never used this area, it would remain unallocated and thus be available for other uses. Note that this approach provides for dynamic memory expansion of growing data areas. If the size of the static data areas is known at load-time, then memory can be allocated to a task as needed. Heap management (such as for an editor buffer) can be handled by task resident memory allocation routines which make operating system calls to obtain more heap space.

The FIFO scheme does not implement a demand paging system. It is assumed that once a page has been assigned to a task the page remains assigned until the task ends execution or possibly gives it back (via a system call) to the operating system.

DMA/$\overline{\text{VMA}}$ CIRCUIT

The following circuit, Figure 8 , is suggested to keep the MC6829 deselected during dead bus cycles of DMA. This circuit will also work in a non-MMU system.

FIGURE 8 — M6809 DMA/$\overline{\text{VMA}}$ LOGIC



COMMON MMU EQUATES

Here is a list of assembler equates that are used in the following examples:

```
MMU       EQU    $F800      START OF MMU REGISTERS (IN TASK 0)
MMU0      EQU    MMU+$40    FIRST MMU'S KEY VALUE REGISTER
MMU7      EQU    MMU+$47    LAST MMU'S KEY VALUE REGISTER
SBIT      EQU    MMU+$48    SYSTEM/USER FLAG BIT
FUSE      EQU    MMU+$49    MAP SWITCH COUNT-DOWN REGISTER
ACCESS    EQU    MMU+$4A    ACCESS.KEY
OPERAT    EQU    MMU+$4B    OPERATE KEY
NTASK     EQU    32         NUMBER OF TASKS IN SYSTEM
NPAGE     EQU    32         NUMBER OF PAGES PER TASK
MAXPGE    EQU    $400       MAXIMUM NUMBER OF PAGES IN SYSTEM
PSIZE     EQU    2048       NUMBER OF BYTES IN A PAGE
```

Programming Examples

Example #1 — Only works if $\overline{\text{KVA}}$ is fully decoded

Write a program to initialize all MMU Key Value Registers except MMU0.

```
•
•           RESET ENTRY POINT FOR MMU SYSTEM
•
            LDX         #MMU7+1         POINT TO LAST MMU KEY VALUE REGISTER +1
            LDA         #7              INITIALIZE VALUE
KVINIT      STA         ,-X
            DECA
            BNE         KVINIT
            •
            •           CONTINUE INITIALIZATION
            •
```

At this point, each MMU will have a unique key value. Note that the Key Value Register for MMU0 has not yet been written so that page $3FF is still on the physical address bus. The difference is that now only one MMU is driving the address bus.

Example #2 —

Write an initialization program that sets up the pages of Task #0 so that an address $XXXX in Task #0 corresponds to physical address $1FXXXX.

```
            •
            •           FROM KEY VALUE INITIALIZATION
•           •
•           NOW INITIALIZE IDENTITY MAP FOR TASK 0
•
            CLR         ACCESS      TALK TO TASK 0 (ALREADY ZERO ANYWAY)
            LDX         #MMU
            LDD         #$3E0       LAST PAGE -32
MOINIT      STD         ,X++
            INCB                    QUIT WHEN D=$200
            BNE         MOINIT
            CLR         MMU0        LET MMU #0 GO
            JMP         EXBUG       TRANSFER TO MONITOR (EXBUG09)
```

Example #3 —

Give task #9 physical page #88 and place it in the task's address space so that #9 refers to this page with addresses $1000-$17FF. Write protect this page for this task. (The write protect bit is defined as PA20 of the MMU.)

```
PROTEC      EQU         $200        WRITE PROTECT BIT POSITION (PA20)
            •
            •
            •
            LDA         #9          SELECT TASK #9 FOR
            STA         ACCESS      MODIFICATION
            LDX         #88+PROTEC  WRITE PHYSICAL PAGE INTO
            STX         MMU+4       THE APPROPRIATE REGISTER
            •
            •
```

Example #4 —

Write a subroutine that reads a byte from any task. On entry, the A register contains the task number, and the X register contains the address of that task to read. Assume that the OS task has its third page free for this use. The byte that is read is returned in A.

```
FPAGE       EQU         $1000       DEDICATED FREE PAGE
FREE        EQU         4           OFFSET INTO MMU OF FPAGE
•
•           FUBYTE - FETCH USER BYTE
•
FUBYTE      LBSR        GETPAGE     POINT TO PAGE
            LDA         ,X          PICKUP BYTE
            RTS
```

Example #5 –

Write a subroutine that writes a byte to any task. On entry the A register contains the task number and the X register contains the address of that task to read. The B register contains the byte to place in the task's memory. Assume that the OS task has its third page free for this use.

```
*
*        SUBYTE – SET USER BYTE
*
SUBYTE   LBSR        GETPAGE PLACE USER PAGE IN FPAGE
         STB         ,X
         RTS
```

Example #6 –

Write a subroutine to be given a task number and memory address that returns a pointer to that byte of the named task. On entry, the A register contains the task number and the X register contains the task address.

```
* GET PAGE – POINT TO USER BYTE
* Given a task number in A and a task address in X,
* return with X pointing to that byte in task 0.
* This subroutine assumes that task 0 has a free
* page (FPAGE) that it uses to map a page of the
* specified task into task 0's map.
*
*
GETPAGE   PSHS    D, Y          SAVE SOME REGISTERS
          STA     ACCESS        SETUP WINDOW TO TASK
          TFR     X, D          MOVE POINTER INTO ACCUMULATOR
          ASRA                  FIND PHYSICAL PAGE #
          ASRA
          ANDA    #%00111110    MASK ALL BUT PAGE #
          LDY     #MMU
          LDY     A, Y          PICKUP PAGE
          CLR     ACCESS        NOW TALK TO OS MAP
          STY     MMU+FREE      'FREE' OS PAGE
          TFR     X, D          NOW POINT TO OFFSET
          ANDA    #%111         MASK HIGH BITS OF ADDRESS
          LDX     #FPAGE        POINT TO PAGE START
          LEAX    D, X          ADD OFFSET
          PULS    D, Y, PC      RESTORE AND RETURN
```

The above method of fetching bytes from other tasks is appropriate where only a few bytes of memory are to be transferred. When larger amounts of memory are to be moved, a more general subroutine can be written that transfers up to 2K bytes (one page) before the MMU registers need to be changed.

MOTOROLA

MC6835
(1.0 MHz)

MC68A35 MC68B35
(1.5 MHz) (2.0 MHz)

MOS

(HIGH-DENSITY, N-CHANNEL, SILICON-GATE DEPLETION LOAD)

MASK PROGRAMMED CRTC CONTROLLER (CRTC)

Advance Information

CRT CONTROLLER (CRTC)

The MC6835 is a ROM based CRT Controller which interfaces an MPU system to a raster scan CRT display. It is intended for use in MPU based controllers for CRT terminals in stand-alone or cluster configurations. The MC6835 supports two selectable mask programmed screen formats using the program select input (PROG).

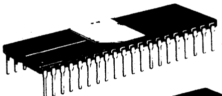
The CRTC is optimized for the hardware/software balance required for maximum flexibility. All keyboard functions, reads, writes, cursor movements, scrolling, and editing are under processor control. The mask programmed registers of the CRTC are programmed to control the video format and timing.

- Cost Effective ROM Based CRTC Which Supports Two Screen Formats
- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games; Information Displays
- Alphanumeric, Semigraphic, and Full Graphic Capability
- Timing May Be Generated for Almost Any Alphanumeric Screen Format, e.g., 80 × 24, 72 × 64, 132 × 20
- Single +5 Volt Supply
- M6800 Compatible Bus Interface
- TTL-Compatible Inputs and Outputs
- Start Address Register Provides Hardware Scroll (By Page, Line, or Character)
- Programmable Cursor Register Allows Control of Cursor Position
- Refresh (Screen) Memory May Be Multiplexed Between the CRTC and the MPU Thus Removing the Requirements for Line Buffers or External DMA Devices
- Mask Programmable Interlace or Non-Interlace Scan Modes
- 14-Bit Refresh Address Allows Up to 16K of Refresh Memory for Use in Character or Semigraphic Displays
- 5-Bit Row Address Allows up to 32 Scan-Line Character Blocks
- By Utilizing Both the Refresh Addresses and the Row Addresses, a 512K Address Space is Available for Use in Graphics Systems
- Refresh Addresses are Provided During Retrace, Allowing the CRTC to provide Row Addresses to Refresh Dynamic RAMs
- Pin Compatible with the MC6845. The MC6845 May Be Used as a Prototype Part to Emulate the MC6835.

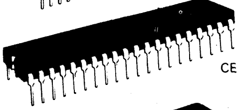
MAXIMUM RATINGS

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 MC6835, MC68A35, MC68B35 MC6835C, MC68A35C, MC68B35C	T_A	0 to +70 −50 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

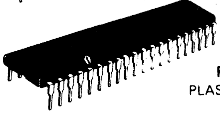
*With respect to GND (V_{SS}).



L SUFFIX
CERAMIC PACKAGE
CASE 715



S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 — PIN ASSIGNMENTS

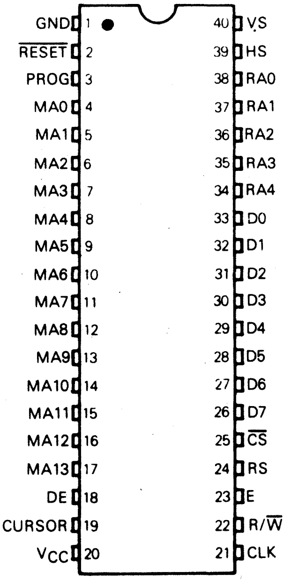
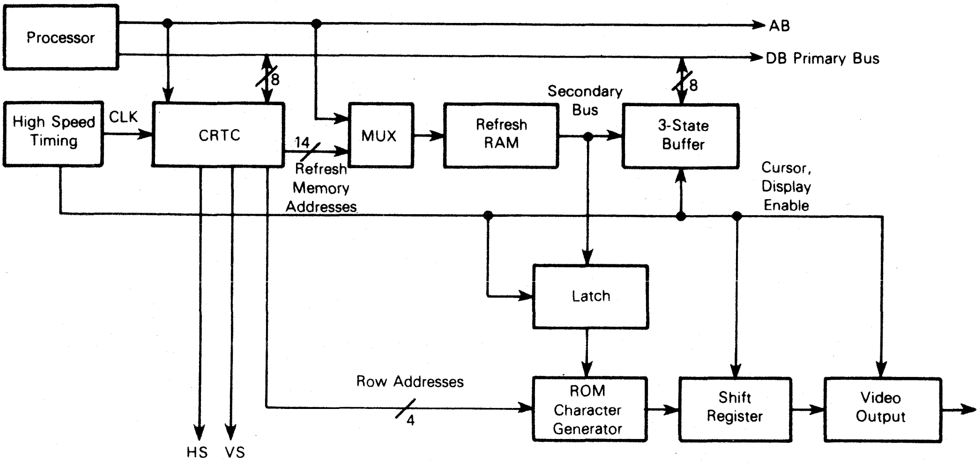


FIGURE 2 — TYPICAL CRT CONTROLLER APPLICATION



THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic Cerdip Ceramic	θ_{JA}	100 60 50	°C/W

RECOMMENDED OPERATING CONDITIONS

Characteristic	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V

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 $V_{out}) \leq V_{CC}$. 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:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS ($V_{CC}=5.0$ Vdc ±5%, $V_{SS}=0$, $T_A=0$ to 70°C unless otherwise noted) (Reference Figures 3-5)

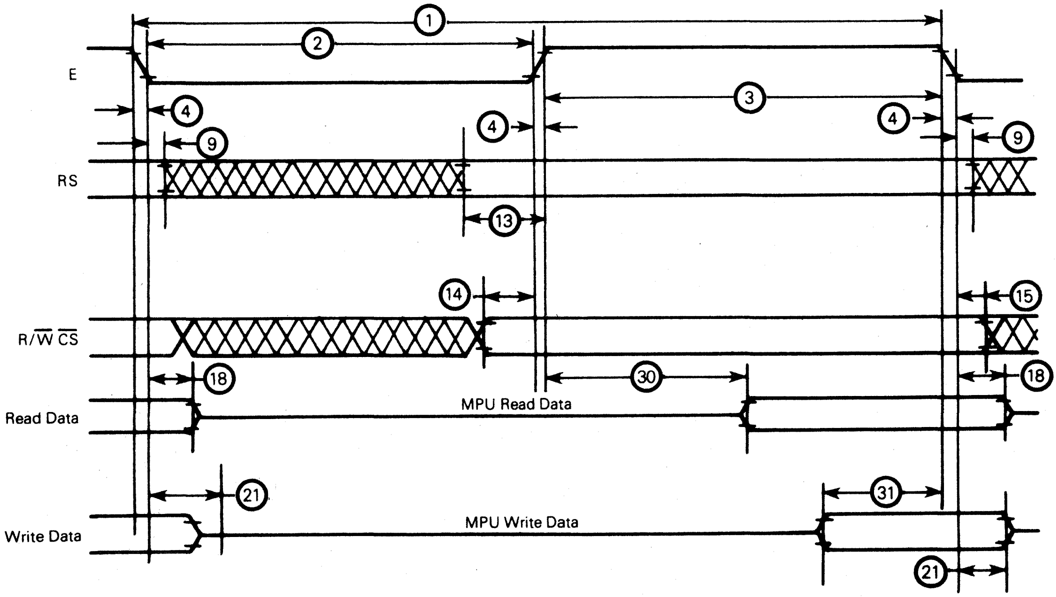
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	−0.3	–	0.8	V
Input Leakage Current		I_{in}	–	0.1	2.5	μA
Three-State ($V_{CC}=5.25$ V) ($V_{in}=0.4$ to 2.4 V)		I_{TSI}	−10	–	10	μA
Output High Voltage ($I_{load}=-205\ \mu A$)	D0-D7	V_{OH}	2.4	3.0	–	V
($I_{Load}=-100\ \mu A$)	Other Outputs		2.4	3.0	–	
Output Low Voltage ($I_{load}=1.6$ mA)		V_{OL}	–	0.3	0.4	V
Internal Power Dissipation (Measured at $T_A=T_L$)		P_D	–	600	750	mW
Input Capacitance	D0-D7	C_{in}	–	–	12.5	pF
	All Others		–	–	10	
Output Capacitance	All Outputs	C_{out}	–	–	10	pF

BUS TIMING CHARACTERISTICS (Reference Figures 3 and 4)

Ident. Number	Characteristics	Symbol	MC6835		MC68A35		MC68B35		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Transition Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time (RS)	t_{AH}	10	–	10	–	10	–	ns
13	RS Setup Before E	t_{AS}	80	–	60	–	40	–	ns
14	R/$\overline{W}$ and $\overline{CS}$ Setup Before E	t_{CS}	80	–	60	–	40	–	ns
15	Hold Time for R/$\overline{W}$ and $\overline{CS}$	t_{CH}	10	–	10	–	10	–	ns
18	Peripheral Read Data Hold Time Provided	t_{DDR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time Required	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay	t_{DDR}	–	290	–	180	0	150	ns
31	Peripheral Input Data Setup	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (high impedance).

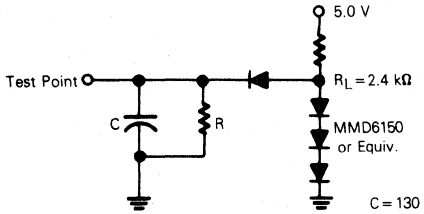
FIGURE 3 — MC6835 BUS TIMING



CRTC TIMING CHARACTERISTICS (See Figure 5)

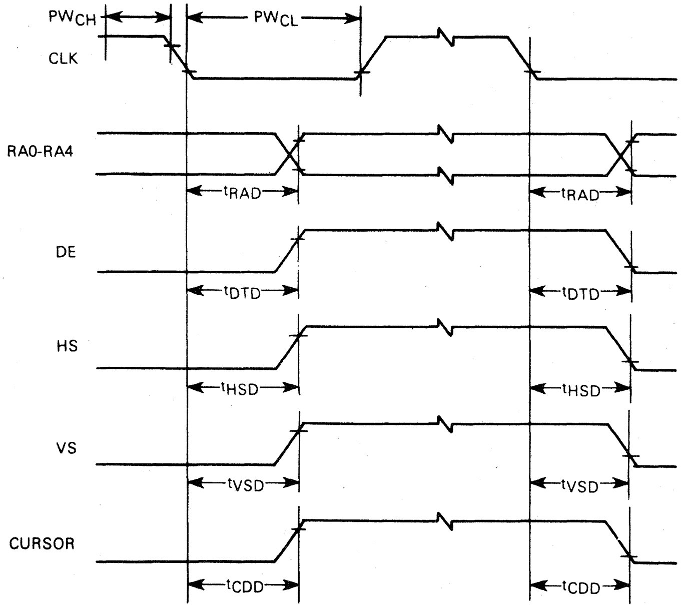
Characteristics	Symbol	Min	Max	Unit
Minimum Clock Pulse Width, Low	PW_{CL}	160	–	ns
Minimum Clock Pulse Width, High	PW_{CH}	200	–	ns
Clock Frequency	f_c	–	2.5	MHz
Rise and Fall Time for Clock Input	t_{cr}, t_{cf}	–	20	ns
Memory Address Delay Time	t_{MAD}	–	160	ns
Raster Address Delay Time	t_{RAD}	–	160	ns
Display Timing Delay Time	t_{DTD}	–	300	ns
Horizontal Sync Delay Time	t_{HSD}	–	300	ns
Vertical Sync Delay Time	t_{VSD}	–	300	ns
Cursor Display Timing Delay Time	t_{CDD}	–	300	ns

FIGURE 4 – BUS TIMING TEST LOAD



C = 130 pF for D0-D7
= 30 pF for MA0-MA13, RA0-RA4, DE, HS, VS, and CURSOR
R = 11 kΩ for D0-D7
= 24 kΩ for All Other Outputs

FIGURE 5 – CRTC TIMING CHART



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.

CRTC INTERFACE SYSTEM DESCRIPTION

The MC6835 CRT Controller generates the signals necessary to interface a digital system to a raster scan CRT display. In this type of display, an electron beam starts in the upper left hand corner, moves quickly across the screen and returns. This action is called a horizontal scan. After each horizontal scan the beam is incrementally moved down in the vertical direction until it has reached the bottom. At this point one frame has been displayed, as the beam has made many horizontal scans and one vertical scan.

Two types of raster scanning are used in CRTs, interlace and non-interlace, shown in Figures 6 and 7. Non-interlacing scanning consists of one field per frame. The scan lines in Figure 6 are shown as solid lines and the retrace patterns are indicated by the dotted lines. Increasing the number of frames per second will decrease the flicker. Ordinarily, either a 50 or 60 frame per second refresh rate is used to minimize beating between the frequency of the CRT horizontal oscillator and the power line frequency. This prevents the displayed data from weaving or swimming.

Interlace scanning is used in broadcast TV and on data monitors where high density or high resolution data must be displayed. Two fields, or vertical scans are made down the screen for each single picture or frame. The first field (Even field) starts in the upper left hand corner; the second (Odd field) in the upper center. Both fields overlap as shown in Figure 7, thus interlacing the two fields into a single frame.

In order to display the characters on the CRT screen the frames must be continually repeated. The data to be displayed is stored in the Refresh (Screen) memory by the MPU controlling the data processing system. The data is usually written in ASCII code, so it cannot be directly displayed as characters. A Character Generator ROM is typically used to convert the ASCII codes into the "dot" pattern for every character.

The most common method of generating characters is to create a matrix of "x" dots (columns) wide and "y" dots (rows) high. Each character is created by selectively filling in the dots. As "x" and "y" get larger a more detailed character may be created. Two common dot matrices are 5×7 and 7×9. Many variations of these standards will allow Chinese, Japanese, or Arabic letters instead of English. Since characters require some space between them, a character block larger than the character is typically used as shown in Figure 8. The figure also shows the corresponding timing and levels for a video signal that would generate the characters.

FIGURE 6 — RASTER SCAN SYSTEM (NON-INTERLACE)

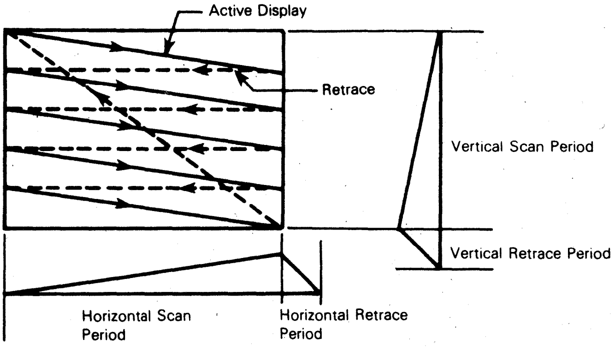


FIGURE 7 — RASTER SCAN SYSTEM (INTERLACE)

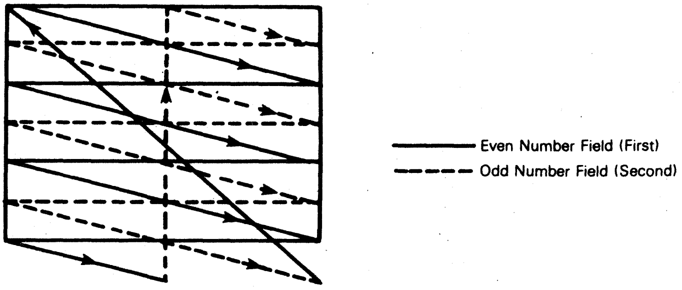
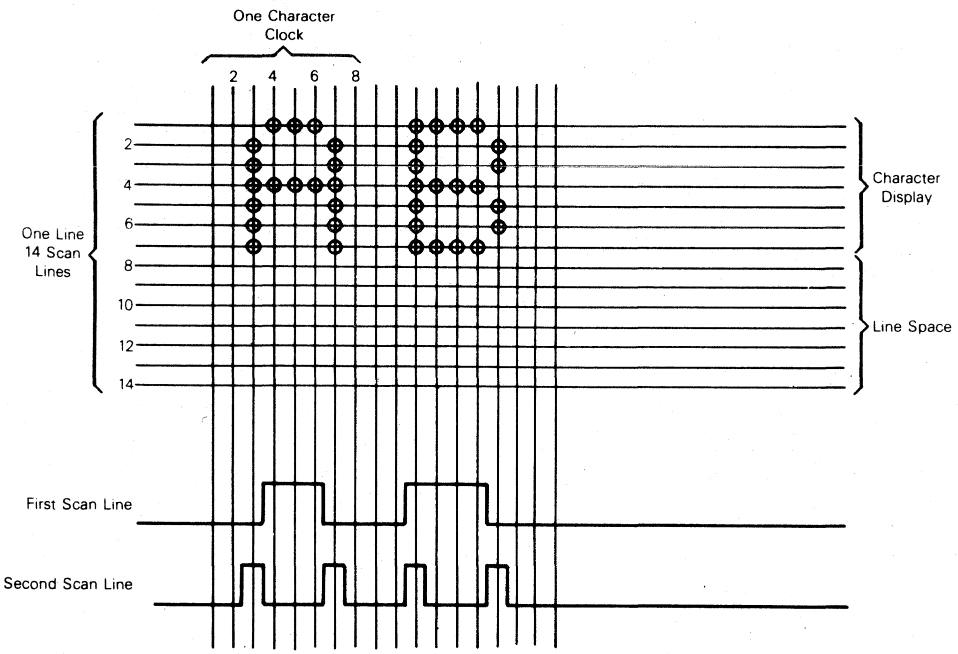


FIGURE 8 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL



Referring to Figure 2, the MC6835 CRT controller generates the Refresh addresses (MA0-MA13), row addresses (RA0-RA4), and the video timing (vertical sync — VS, horizontal sync — HS and display enable — DE). Other functions include an internal cursor register which generates a Cursor output when its contents compare to the current Refresh address. A select input, PROG, allows selection of one of two mask programmed video formats (e.g., for 50 Hz and 60 Hz compatibility).

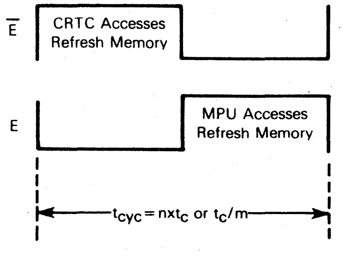
All timing in the CRTC is derived from the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high speed logic (TTL) to generate the CLK signal. The high speed logic must also generate the timing and control signals necessary for the Shift Register, Latch and MUX Control shown in Figure 2.

The processor communicates with the CRTC through an 8-bit data bus by writing into the five user programmable registers of the MC6835.

The Refresh memory address is multiplexed between the processor and the CRTC. Data appears on a secondary bus separate from the processor's bus. The secondary data bus concept in no way precludes using the Refresh RAM for other purposes. It looks like any other RAM to the processor. A number of approaches are possible for solving contentions for the Refresh memory.

1. Processor always gets priority. (Generally, "hash" occurs as MPU and CRTC clocks are not synchronized.)
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize the processor with memory wait cycles (states).
4. Synchronize the processor to the character rate as shown in Figure 9. The M6800 processor family works very well in this configuration as constant cycle lengths are present. This method provides no overhead for the processor as there is never a contention for a memory access. All accesses are transparent.

FIGURE 9 — TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING M6800 FAMILY MPU



Where: m, n are integers; t_c is character period

PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{\text{CS}}$, RS, E, and R/$\overline{\text{W}}$ for control signals.

Data Bus (D0-D7) — The bidirectional data lines (D0-D7) allow data transfers between the internal CRTC register file and the processor. Data bus output drivers are high-impedance buffers which remain in the high-impedance state until the processor performs a CRTC read operation.

Enable (E) — The Enable signal is a high-impedance TTL/MOS-compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock. The high-to-low transition is the active edge.

Chip Select ($\overline{\text{CS}}$) — The $\overline{\text{CS}}$ line is an active-low high-impedance TTL/MOS-compatible input which selects the CRTC to read or write to the internal register file. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) — The RS line is a high-impedance TTL/MOS-compatible input which selects either the Address Register (RS = "0") or one of the Data Registers (RS = "1") of the internal register file when $\overline{\text{CS}}$ is low.

Read/Write (R/$\overline{\text{W}}$) — The R/W line is a high-impedance TTL/MOS-compatible input which determines whether the internal register file gets written or read. A write is defined as a low level.

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and display enable (DE) signals.

NOTE — Care should be exercised when interfacing to CRT monitors as many monitors claiming to be "TTL compatible," have transistor input circuits which require the CRTC or TTL devices buffering signals from the CRTC/video circuits to exceed the maximum rated drive currents.

Vertical Sync (VS) and Horizontal Sync (HS) — These TTL-compatible outputs are active-high signals which drive the monitor directly or are fed to the video processing circuitry to generate a composite video signal. The VS signal determines the vertical position of the displayed text while the HS signal determines the horizontal position of the displayed text.

Display Enable (DE) — This TTL-compatible output is an active-high signal which indicates the CRTC is providing addressing in the active Display Area.

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides Memory Addresses (MA0-MA13) to scan the Refresh RAM. Row Addresses (RA0-RA4) are also provided for use with character generator ROMs. In a graphics system both the Memory Addresses and the Row Addresses would be used to scan the Refresh RAM. Both the Memory Addresses and the Row Addresses continue to run during vertical retrace thus allowing the CRTC to provide the refresh addresses required to refresh dynamic RAMs.

Refresh Memory Addresses (MA0-MA13) — These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs are capable of driving one standard TTL load and 30 pF.

Row Addresses (RA0-RA4) — These five outputs from the internal Row Address counter are used to address the Character Generator ROM. These outputs are capable of driving one standard TTL load and 30 pF.

OTHER PINS

Cursor — This TTL-compatible output indicates a valid Cursor address to external video processing logic. It is an active-high signal.

Clock (CLK) — The CLK is a TTL/MOS-compatible input used to synchronize all CRT functions except for the processor interface. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high-to-low.

Program Select (PROG) — This TTL-compatible input allows selection of one of two sets of mask programmed video formats. Set zero is selected when PROG is low and set one is selected when PROG is high.

V_{CC}, GND — These inputs supply +5 Vdc ±5% to the CRTC.

$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input is used to reset the CRTC. Functionality of $\overline{\text{RESET}}$ differs from that of other M6800 parts. $\overline{\text{RESET}}$ must remain low for at least one cycle of the character clock (CLK). A low level on the $\overline{\text{RESET}}$ input forces the CRTC into the following state:

a. All counters in the CRTC are cleared and the device stops the display operation.,

b. All the outputs are driven low, except the MA0-MA13 outputs which are driven to the current value in the Start Address Register.

c. The control registers of the CRTC are not affected and remain unchanged.

d. The CRTC resumes the display operation immediately after the release of $\overline{\text{RESET}}$.

CRTC DESCRIPTION

The CRTC consists of mask-programmable horizontal and vertical timing generators, software-programmable linear address register, mask-programmable cursor logic and control circuitry for interfacing to a M6800 family microprocessor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the

TABLE 1 — INTERNAL REGISTER ASSIGNMENT

CS	RS	Address Register					Register #	Register File	Program Unit	Read	Write	Number of Bits							
		4	3	2	1	0						7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X	–	–	–	–	╲	╲	╲	╲	╲	╲	╲	╲
0	0	X	X	X	X	X	AR	Address Register	–	No	Yes	╲	╲	╲					
Note 3							R0	Horizontal Total	Char.	No	No								
							R1	Horizontal Displayed	Char.	No	No								
							R2	H. Sync Position	Char.	No	No								
							R3	Sync Width	–	No	No	V	V	V	V	H	H	H	H
							R4	Vertical Total	Char. Row	No	No	╲							
							R5	V. Total Adjust	Scan Line	No	No	╲	╲	╲					
							R6	Vertical Displayed	Char. Row	No	No	╲							
							R7	V. Sync Position	Char. Row	No	No	╲							
							R8	Interlace Mode and Skew	Note 1	No	No	C	C	D	D			I	I
							R9	Max Scan Line Address	Scan Line	No	No	╲	╲	╲					
							R10	Cursor Start	Scan Line	No	No	╲	B	P			(Note 2)		
							R11	Cursor End	Scan Line	No	No	╲	╲	╲					
0	1	0	1	1	0	0	R12	Start Address (H)	–	No	Yes	0	0						
0	1	0	1	1	0	1	R13	Start Address (L)	–	No	Yes								
0	1	0	1	1	1	0	R14	Cursor (H)	–	No	Yes	0	0						
0	1	0	1	1	1	1	R15	Cursor (L)	–	No	Yes								

NOTES:
1. The Interlace Control is shown in Table 2 while Skew Control is shown in Table 3.
2. Bit 5 of the Cursor Start Raster Register is used to blink period control, and Bit 6 is used to select blink or non-blink.
3. R0-R11 are mask-programmable and are not accessible via the data bus.

mask programmable register file, R0-R11. For horizontal timing generation, comparisons result in:

1. Horizontal sync pulse (HS) of a frequency, position and width determined by the register contents.
2. Horizontal Display signal of a frequency, position and duration determined by the register contents.

The horizontal counter produces H clock which drives the Scan Line Counter and Vertical Control. The contents of the Raster Counter are continuously compared to the Max Scan Line Address Register. A coincidence resets the Raster Counter and clocks the Vertical Counter.

Comparisons of Vertical Counter contents and Vertical Registers result in:

1. Vertical sync pulse (VS) of a frequency, position and width determined by the register contents.
2. Vertical Display signal of a frequency, position, and duration determined by the register contents.

The Vertical Control Logic has other functions:

1. Generate row selects, RA0-RA4, from the Raster Count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the Vertical Total Adjust Register.

The cursor logic determines the size and blink rate of the cursor as indicated by the register contents.

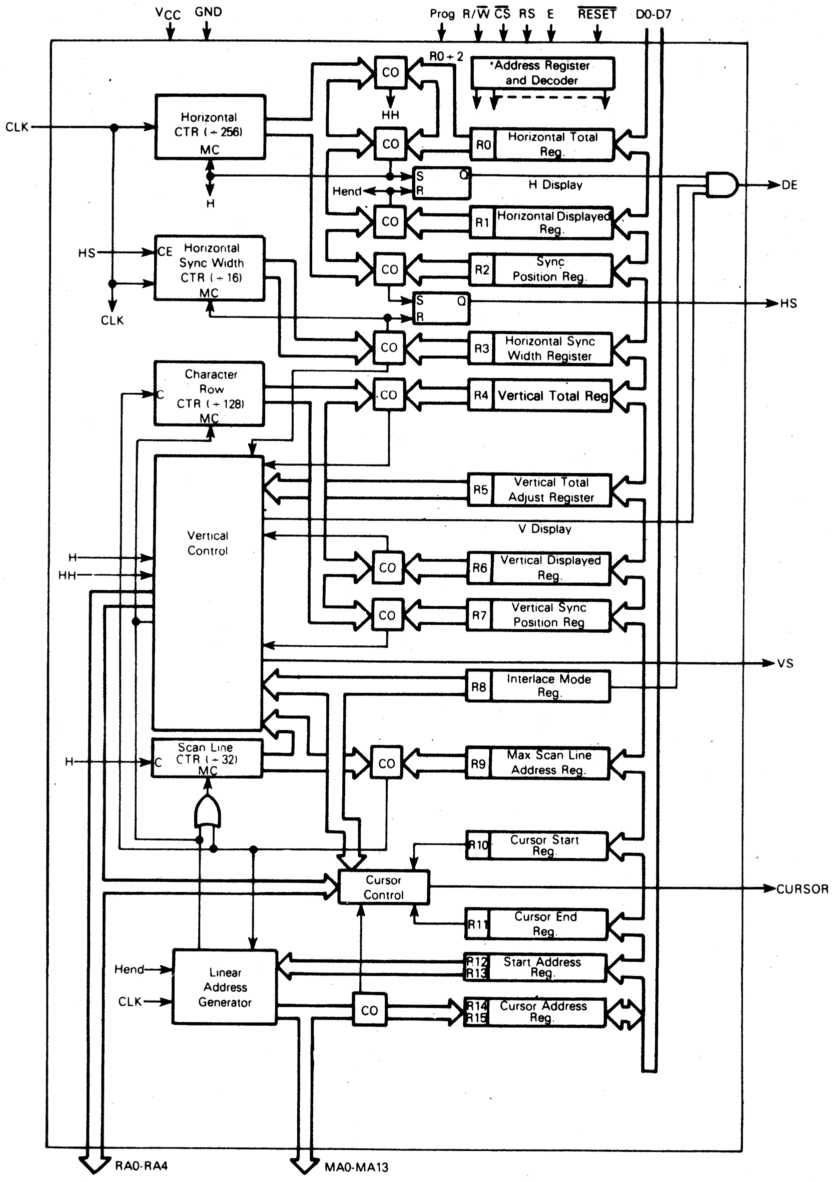
The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory and their positions on the screen. Fourteen outputs, MA0-MA13, are available for addressing up to four pages of 4K characters, eight pages of 2K characters, etc.

Five additional write-only registers define the Start Address and cursor position. Using the Start Address Register, hardware scrolling through 16K characters is possible. The Linear Address Generator repeats the same sequence of addresses for each scan line of a character row. The Start Address Register and the Cursor Position Register are programmed by the processor through the data bus, D0-D7 and the control signals — R/$\overline{\text{W}}$, $\overline{\text{CS}}$, RS, and E. Refer to Figure 10.

REGISTER FILE DESCRIPTION

The MC6835 has 17 control registers of which 12 are mask programmable. The remaining five registers — Address register, Start Address register pair, and Cursor Position register pair — are write-only registers programmed by the MPU. These registers control horizontal timing, vertical timing, interlace operation, row address operation and define the cursor, cursor address, start address and light pen register. The register addresses and sizes are shown in Table 1.

FIGURE 10 — CRTC BLOCK DIAGRAM



MASK PROGRAMMABLE REGISTERS R0-R11

The twelve mask programmable registers determine the display format generated by the MC6835. The PROG input is used to select one of two sets of register values.

Figure 11 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in character clock time units with respect to the reference as shown in Figure 12. The point of reference for the vertical registers is the top character position displayed. Vertical registers are programmed in character row times or scan line times as shown in Figure 13.

Horizontal Total Register (R0) — This 8-bit register determines the horizontal sync (HS) frequency by defining the HS period in character times. It is the total of the displayed characters plus the non-displayed character times (retrace) minus one.

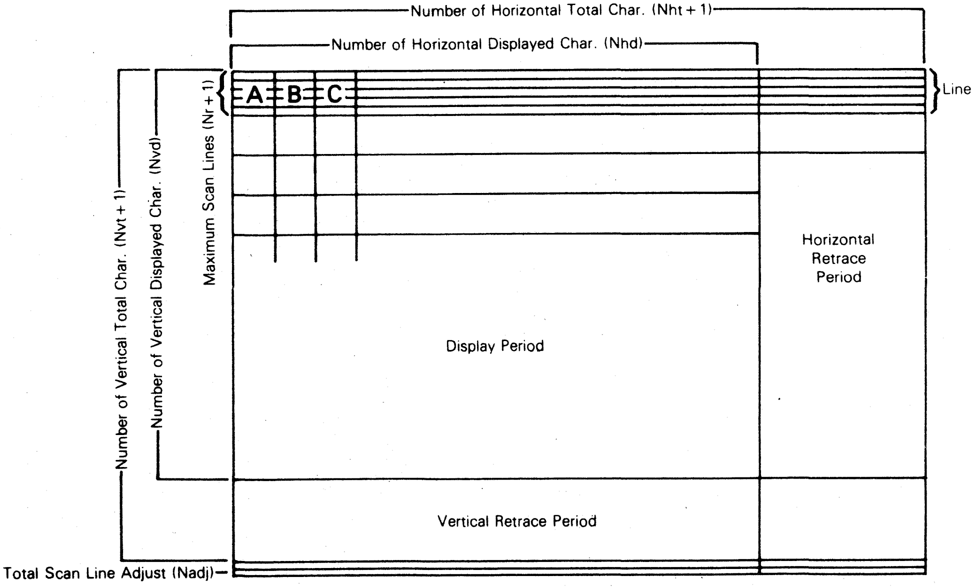
Horizontal Displayed Register (R1) — This 8-bit register determines the number of displayed characters per line. Any 8-bit number may be programmed as long as the contents of R0 are greater than the contents of R1.

Horizontal Sync Position Register (R2) — This 8-bit register controls the HS position. The horizontal sync position defines the horizontal sync delay (Front Porch) and the horizontal scan delay (Back Porch). When the programmed value of this register is increased, the display on the CRT screen is shifted to the left. When the programmed value is decreased the display is shifted to the right. Any 8-bit number may be programmed as long as the sum of the contents of R1, R2, and the lower four bits of R3 are less than the contents of R0.

Sync Width Register (R3) — This 8-bit register determines the width of the vertical sync (VS) pulse and the horizontal sync (HS) pulse. Programming the upper four bits for 1-to-15 will select VS pulse widths from 1-to-15 scan-line times. Programming the upper four bits as zeros will select a VS pulse width of 16 scan line times. The HS pulse width may be programmed from 1-to-15 character clock periods thus allowing compatibility with the HS pulse width specifications of many different monitors. If zeros are written into the lower four bits of this register, then no HS is provided.

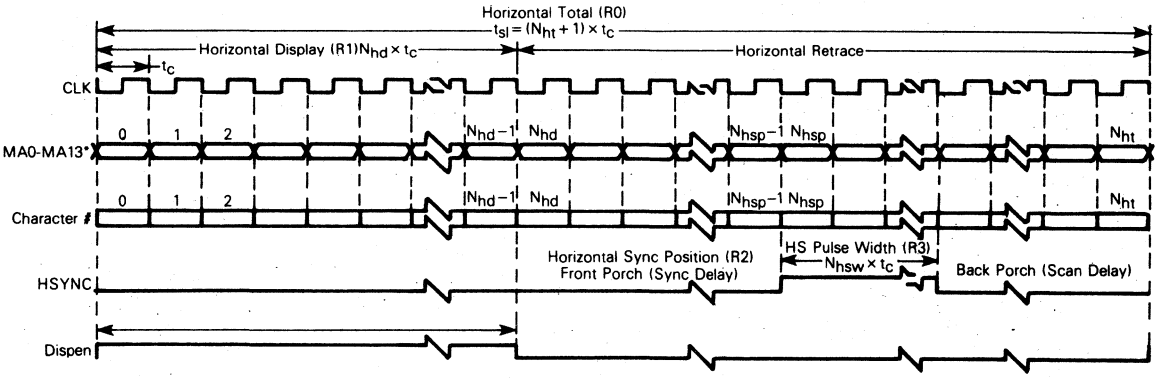
Horizontal Timing Summary (Figure 12) — The difference between R0 and R1 is the horizontal blanking interval. This interval in the horizontal scan period allows the beam to return (retrace) to the left side of the screen. The retrace time is determined by the monitor's horizontal scan components. Retrace time is less than the horizontal blanking interval. A good rule of thumb is to make the horizontal blanking about 20% of the total horizontal scanning period for a CRT. In inexpensive TV receivers, the beam overscans the display screen so that aging of parts does not result in underscanning. Because of this, the retrace time should be about 1/3 the horizontal scanning period. The horizontal sync delay, HS pulse width and horizontal scan delay are typically programmed with 1:2:2 ratio.

FIGURE 11 — ILLUSTRATION OF THE CRT SCREEN FORMAT



Note 1: Timing values are described in Table 8.

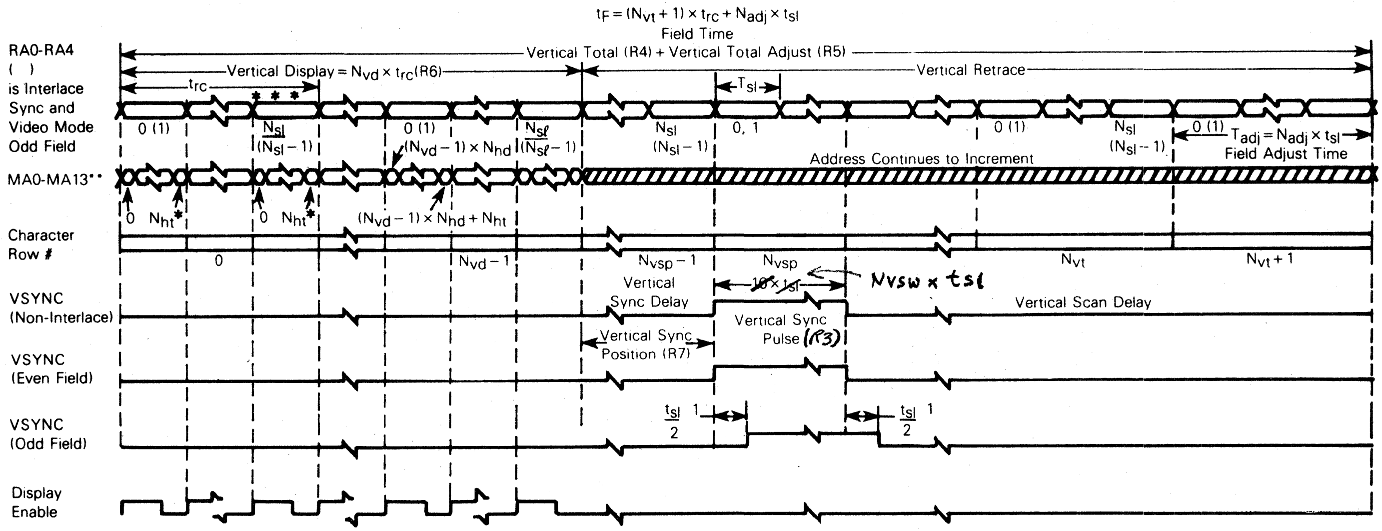
FIGURE 12 — CRTC HORIZONTAL TIMING



*Timing is shown for first displayed scan row only. See Chart in Figure 16 for other rows. The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

Note 1: Timing values are described in Table 5.

FIGURE 13 — CRTC VERTICAL TIMING



*N_{ht} must be an odd number for both interlace modes.
**Initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.
***N_{sl} must be an odd number for Interlace Sync and Video Mode.

NOTES:
1. Refer to Figure 7 — The Odd Field is offset ½ horizontal scan time.
2. Timing values are described in Table 5.

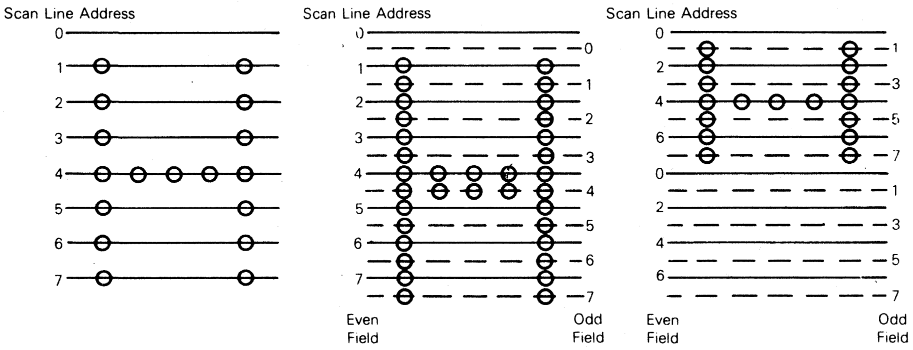
TABLE 2 — INTERLACE MODE REGISTER

Bit 1	Bit 0	Mode
0	0	Normal Sync Mode (Non-Interlace)
1	0	
0	1	Interlace Sync Mode
1	1	Interlace Sync and Video Mode

TABLE 3 — CURSOR START REGISTER

Bit 6	Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate

FIGURE 14 — INTERLACE CONTROL



a) Normal Sync

b) Interlace Sync

c) Interlace Sync and Video

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) — The frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7-bit Vertical Total Register (R4). The fraction of character line times is programmed in the 5-bit Vertical Total Adjust Register (R5) as a number of scan line times.

Vertical Displayed Register (R6) — This 7-bit register specifies the number of displayed character rows on the CRT screen, and is programmed in character row times. Any number smaller than the contents of R4 may be programmed into R6.

Vertical Sync Position (R7) — This 7-bit register controls the position of vertical sync with respect to the reference. It is programmed in character row times. The value programmed in the register is one less than the number of computed character line times. When the programmed value of this register is increased, the display position of the CRT screen is shifted up. When the programmed value is decreased the display position is shifted down. Any number equal to or less than the vertical total (R4) may be used.

Interlace Mode and Skew Register (R8) — This 6-bit register controls the interlace modes and allows a programmable delay of zero to two character clock times for the DE (display enable) and Cursor outputs. Table 2 shows the interlace modes available to the user. These modes are selected using the two low order bits of this 6-bit register.

Table 4 describes operation of the Cursor and DE skew bits. Cursor skew is controlled by bits 6 and 7 of R8 while DE skew is controlled by bits 4 and 5.

In the normal sync mode (non-interlace) only one field is available as shown in Figures 6 and 14a. Each scan line is refreshed at the VS frequency (e.g., 50 or 60 Hz).

Two interlace modes are available as shown in Figures 7, 14b, and 14c. The frame time is divided between even and odd alternating fields. The horizontal and vertical timing relationship (VS delayed by 1/2 scan line time) results in the displacement of scan lines in the odd field with respect to the even field.

In the Interlace Sync mode the same information is painted in both fields as shown in Figure 14b. This is a useful mode for filling in a character to enhance readability.

In the Interlace Sync and Video mode alternating lines of the character are displayed in the even field and the odd field. This effectively doubles the number of characters that may be displayed on a CRT monitor of a given bandwidth.

Care must be taken when using either interlace mode to avoid an apparent flicker effect. This flicker effect is due to the doubling of the refresh period for all scan lines since each field is displayed alternately. Flicker may be minimized with proper monitor design (e.g., longer persistence phosphors).

In addition, there are restrictions on the programming of the CRTC registers for interlace operation:

a. The Horizontal Total Register value, R0, must be odd (i.e., an even number of character times).

b. For the Interlace Sync and Video mode only, the Vertical Displayed Register (R6) must be even. The programmed number, Nvd, must be ½ the actual number required.

TABLE 4 — CURSOR AND DE SKEW CONTROL

Value	Skew
00	No Character Skew
01	One Character Skew
10	Two Character Skew
11	Not Available

Maximum Scan Line Address Register (R9) — This 5-bit register determines the number of scan lines per character row including the spacing thus controlling operation of the Row Address counter. The programmed value is a maximum address and is one less than the number of scan lines.

Cursor Start Register (R10) and Cursor End Register (R11) — These registers allow a cursor of up to 32 scan lines in height to be placed on any scan line of the character block as shown in Figure 15. R10 is a 7-bit register used to define the start scan line and blink rate for the cursor. Bits 5 and 6 of the Cursor Start Address Register control the cursor operation as shown in Table 4. Non-display, display and two blink modes (16 times or 32 times the field period) are available. R11 is a 5-bit register which defines the last scan line of the cursor.

When an external blink feature on characters is required, it may be necessary to perform cursor blink externally so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for a blinking cursor and externally inverting the video signal with an exclusive-OR gate.

PROGRAMMABLE REGISTERS

The five programmable registers allow the MPU to position the cursor anywhere on the screen and allow the start address to be modified.

The Address Register is a five-bit write-only register used as an "indirect" or "pointer" register. Its contents are the address of one of the other 18 registers. When both RS and $\overline{\text{CS}}$ are low, the Address Register is selected. When $\overline{\text{CS}}$ is low and RS is high, the register pointed to by the Address Register is selected.

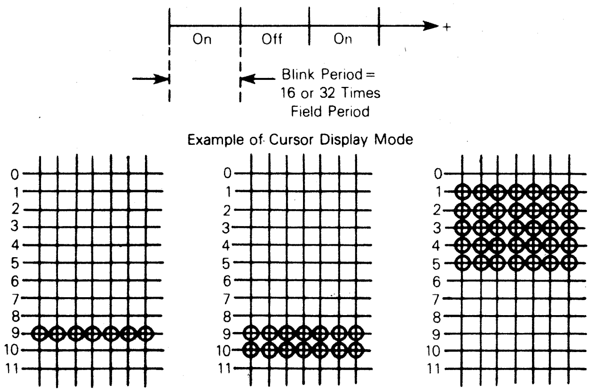
Start Address Register (R12-H, R13-L) — This 14-bit write-only register pair controls the first address output by the CRTC after vertical blanking. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. The start address register determines which portion of the refresh RAM is displayed on the CRT screen. Hardware scrolling by character, line or page may be accomplished by modifying the contents of this register.

Cursor Register (R14-H, R15-L) — This 14-bit write-only register pair is programmed to position the cursor anywhere in the refresh RAM area thus allowing hardware paging and scrolling through memory without loss of the original cursor position. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register.

CRTC INITIALIZATION

Registers R12-R15 must be initialized after the system is powered up. The processor will normally load the CRTC register file from a firmware table. Figure 16 shows an M6800 program which could be used to program the CRT Controller.

FIGURE 15 — CURSOR CONTROL



Cursor Start Adr. = 9
Cursor End Adr. = 9

Cursor Start Adr. = 9
Cursor End Adr. = 10

Cursor Start Adr. = 1
Cursor End Adr. = 5

ADDITIONAL CRTC APPLICATIONS

The foremost system function which may be performed by the CRTC controller is the refreshing of dynamic RAM. This is quite simple as the refresh addresses continually run.

Both the VS and the HS outputs may be used as a real time clock. Once programmed, the CRTC will provide a stable reference frequency.

SELECTING MASK PROGRAMMED REGISTER VALUES

A prototype system may be developed using the MC6845 CRTC. This will allow register values to be modified as required to meet system specifications. The worksheet of Table 5 is extremely useful in computing proper register values for the MC6835. The program shown in Figure 16 may be expanded to properly load the calculated register values in the MC6845. Once the two sets of register values have been developed, fill out the ROM program worksheet of Figure 19.

To order a custom programmed MC6835, contact your local field service office, local sales person or your local Motorola representative. A manufacturing mask will be developed for the data entered in Figure 19.

FIGURE 16 — M6800 PROGRAM FOR CRTC INITIALIZATION

```
PAGE  001  CRTCINIT.SA:1  MC6835 CRTC initialization program

00001                         NAM    MC6835
00002                         TTL    CRTC initialization program
00003                         OPT    G,S,LLE=85 print FCB'x, FDB's & XREF table
00004                  *********************************************************
00005                  * Assign CRTC address
00006                  *
00007          9000  A CRTCAD EQU    $9000    Address Register
00008          9001  A CRTCRG EQU    CRTCAD+1 Data Register
00009                  *********************************************************
00010                  * Initialization Program
00011                  *
00012A 0000                   ORG    0        a place to start
00013A 0000 C6 0C    A        LDAB   $C       initialize pointer
00014A 0002 CE 1020  A        LDX    38RTTAB  table pointer
00015A 0005 F7 9000  A CRTC1  STAB   CRTCAD   load address register
00016A 0008 A6 00    A        LDAA   0,X      get register value from table
00017A 000A B7 9001  A        STAA   CRTCRG   program register
00018A 000D 08                INX             increment counter
00019A 000E 5C                INCB
00020A 000F D1 10    A        CMPB   $10      finished?
00021A 0011 26 F2 0005        BNE    CRTC1    no: take branch
00022A 0013 3F                SWI             yes: call monitor
00023                  *********************************************************
00024                  * CRTC register initialization table
00025                  *
00026A 1020                   ORG    $1020    start of table
00027A 1020    0080  A CRTTAB FDB    $0080    R12, R13 - Start Address
00028A 1022    0080  A        FDB    $0080    R14, R15 - Cursor Address
00029                         END
TOTAL ERRORS 00000--00000


CRTC1  0005  CRTCAD 9000  CRTCRG 9001  CRTTAB 1020
```

TABLE 5 — CRTC FORMAT WORKSHEET

Display Format Worksheet

1. Displayed Characters per Row		________	Char.
2. Displayed Character Rows per Screen		________	Rows
3. Character Matrix	a. Columns	________	Columns
	b. Rows	________	Rows
4. Character Block	a. Columns	________	Columns
	b. Rows	________	Rows
5. Frame Refresh Rate		________	Hz
6. Horizontal Oscillator Frequency		________	Hz
7. Active Scan Lines (Line 2 × Line 4b)		________	Lines
8. Total Scan Lines (Line 6 ÷ Line 5)		________	Lines
9. Total Rows Per Screen (Line 8 ÷ Line 4b)		____ Rows	and ____ Lines
10. Vertical Sync Delay (Char. Rows)		________	Rows
11. Vertical Sync Width (Scan Lines (16)) *		________	Lines
12. Horizontal Sync Delay (Character Times)		________	Char. Times
13. Horizontal Sync Width (Character Times)		________	Char. Times
14. Horizontal Scan Delay (Character Times)		________	Char. Times
15. Total Character Times (Line 1 + 12 + 13 + 14)		________	Char. Times
16. Character Rate (Line 6 × 15)		________	Hz
17. Dot Clock Rate (Line 4a × 16)		________	Hz

CRTC Registers

		Decimal	Hex
R0	Horizontal Total (Line 15 – 1)	________	________
R1	Horizontal Displayed (Line 1)	________	________
R2	Horizontal Sync Position (Line 1 + Line 12)	________	________
R3	Horizontal Sync Width (Line 13) *	________	________
R4	Vertical Total (Line 9 – 1)	________	________
R5	Vertical Adjust (Line 9 Lines)	________	________
R6	Vertical Displayed (Line 2)	________	________
R7	Vertical Sync Position (Line 2 + Line 10)	________	________
R8	Interlace (00 Normal, 01 Interlace ** 03 Interlace, and Video)		________
R9	Max Scan Line Add (Line 4b – 1)	________	________
R10	Cursor Start	________	________
R11	Cursor End	________	________
R12, R13	Start Address (H and L)	________	________
R14, R15	Cursor (H and L)		________

* NOTE 1: R3 is also programmed with Vertical Sync.Width

** NOTE 2: R8 is also programmed with Cursor and De Skew (See Table 4)

TABLE 6 — WORKSHEET FOR 80 × 24 FORMAT

Display Format Worksheet

1.	Displayed Characters per Row		80	Char.
2.	Displayed Character Rows per Screen		24	Rows
3.	Character Matrix	a. Columns	7	Columns
		b. Rows	9	Rows
4.	Character Block	a. Columns	9	Columns
		b. Rows	11	Rows
5.	Frame Refresh Rate		60	Hz
6.	Horizontal Oscillator Frequency		18,600	Hz
7.	Active Scan Lines (Line 2 × Line 4b)		264	Lines
8.	Total Scan Lines (Line 6 ÷ Line 5)		310	Lines
9.	Total Rows Per Screen (Line 8 ÷ Line 4b)		28	Rows and 2 Lines
10.	Vertical Sync Delay (Char Rows)			Rows
11.	Vertical Sync Width (Scan Lines (16)) *		16	Lines
12.	Horizontal Sync Delay (Character Times)		6	Char. Times
13.	Horizontal Sync Width (Character Times)		9	Char. Times
14.	Horizontal Scan Delay (Character Times)		7	Char. Times
15.	Total Character Times (Line 1 + 12 + 13 + 14)		102	Char. Times
16.	Character Rate (Line 6 times 15)		1.8972 M	MHz
17.	Dot Clock Rate (Line 4a times 16)		17.075 M	MHz

CRTC Registers

		Decimal	Hex
R0	Horizontal Total (Line 15 minus 1)	101	65
R1	Horizontal Displayed (Line 1)	80	50
R2	Horizontal Sync Position (Line 1 + Line 12)	86	56
R3	Horizontal Sync Width (Line 13) *	9	9
R4	Vertical Total (Line 9 minus 1)	24	18
R5	Vertical Adjust (Line 9 Lines)	10	0A
R6	Vertical Displayed (Line 2)	24	18
R7	Vertical Sync Position (Line 2 + Line 10)	24	18
R8	Interlace (00 Normal, 01 Interlace, 03 Interlace, and Video) * *		0
R9	Max Scan Line Add (Line 4b minus 1)	11	B
R10	Cursor Start	0	0
R11	Cursor End	11	B
R12, R13	Start Address (H and L)	128	00 80
R14, R15	Cursor (H and L)	128	00 80

* NOTE 1: R3 is also programmed with Vertical Sync.Width

** NOTE 2: R8 is also programmed with Cursor and De Skew (See Table 4)

OPERATION OF THE CRTC

Timing of the CRT Interface Signals — Timing charts of CRT interface signals are illustrated in this section with the aid of programmed example of the CRTC. When values listed in Table 7 are programmed into CRTC control registers, the device provides the outputs as shown in the Timing Diagrams (Figures 12, 13, 17, and 18). The screen format of this example is shown in Figure 11. Figure 18 is an illustration of the relation between Refresh Memory Address (MA0-MA13), Raster Address (RA0-RA4) and the position on the screen. In this example, the start address is assumed to be "0".

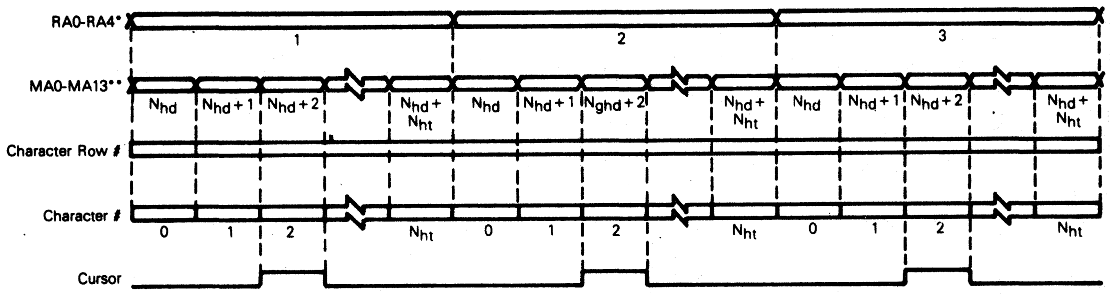
TABLE 7 — VALUES PROGRAMMED INTO CRTC REGISTERS

Register Number	Register Name	Value	Programmed Value
R0	H. Total	$N_{ht}+1$	N_{ht}
R1	H. Displayed	N_{hd}	N_{hd}
R2	H. Sync Position	N_{hsp}	N_{hsp}
R3	Sync Width	*	* *
R4	V. Total	$N_{vt}+1$	N_{vt}
R5	V. Scan Line Adjust	N_{adj}	N_{adj}
R6	V. Displayed	N_{vd}	N_{vd}
R7	V. Sync Position	N_{vsp}	N_{vsp}
R8	Interlace Mode		
R9	Max. Scan Line Address	N_{sl}	N_{sl}
R10	Cursor Start		
R11	Cursor End		
R12	Start Address (H)	0	
R13	Start Address (L)	0	
R14	Cursor (H)		
R15	Cursor (L)		

* N_{vsw}, N_{hsw}

** N_{vsw}, N_{hsw}

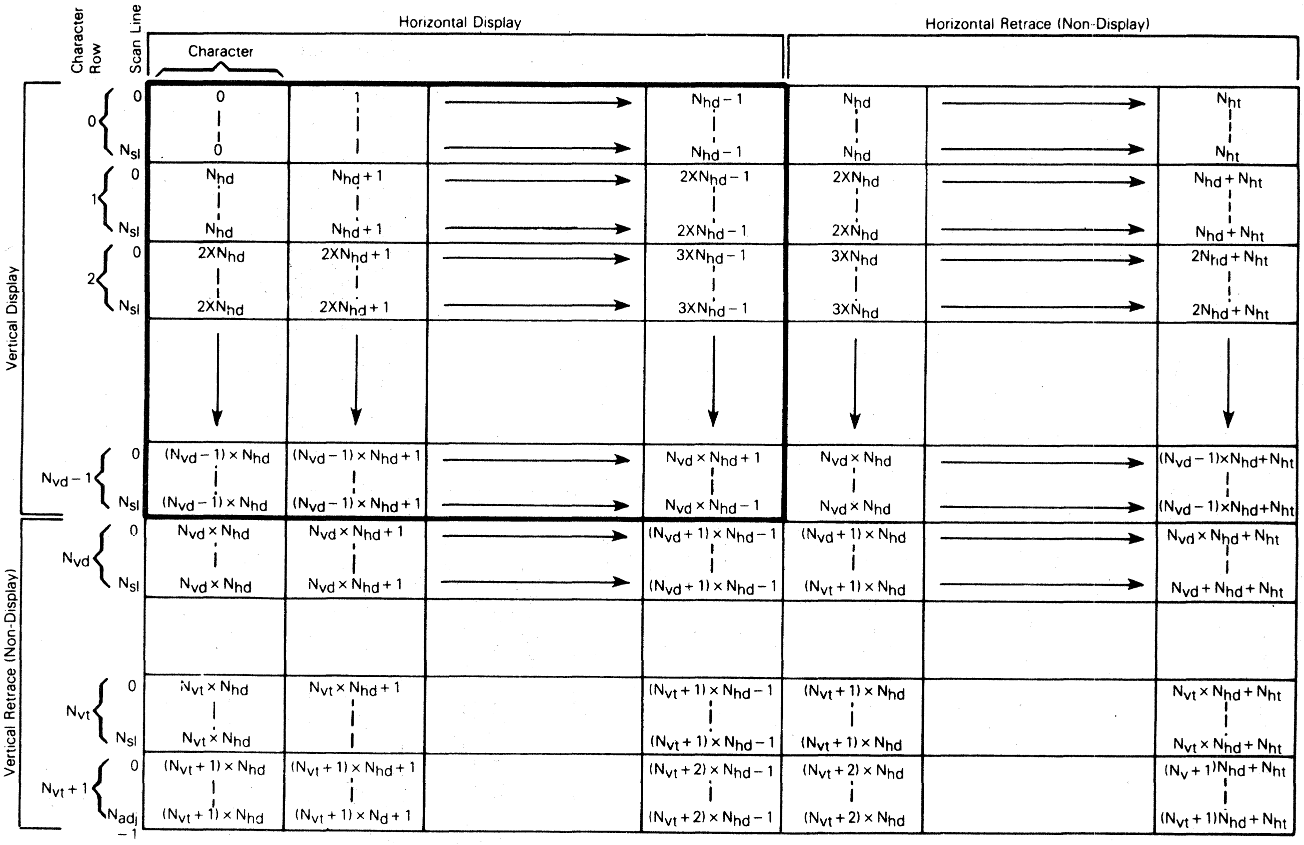
FIGURE 17 — CURSOR TIMING



*Timing is shown for non-interlace and interlace sync modes.
Example shown has cursor programmed as:
Cursor Register = $N_{hd} + 2$
Cursor Start = 1
Cursor End = 3
**The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

Note 1: Timing values are described in Table 8.

FIGURE 18 — REFRESH MEMORY ADDRESSING (MA0-MA13) STATE CHART



NOTE 1: The initial MA is determined by the contents of start address register, R12/R13. Timing is shown for R12/R13=0. Only Non-Interlace and Interlace Sync Modes are shown.

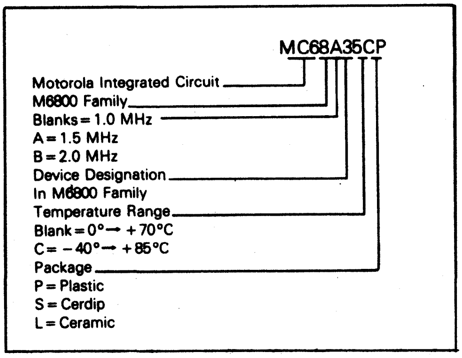
FIGURE 19 — ROM PROGRAM WORKSHEET

The value in each register of the MC6845 should be entered without any modifications. Motorola will take care of translating into the appropriate format.

☐ All numbers are in decimal. ☐ All numbers are in hex.

	ROM Program Zero (PROG = 0)	ROM Program One (PROG = 1)
R0		
R1		
R2		
R3		
R4		
R5		
R6		
R7		
R8		
R9		
R10		
R11		

ORDERING INFORMATION



Speed	Device	Temperature Range
1.0 MHz	MC6835P,L,S MC6835CP,CL,CS	0 to 70°C −40 to +85°C
1.5 MHz	MC68A35P,L,S MC68A35CP,CL,CS	0 to +70°C −40 to +85°C
2.0 MHz	MC68B35P,L,S	0 to +70°C

MOTOROLA

MC68A39
(1.5 MHz)
MC68B39
(2.0 MHz)

Advance Information

FLOATING-POINT ROM

The MC6839 standard product ROM provides floating point capability for the MC6809 or MC6809E MPU. The MC6839 implements the entire *IEEE Proposed Standard for Binary Floating Point Arithmetic Draft 8.0,* providing a simple, economical and reliable solution to a wide variety of numerical applications. The single- and double-precision formats provide results which are bit-for-bit reproducible across all Draft 6.0 implementations, while the extended format provides the extra precision needed for the intermediate results of long calculations, in particular the implementation of transcendental functions and interest calculations. All applications benefit from extensive error-checking and well-defined responses to exceptions, which are strengths of the IEEE proposed standard.

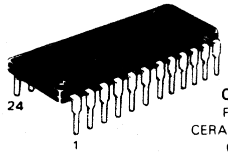
The MC6839 takes full advantage of the advanced architectural features of the MC6809 microprocessor. It is position-independent and re-entrant, facilitating its use in real-time, multi-tasking systems.

- Totally Position Independent
  - Operates in any Contiguous 8K Block of Memory
- Re-Entrant
  - No Use of Absolute RAM
  - All Memory References are made Relative to the Stack Pointer
- Flexible User Interface
  - Operands are Passed to the Package by One of Two Methods
    1) Machine Registers are used as Pointers to the Operands
    2) The Operands are Pushed onto the Hardware Stack
  - The Latter Method Facilitates the use of the MC6839 in High-Level Language Implementations
- Easy to Use Two/Three Address Architecture
  - The User Specifies Addresses of Operands and Result and Need Not be Concerned with any Internal Registers or Intermediate Results
- A Complete Implementation of the Proposed IEEE Standard Draft 6.0
  - Includes All Precisions, Modes, and Operations Required or Suggested by the Standard
  - Single, Double, and Extended Formats
  - Includes the Following Operations:
    - Add
    - Subtract
    - Multiply
    - Divide
    - Remainder
    - Square Root
    - Integer Part
    - Absolute Value
    - Negate
    - Predicate Compares
    - Condition Code Compares
    - Convert Integer ↔ Floating Point
    - Convert Binary Floating Point ↔ Decimal String

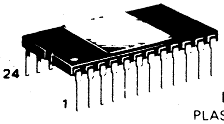
MOS

(N-CHANNEL, SILICON-GATE)

FLOATING-POINT READ-ONLY MEMORY

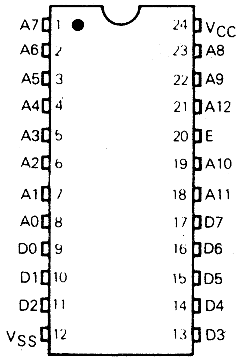


C SUFFIX
FRIT-SEAL
CERAMIC PACKAGE
CASE 716

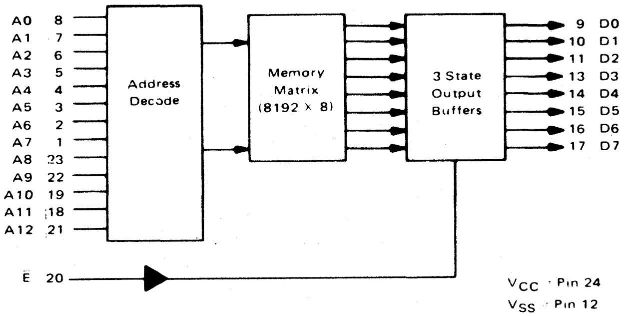


P SUFFIX
PLASTIC PACKAGE
CASE 709

PIN ASSIGNMENT



BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.5 to +7.0	V
Input Voltage	V_{in}	−0.5 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

CAPACITANCE

($f = 1.0$ MHz, $T_A = 25°C$, periodically sampled rather than 100% tested)

Characteristic	Symbol	Max	Unit
Input Capacitance	C_{in}	8	pF
Output Capacitance	C_{out}	15	pF

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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

DC OPERATING CONDITIONS AND CHARACTERISTICS

(Full operating voltage and temperature range unless otherwise noted)

RECOMMENDED DC OPERATING CONDITIONS

Parameter	Symbol	Min	Nom	Max	Unit
Supply Voltage (V_{CC} must be applied at least 100 μs before proper device operation is achieved)	V_{CC}	4.5	5.0	5.5	V
Input High Voltage	V_{IH}	2.0	–	5.5	V
Input Low Voltage	V_{IL}	−0.5	–	0.8	V

DC CHARACTERISTICS

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current ($V_{in} = 0$ to 5.5 V)	I_{in}	−10	–	10	μA
Output High Voltage ($I_{OH} = -220$ μA)	V_{OH}	2.4	–	–	V
Output Low Voltage ($I_{OL} = 3.2$ mA)	V_{OL}	–	–	0.4	V
Output Leakage Current (Three-State) ($E = 2.0$ V, $V_{out} = 0$ V to 5.5 V)	I_{LO}	−10	–	10	μA
Supply Current – Active* (Minimum Cycle Rate)	I_{CC}	–	25	40	mA
Supply Current – Standby ($E = V_{IH}$)	I_{SB}	–	7	10	mA

*Current is proportional to cycle rate.

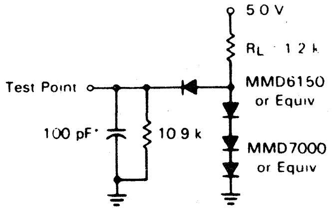
AC OPERATING CONDITIONS AND CHARACTERISTICS

(Read Cycle)

RECOMMENDED AC OPERATING CONDITIONS ($T_A = 0$ to 70°C, $V_{CC} = 5.0$ V ± 10%. All timing with $t_r = t_f = 20$ ns, load of Figure 1)

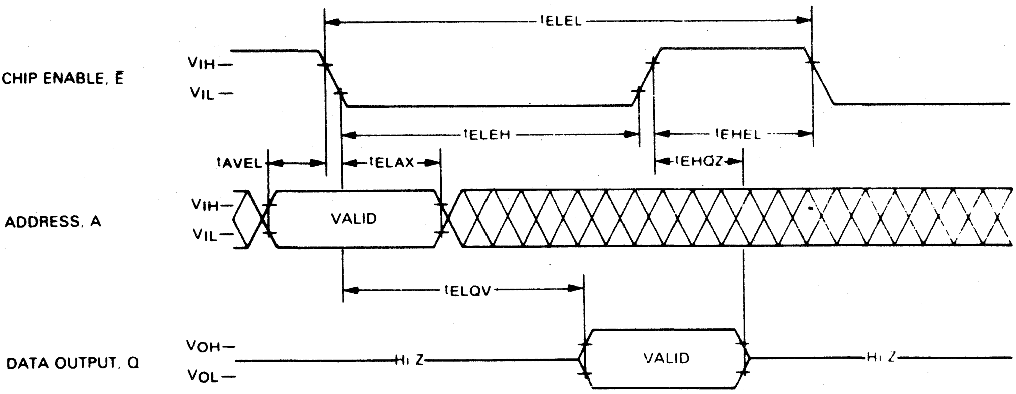
Parameter	Symbol	MC68A39		MC68B39		Unit
		Min	Max	Min	Max	
Chip Enable Low to Chip Enable Low of Next Cycle (Cycle Time)	t_{ELEL}	450	–	375	–	ns
Chip Enable Low to Chip Enable High	t_{ELEH}	300	–	250	–	ns
Chip Enable Low to Output Valid (Access)	t_{ELQV}	–	300	–	250	ns
Chip Enable High to Output High Z (Off Time)	t_{EHQZ}	–	75	–	60	ns
Chip Enable Low to Address Don't Care (Hold)	t_{ELAX}	75	–	60	–	ns
Address Valid to Chip Enable Low (Address Setup)	t_{AVEL}	0	–	0	–	ns
Chip Enable Precharge Time	t_{EHEL}	110	–	70	–	ns

FIGURE 1 – AC TEST LOAD



*Includes Jig Capacitance

FIGURE 2 – TIMING DIAGRAM



INTRODUCTION

Since the earliest days of computers it has has been obvious that no computer was capable of doing all desirable mathematical operations in binary integer arithmetic. To meet the needs of those applications requiring the manipulation of real numbers, floating point (FP) evolved and became widely used. Unfortunately, each computer manufacturer created his own floating point (FP) representation and the ensuing wide variation in formats, accuracy, and exception handling almost guarantees that a program executed on one computer will get different results if executed on another computer.

Meanwhile, research has been completed which formulates an optional binary floating point representation. Unfortunately, the existing manufacturers have far too much money invested in software and hardware to incur the costs of conversion to a new standard. Powerful microprocessors, on the other hand, were in their infancy and the floating point experts saw the opportunity to standardize a floating point format for microprocessors. The IEEE appointed a committee to address the standard and their work resulted in the *IEEE Proposed Standard for Binary Floating Point Arithmetic Draft 8.0.*

The MC6839 represents a complete implementation of the IEEE proposed standard. Since hardware implementations of floating point (FP) are always several orders of magnitude faster (and more expensive) than software implementations, the MC6839 substitutes increased functionality for speed. Therefore, the MC6839 supports all precisions, modes, and operations required or suggested by the IEEE proposed standard.

From its very inception, the M6809 microprocessor was designed to support a concept of ROMable software by an improved instruction set and addressing modes. It was felt that the only way to reduce the escalating cost of software was for the silicon manufacturer to supply software on silicon. Since the manufacturer can amortize the cost of developing the software over a very large volume, the cost of this software, above normal masked ROM costs, will be low. Also, to be useful in many diverse systems, the ROM must be position-independent and re-entrant.

The intent of this Advance Information (data) Sheet is to provide the reader with enough information to make an intelligent decision as to whether the MC6839 is applicable to his system. The intent is not to provide all the details necessary to interface or program the MC6839; a users manual is available for that purpose. A familiarity with the MC6809 instruction set is assumed in this document.

PHYSICAL CHARACTERISTICS

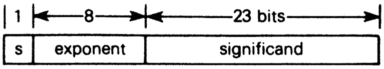
The MC6839 is housed in one 24-pin 8K-by-8 mask programmable ROM: the MCM68364. This ROM uses a single 5 V power supply and is available with access times of 250 or 350 ns. The MC6839 is designed to be used in MC6809 or MC6809E systems with up to 2 MHz internal clocks. Full device characteristics can be found at the front of this data sheet.

FLOATING POINT FORMATS

The MC6839 supports the three precisions suggested by the IEEE Proposed Floating Point Standard: single, double, and extended. The values occupy 32, 64, and 80 bits (4, 8, and 10 bytes) respectively in the users memory. The formats of the three precisions are described in the following paragraphs.

SINGLE FORMAT

All single precision numbers are represented in four bytes as:



The exponent is biased by + 127. That is exponent of: 2^0 is 127, 2^2 is 129, and 2^{-2} is 125. The significand is stored in sign magnitude rather than twos complement form. The equation for the single form representation is:

$x = (-1)^s \times 2^{(exp - 127)} \times (1.\ significand)$

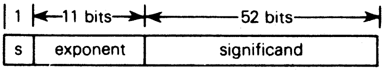
s = sign of the significand
exp = biased exponent
significand = bit string of length 23 encoding the significant bits of the number that follow the binary point, yielding a 24-bit significant digit field for the number that always begins "1______."

Examples:

$+1.0 =$	$1.0 \times 2^0 =$	\$3F	80	00	00
$+3.0 =$	$1.5 \times 2^1 =$	\$40	40	00	00
$-1.0 =$	$-1.0 \times 2^0 =$	\$BF	80	00	00

DOUBLE FORMAT

All double precision numbers are represented by an 8-byte string as:



For double formats the exponent is biased by + 1023. The rest of the interpretation is the same as for single format. The equation for double format is:

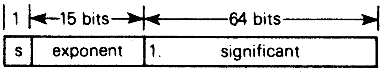
$x = (-1)^s \times 2^{(exp - 1023)} \times (1.\ significand)$

Examples:

$7.0 =$	$1.75 = 2^2 =$	\$40	1C	00	00	00	00	00	00
$-30.0 =$	$-1.875 \times 2^4 =$	\$C0	3E	00	00	00	00	00	00
$0.25 =$	$1.0 \times 2^2 =$	\$3F	D0	00	00	00	00	00	00

EXTENDED FORMAT

Single- and double-formats should be used to represent the bulk of floating point (FP) numbers in the user's system (e.g., storage of arrays). Extended should only be used for intermediate calculations such as occur in the evaluation of a complex expression. In fact, extended may not be used at all by most users, but since it is required internally, it is optionally provided. Extended numbers are represented in 10 bytes as:



A notable difference between this format and single and double is the 1.0 is explicitly present in the significand and the exponent contains no bias and is in twos complement form. The equation for double extended is:

$x = (-1)^s \times 2^{exp} \times \text{significand}$

where the significand contains the explicit 1.0.

Examples:

$0.5 =$	$1.0 \times 2^{-1} =$	$7F	FF	80	00	00	00	00	00	00	00
$-1.0 =$	-1.0×2^{0}	=	$80	00	80	00	00	00	00	00	00	00
$384.0 =$	1.5×2^{8}	=	$00	08	C0	00	00	00	00	00	00	00

BCD STRINGS

A BCD string is the input to the BCD-to-Floating-Point conversion operation and the output of the Floating-Point-to-BCD conversion operation. All BCD strings have the following format:

0	1	5	6 24	25
se	4 digit BCD exponent	sf	19 digit BCD fraction	p

se = sign of the exponent. $00 = plus, $0F = minus. (one byte)
sf = sign of the fraction. $00 = plus, $0F = minus. (one byte)
p = number of fraction digits to the right of the decimal point. (one byte)

All BCD digits are *unpacked* and right justified in each byte:

7	0
0 0 0 0	0-9

The byte ordering of the fraction and exponent is consistent with all Motorola processors in that the most-significant BCD digit is in the lowest memory address.

Examples:

$2.0 = 2.0 \times 10^{0}$ (p = 0)

Address	Data					
0000	00					(se = +)
0001	00	00	00	00		(exponent = 0)
0005	00					(sf = +)
0006	00	00	00	00	00	(fraction = 2)
000B	00	00	00	00	00	
0010	00	00	00	00	00	
0015	00	00	00	02		
0019	00					(p = 0)

or $2.0 = 20{,}000 \times 10^{-4}$ (p = 0)

Address	Data					
0000	0F					(se = −)
0001	00	00	00	04		(exponent = 4)
0005	00					(sf = +)
0006	00	00	00	00	00	(fraction = 20000)
000B	00	00	00	00	00	
0010	00	00	00	00	02	
0015	00	00	00	00		
0019	00					(p = 0)

(The above might be the output of a Floating-Point-to-BCD with k = 5)

or $2.0 = 2.0 \times 10^{0}$ (p = 10)

Address	Data					
0000	00					(se = +)
0001	00	00	00	00		(exponent = 0)
0005	00					(sf = +)
0006	00	00	00	00	00	(fraction = 20000000000)
000B	00	00	00	02	00	
0010	00	00	00	00	00	
0015	00	00	00	00		
0019	0A					(p = 10)

INTEGERS

Two sizes of integers are supported; short and double. Short integers are 16 bits long and double integers are 32 bits long. The byte ordering is consistent with all Motorola processors in that the most-significant bits are in the lowest address.

SPECIAL VALUES

No derivable floating point format can represent the infinite number of possible real numbers, so it is very useful if some special numbers are recognized by a floating point package. These numbers are: +0, −0, + infinity, − infinity, very small (almost zero) numbers, and in some cases unnormalized numbers. Also, it is convenient to have a sepcial format which indicates that the contents of memory do not contain a valid floating point number. This "not a number" might occur if a variable is defined in a HLL and is used before it is initialized with a value. The most positive and negative exponents of each format are reserved to represent these special vaues.

The detailed description of these special values is given in a later section.

ARCHITECTURE

All floating point operations are of the "two address" or "three address" variety; all the user need supply are the addresses of the operand(s) and the result. The package looks for operands at the specified location(s) and delivers the result to the specified destination. For example,

Arg1 + Arg2 → Result
<source> <source> <destination>

Intermediate results are never presented to the user; therefore, there are no internal "registers" to be concerned about, keeping the interface as simple as possible. The end result is ease of use.

There is a user defined floating point control block (fpcb) that defines the mode of the package. This control block is much like the control blocks frequently used to define I/O or operating system operations. The fpcb is discussed in detail in a later section.

SUPPORTED OPERATIONS

The MC6839 supports the following operations. On any particular call to the floating point ROM a 1-byte opcode which immediately follows the LBSR instruction chooses the desired operation. Below are short descriptions of the functions implemented in the MC6839 along with suggested menmonics. A table containing the opcodes and calling sequences for these functions is presented at the end of this data sheet.

ASCII Mnemonic	Description
FADD	Add arg1 to arg2 and store the result.
FSUB	Subtract arg2 from arg1 and store the result.
FMUL	Multiply arg1 times arg2 and store the result.
FDIV	Divide arg1 by arg2 and store the result.
FREM	Take the remainder of arg1 divided by arg2 and store the result. The remainder is biased to lie in the range −arg2/2 <remainder< +arg2/2, instead of the usual range of 0≤remainder<arg2. This bias makes the function more useful in the implementation of trigonometric and other functions.
FCMP	Compare arg1 with arg2 and set the condition codes to the result of the compare. Arg1 and arg2 can be of different precisions.
FTCMP	Compare arg1 with arg2 and set the condition codes to the result of the compare. In addition, trap if an unordered exception occurs regardless of the state of the UNOR (unordered) bit in the trap enable byte of the fpcb.
FPCMP	A predicate compare; this means compare arg1 with arg2 and affirm or disaffirm the input predicate (e.g., 'is arg1 = arg2' or 'is arg1 < arg2').
FTPCMP	A trapping predicate compare; same as the predicate compare except trap on an unordered exception regardless of the state of the UNOR (unordered) bit in the trap enable byte of the fpcb.
FSQRT	Returns the square root of arg2 in the result.
FINT	Returns the interger part of arg2 in the result. The result is still a floating point number. For example, the integer part of 3.14159 is 3.00000.
FFIXS	Convert arg2 to a short (16-bit) binary integer.
FFIXD	Convert arg2 to a long (32-bit) binary integer.
FFLTS	Convert a short binary integer to a floating point result.
FFLTD	Convert a long binary integer to a floating point result.
BINDEC	Convert a binary floating point value to a BCD decimal string.
DECBIN	Convert a BCD decimal string to a binary floating point result.
FABS	Return the absolute value of arg2 in the result.
FNEG	Return the negative of arg2 in the result.
FMOV	Move (or convert) arg1→arg2. This function is useful for changing precisions (e.g., single to double) with full exception checking for possible overflow or underflow.

All routines, except FMOV and the compares, accept arguments of the same precision and generate a result with the same precision. For moves and compares the sizes of the arguments are passed to the package in a parameter word.

Details of each operation can be found in the *MC6839 Users Manual*.

MODES OF OPERATION

In addition to supporting a wide range of precisions and operations, the MC6839 supports all modes required or suggested by the IEEE Proposed Floating Point Standard. These include rounding modes, infinity closure modes, and exception handling modes. The various modes are selected by bits in the floating point control block (fpcb) that resides in user memory. Thus, each user or task can have a unique set of modes in effect for his calculations. The selection bits are defined in a later section on the fpcb.

ROUNDING MODES

Four rounding modes are suggested by the IEEE Proposed Floating Point Standard. They are:

1. Round to nearest (RN)
2. Round toward zero (RZ)
3. Round toward plus infinity (RP)
4. Round toward minus infinity (RN)

Round nearest will be used by most users because it provides the most accurate answers for most calculations. Round towards zero (truncate) is useful when the MC6839 implements real numbers in some high level languages that require truncation (i.e., FORTRAN). Round towards plus and minus infinity are used in interval arithmetic.

Normally a result is rounded to the precision of its destination. However, when the destination is Extended, the user can specify that the result significand be rounded to the precision of the basic format — single, double, or extended — of his choice, although the exponent range remains extended.

NO DOUBLE ROUNDING — The MC6839 is implemented such that no result will undergo more than one rounding error.

INFINITY CLOSURE MODES

The way in which infinity is handled in a floating point package may limit the number of applications in which the package can be used. To solve this problem, the proposed IEEE standard requires two types of infinity closures. A bit in the control byte of the Floating Point Control Block (fpcb) will select the type of closure that is in effect at any time.

AFFINE CLOSURE — In affine closure:

minus infinity < {every finite number} < plus infinity

Thus, infinity takes part in the real number system in the same manner as any other signed quantity.

PROJECTIVE CLOSURE — In projective closure:

infinity = minus infinity = plus infinity

and all comparisons between infinity and a floating point number involving order relations other than equal (=) or not equal (≠) are invalid operations. In projective closure the real number system can be thought of as a circle with zero at the top and infinity at the bottom.

NORMALIZE MODE

The purpose of the normalize mode is to prevent unnormalized results from being generated, which can otherwise happen. Such an unnormalized result arises when a denormalized operand is operated on such that its fraction remains not normalized but its exponent is no longer at its original minimum value. By transforming denormalized operands to normalized, internal form upon entering each operation, unnormalized results are guaranteed not to occur.

Thus, when operating in this mode the user can be assured that no attempt will be made to return an unnormalized value to a single or double destination. A bit in the control byte of the fpcb selects whether or not this mode is in effect. This mode is forced whenever the round mode is either round toward plus or minus infinity. Unnormalized numbers entering an operation are not affected by this mode, only denormalized ones are. Unnormalized and denormalized operands are discussed in a later section.

EXCEPTIONS

One of the greatest strengths of the IEEE Proposed Floating Point Standard is the regular and consistant handling of exceptions. Existing floating point implementations are quite varied in the way they handle exceptions, so the proposed IEEE standard has very carefully prescribed how exceptions must be handled and what constitutes an exception. Seven types of exceptions will be recognized by the MC6839. Only the first 5 are required by the proposed IEEE standard. They are:

1. Invalid Operation — a general exception that arises when an operation has gone so wrong that the program cannot return any reasonable result or fit the exception into any of the other more specific classes.
2. Underflow — arises when an operation generates a result that is too small to fit into the desired result precision.
3. Overflow — arises when an operation generates a result that is too large to fit into the desired result precision.
4. Division by Zero — arises when division by zero is attempted.
5. Inexact Result — arises when the result of an operation was not exact and therefore was rounded to the desired precision before being returned to the user.
6. Integer Overflow — arises when the binary integer result of a FIXS(D) operation cannot fit into 16(32) bits.
7. Comparison of Unordered Values — arises when one of the arguments to a compare operation is a "NAN" or an infinity in the projective closure mode. (See the Infinity and Not a Number paragraphs for further explanation of NANs and infinity.)

For each exception the caller will be given the option of specifying whether the package should: (1) trap to a user supplied trap routine to process the exception, or (2) deliver a default result specified by the proposed standard and proceed with execution. For most users the default result is adequate and the user need not write any trap handlers. Regardless of whether a trap is specified or not, a status bit will be set in the status byte of the fpcb and will remain set until cleared by the caller's program. Selection of whether to trap or to continue will be made by setting bits in the trap enable byte of the fpcb. For more details on the fpcb see the section on the Floating Point Control Block (fpcb).

If a trap is taken, the floating point package supplies a pointer that points to an area on the stack containing the following diagnostic information:

1. Event that caused the trap (overflow, etc.)
2. Where in the caller's program
3. Opcode
4. The input operands
5. The default result in internal format

In the event more than one exception occurs during the same operation, only one trap is invoked according to the following precedence.

1. Invalid Operation
2. Overflow
3. Underflow
4. Division by Zero
5. Unordered
6. Integer Overflow
7. Inexact Result

The user supplied trap routine (if any) will usually do 1 of 3 things:

1. Fix the result
2. Do nothing to the result and allow the floating point package to deliver the default value to the result.
3. Abort execution.

Sufficient detils on how to write a trap routine are furnished in the *MC6839 Users Manual.*

USER INTERFACE

There are two types of calls to the floating point package: register calls and stack calls. For register calls the user loads the machine registers with pointers (addresses) to the operand(s) and to the result; the call to the package is then performed. For stack calls the operand(s) is pushed on the stack and the call to the package is performed with the result replacing the operands on the stack after completion. The operand(s) must be pushed least-significant bytes first; this is consistent with the other Motorola architectures in that the most-significant byte resides in the lowest address. The two types of calls look like:

General form of a register call:

```
load registers
LBSR fpreg   register call
FCB  opcode
```

Example of a position-independent call to the add routine:

```
LEAU    arg1, pcr
LEAY    arg2, pcr
LEAX    fpcbptr, pcr      pointer to fpcb
TFR     x, d
LEAX    result, pcr
LBSR    fpreg
FCB     fadd
```

General form of a stack call:

```
push arguments
LBSR    fpstak       stack call
FCB     opcode
pull    result
```

Example of a stack call to the add routine:

```
push argument 1
push argument 2
push    fpcbptr      pointer to fpcb
LBSR    fpstak
FCB     fadd
pull    result
```

Details of the calling sequence for every type of operation can be found in the *MC6839 Users Manual;* a reference table of calling sequences and opcodes can be found at the end of this data sheet.

STACK REQUIREMENTS

When the MC6839 is called by the user, the package reserves local storage on the hardware stack. It then moves the input arguments from user memory to the local storage area and expands them into a convenient internal format. The operations use these "internal" numbers to arrive at an "internal" result which is then converted to the memory format of the result and returned to the user. For this reason, the user must insure that adequate memory exists on the hardware stack before calling the MC6839. The maximum stack sizes that any particular function will ever find necessary are:

register calls	150 bytes
stack calls	185 bytes

FLOATING POINT CONTROL BLOCK (fpcb)

The fpcb is a user-defined block that contains information needed by the floating point package. The fpcb is also used to pass status back to the caller or to invoke the trap routine. The fpcb must reside in the user RAM space to insure that the package can remain re-entrant. The caller of the floating point package must pass the address of the fpcb on each call. The format of the fpcb is:

control byte	0
trap enable byte	1
status byte	2
secondary status byte	3
address of trap routine	4 5

The meaning of the various bit fields within the fpcb are discussed in detail in the following paragraphs.

CONTROL BYTE – The control byte configures the floating point package for the caller's operation and is written by the user. Various fields in the byte set the precision, round, infinity closure, and normalize modes.

7	6	5	4	3	2	1	0
Precision			x	NRM	Round Mode		A/P

Bit 0 Closure (A/P) Bit
0 = projective closure
1 = affine closure

Bits 1-2 Round Mode
00 = round to nearest (RN)
01 = round to zero (RZ)
10 = round to plus infinity (RP)
11 = round to minus infinity (RM)

Bit 3 Normalize (NRM) Bit
1 = normalize denormalized numbers while in internal format before using. Precludes the creation of unnormalized numbers.
0 = do not normalize denormalized numbers (warning mode)

NOTE

If the rounding mode is RM or RP then normalize mode is forced. Unnormalized numbers are not affected by bit 3.

Bit 4 Undefined, reserved

Bits 5-7 Precision Mode
000 = Single
001 = Double
010 = Extended with no forced rounding of result
011 = Extended – force round result to single
100 = Extended – force round result to double
101 = Undefined, reserved
110 = Undefined, reserved
111 = Undefined, reserved

Note that if the control byte is set to zero by the user, all defaults in the IEEE Proposed Floating Point Standard will be selected.

STATUS BYTE

7	6	5	4	3	2	1	0
x	INX	IOV	UN	DZ	UNF	OVF	IOP

The bits in the status byte are set if any errors have occurred. Each bit of the status byte is a "sticky" bit in that it must be manually reset by the user. The FP package writes bits into the status byte but never clears existing bits. This is done so that a long calculation can be completed and the status need only be checked once at the end.

Bit 0	Invalid opertion (see secondary status)
Bit 1	Overflow
Bit 2	Underflow
Bit 3	Division by zero
Bit 4	Unordered
Bit 5	Integer overflow
Bit 6	Inexact result
Bit 7	Undefined, reserved

TRAP ENABLE BYTE

7	6	5	4	3	2	1	0
x	INX	IOV	UNOR	DZ	UNF	OVF	IOP

A "1" in any bit of the trap enable byte enables the FP package to trap if that error occurs. The bit definitions are the same as for the status byte. Note that if a trapping compare is executed and the result is unordered, then the unordered trap will be taken regardless of the state of the UNOR bit in the trap enable byte.

SECONDARY STATUS (SS)

7	6	5	4	3	2	1	0
x	x	x	Invalid Operation Type				

The FP package will write a status into this byte any time a new IOP occurs. As is the case with the status bytes, it is up to the caller to reset the "IOP type" field.

Bits 0-4 Invalid Operation Type Field

0 = no IOP error
1 = square root of a negative number, infinity in projective mode, or a not normalized number
2 = (+infinity) + (−infinity) in affine mode
3 = tried to convert NAN to binary integer
4 = in division: 0/0, infinity/infinity or divisor is not normalized and the dividend is not zero and is finite
5 = one of the input arguments was a trapping NAN
6 = unordered values compared via predicate other than = or ≠
7 = k out of range for BINDEC or p out of range for DECBIN
8 = projective closure use of +/− infinity
9 = 0 × infinity
10 = in REM arg2 is zero or not normalized or arg1 is infinite
11 = unused, reserved
12 = unused, reserved
13 = BINDEC integer too big to convert
14 = DECBIN cannot represent input string
15 = tried to MOV a single denormalized number to a double destination
16 = tried to return an unnormalized number to single or double (invalid result)
17 = division by zero with divide by zero trap disabled

TRAP VECTOR — If any of the traps occur, the FP package will *jump* indirectly through the trap address in the fpcb with an index in the A accumulator indicating the trap type:

0 = Invalid Operation
1 = Overflow
2 = Underflow
3 = Divide by Zero
4 = Unnormalized
5 = Integer Overflow
6 = Inexact Result

If more than 1 enabled trap occurs, the MC6839 will return the index of the highest priorty enabled error. Index = 0 = invalid operation is the highest priority, and, index = 6 is the lowest.

SPECIAL VALUES (SINGLE- AND DOUBLE-FORMAT)

The encoding of the special values are given below. Generally, when used as operands, the special values flow through an operation creating a predictable result. Note that as with normalized numbers the extended format differs slightly from the single- and double-formats.

ZERO

Zero is represented by a number with both a zero exponent and a zero significand. The sign is significant and differentiates between plus or minus zero.

s	0	0

INFINITY

The infinities are represented by a number with the maximum exponent and a zero significand. The sign differentiates plus or minus infinity.

s	1111......1111	0

DENORMALIZED (SMALL NUMBERS)

When a number is so small that its exponent is the smallest allowable normal biased value (1), and it is impossible to normalize the number without further decrementing the exponent, then the number will be allowed to become denormalized. The format for denormalized numbers has a zero exponent and a non-zero significand. Note that in this form the implicit bit is no longer 1 but is zero. The interpretation for denormalized numbers is:

Single: $X = (-1)^s \times 2^{-126} \times (0.\ \text{significand})$
Double: $X = (-1)^s \times 2^{-1022} \times (0.\ \text{significand})$

Note that the exponent is always interpreted as 2^{-126} for single and 2^{-1022} for double instead of 2^{-127} and 2^{-1023} as might be expected. This is necessary since the only way to insure the implicit bit becomes zero is to right shift the significand (divide by 2) and increment the exponent (multiply by 2). Thus, the exponent ends up with the interpretation of 2^{-126} or 2^{-1022}.

The format for denormalized numbers is:

s	0	non-zero

Note that zero may be considered a special case of denormalized numbers where the number is so small that the significand has been reduced to zero.

Examples:
Single:
$1.0 \times 2^{-128} = 0.25 \times 2^{-126} =$ \$00 20 00 00
Double:
$1.0 \times 2^{-1025} = 0.125 \times 2^{-1022} =$ \$00 02 00 00 00 00 00 00

NOT A NUMBER (NAN)

A number containing a NAN indicates that the number is not a valid floating number. NANs can be used to initialize areas in memory to indicate they have not had a valid floating point number stored in them. They are also created by the MC6839 to indicate that an operation could not return a valid result.

The format for a NAN has the largest allowable exponent, a non-zero significand, and an undefined sign. As an implementation feature (not required by the IEEE Proposed Floating Point Standard), the non-zero fraction and undefined sign are further defined:

d	1111....1111	t	operation address	00....0000

d: 0= This NAN has never entered into an operation with another NAN.
1= This NAN has entered into an operation with other NANs.

t: 0= This NAN will not necessarily cause an invalid operation trap when operated upon.
1= This NAN will cause an invalid operation trap when operated upon (trapping NAN).

Operation address:

The 16 bits, immediately to the right of the t bit, contain the address of the instruction immediately following the call to the FP package of the operation that caused the NAN to be created. If d (double NAN) is also set, the address is arbitrarily one of the addresses in the two or more offending NANs.

SPECIAL VALUES (EXTENDED FORMAT)

ZERO

Zero is represented by a number with the smallest unbiased exponent and a zero significand:

s	100.....0000	0

INFINITY

Infinity has the maximum unbiased exponent and a zero significand:

s	011111.....11	0

DENORMALIZED NUMBERS

Denormalized numbers have the smallest unbiased exponent and a non-zero significand:

s	100......000	0. non-zero

The exponent of denormalized extended and internal numbers is interpreted as having the exponent value 1 greater than the smallest unbiased exponent value. Thus, a denormalized number has the exponent − 16384, but has the value:

$(-1)^s \times 2^{-16383} \times 0.f$

Example:

$1.0 \times 2^{-16387} = 0625 \times 2^{-16383} =$ \$40 00 08 00 00 00 00 00 00 00

NANs

NANs have the largest unbiased exponent and a non-zero significand. The operation addresses "t" and "d" are implementation features and are the same as for single- and double-formats.

d	011.....1111	0	t	operation addr.	00000000

The operation address always appears in the 16 bits immediately to the right of the t bit.

UNNORMALZIED NUMBERS

Unnormalized numbers occur only in extended or internal format. Unnormalized numbers have an exponent greater than the minimum in the extended format (i.e., they are not denormalized or normal zero) but the explicit leading bit is a zero. If the significand is zero, this is an unnormalized zero. Even though unnormalized numbers and denormalized numbers are handled similarly in most cases, they should not be confused. Denormalized numbers are numbers that are very small — have minimum exponent — and hence have lost some bits of significance. Unnormalized numbers are not necessarily small (the exponent may be large or small) but the significand has lost some bits of significance, hence, the explicit bit and possibly some of the bits to the right of the explicit bit are zero.

s	>100...000	0. significand

Note that unnormalized numbers cannot be represented — and hence cannot exist — for single- and double-formats. Unnormalized numbers can only be created when denormalized numbers in single- or double-format are represented in extended or internal formats.

Example:

$.0625 \times 2^2$ (unnorm.) = $00 02 08 00 00 00 00 00 00 00

MC6839 CALLING SEQUENCE AND OPCODE REFERENCE TABLE

Function	Opcode	Register Calling Sequence	Stack Calling Sequence[1]
FADD FSUB FMUL FDIV	$00 $02 $04 $06	U←Addr. of Argument #1 Y←Addr. of Argument #2 D←Addr. of FPCB X←Addr. of Result LBSR FPREQ FCB <opcode>	Push Argument #1 Push Argument #2 Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull Result
FREM FSQRT FINT FFIXS FFIXD FAB FNEG FFLTS FFLTD	$08 $12 $14 $16 $18 $1E $20 $24 $26	Y←Addr. of Argument D←Addr. of FPCB X←Addr. of Result LBSR FPREG FCB <opcode>	Push Argument Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull Result
FCMP FTCMP FPCMP FTPCMP	$8A $CC $8E $D0	U←Addr. of Argument #1 Y←Addr. of Argument #2 D←Addr. of FPCB X←Parameter Word LBSR FPREG FCB <opcode> NOTE: Result returned in the CC register. For predicate compares the Z-Bit is set if predicate is affirmed cleared if disaffirmed.	Push Argument #1 Push Argument #2 Push Parameter Word Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull Result (if predicate compare, NOTE: Result returned in the CC register for regular compares. For predicate compares a one byte result is returned on the top of the stack. The result is zero if affirmed and −1($FF) if disaffirmed.
FMOV	$9A	U←Precision Parameter Word Y←Addr. of Argument D←Addr. of FPCB X←Addr. of Result LBSR FPREG FCB <opcode>	Push Argument Push Precision Parameter Word Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull Result
BINDEC	$1C	U←k (# of digits in result) Y←Addr. of Argument D←Addr. of FPCB X←Addr. of Decimal Result LBSR FPREG FCB <opcode>	Push Argument Push k Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull BCD String
DECBIN	$22	U←Addr. of BCD Input String D←Addr. of FPCB X←Addr. of Binary Result LBSR FPREG FCB <opcode>	Push Addr. of BCD Input String Push Addr. of FPCB LBSR FPSTAK FCB <opcode> Pull Binary Result

[1] All arguments are pushed on the stack least-significant bytes first so that the high-order byte is always pushed last and resides in the lowest address.

Entry points to the MC6839 are defined as follows:

FPREG = ROM start + $3D
FPSTAK = ROM start + $3F

MC6839 EXECUTION TIMES
Time in μs Using 2 MHz 6809

Function	Single Precision	Double Precision	Extended Precision

FADD	1200 – 3300
	t = 1200 + 40(A) + 50(N)
	where:
	A = # shifts to align operands
	N = # shifts to normalize result
		1500 – 3700
		t = 1500 + 40(A) + 50(N)
			1100 – 3800
			t = 1100 + 40(A) + 50(N)

FSUB	ADD + 11	ADD + 11	ADD + 11
FMUL	1400 – 1600	4100 – 4300	4600 – 4800
FDIV	t = 2700 + 60(Q) where: Q = # of quotient bits which are are a '1'	t = 5000 + 60(Q)	5 = 6500 + 60(Q)
FABS	540	750	650
DECBIN (time depends on magnitude of input)	8500 – 14,000	8500 – 23,000	–
BINDEC (time depends on # significand digits requested)	35,000 – 48,000	67,000 – 85,000	–

MOTOROLA

MC6840
(1.0 MHz)
MC68A40
(1.5 MHz)
MC68B40
(2.0 MHz)

MOS

(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

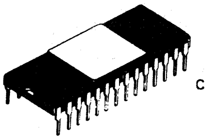
PROGRAMMABLE TIMER

PROGRAMMABLE TIMER MODULE (PTM)

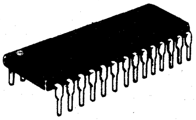
The MC6840 is a programmable subsystem component of the M6800 family designed to provide variable system time intervals.

The MC6840 has three 16-bit binary counters, three corresponding control registers, and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval measuring, and similar tasks. The device may be used for square wave generation, gated delay signals, single pulses of controlled duration, and pulse width modulation as well as system interrupts.

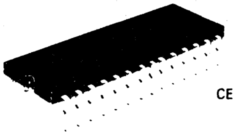
- Operates from a Single 5 Volt Power Supply
- Fully TTL Compatible
- Single System Clock Required (Enable)
- Selectable Prescaler on Timer 3 Capable of 4 MHz for the MC6840, 6 MHz for the MC68A40 and 8 MHz for the MC68B40
- Programmable Interrupts ($\overline{IRQ}$) Output to MPU
- Readable Down Counter Indicates Counts to Go Until Time-Out
- Selectable Gating for Frequency or Pulse-Width Comparison
- $\overline{RESET}$ Input
- Three Asynchronous External Clock and Gate/Trigger Inputs Internally Synchronized
- Three Maskable Outputs



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

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_L to T_h MC6840, MC68A40, MC68B40 MC6840C, MC68A40C	T_A	 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip Plastic Ceramic	θ_{JA}	 65 115 60	°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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

FIGURE 1 — PIN ASSIGNMENT

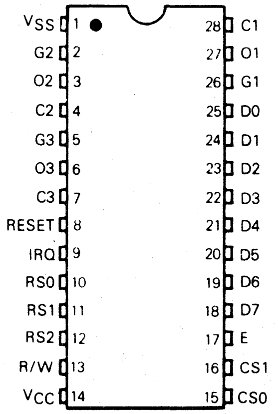
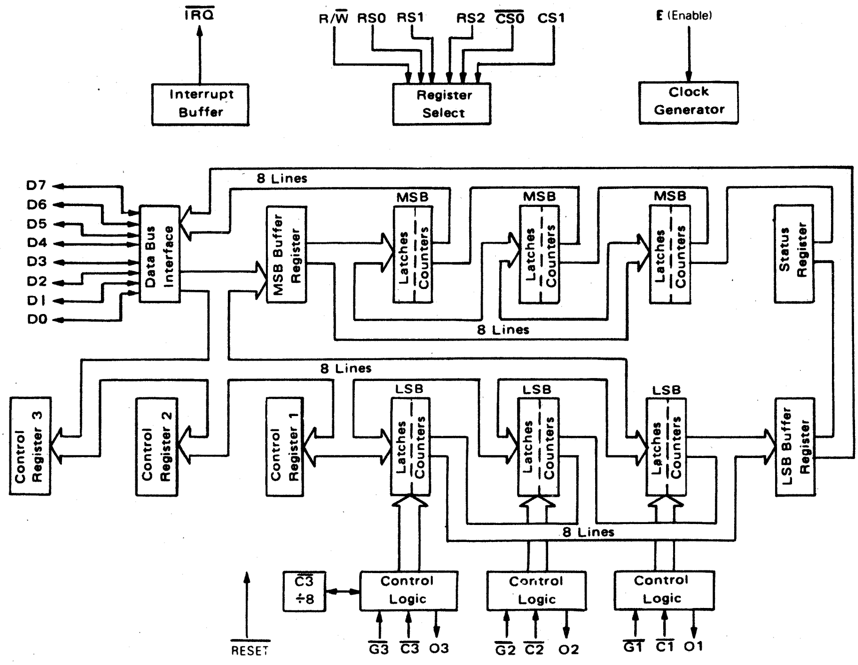


FIGURE 2 — BLOCK DIAGRAM



POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ 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:

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS}+ 2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	V_{SS} – 0.3	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{Load} = – 205 μA)	D0-D7	V_{OH}	V_{SS} + 2.4	–	–	V
(I_{Load} = – 200 μA)	Other Outputs		V_{SS} + 2.4	–	–	
Output Low Voltage (I_{Load} = 1.6 mA)	D0-D7	V_{OL}	–	–	V_{SS} + 0.4	V
(I_{Load} = 3.2 mA)	O1-O3, $\overline{IRQ}$		–	–	V_{SS} + 0.4	
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation (Measured at T_A = T_L)		P_{INT}	–	470	700	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Others		–	–	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{IRQ}$	C_{out}	–	–	5.0	pF
	O1, O2, O3		–	–	10	

AC OPERATING CHARACTERISTICS (See Figures 4-9)

Characteristic	Symbol	MC6840		MC68A40		MC68B40		Unit
		Min	Max	Min	Max	Min	Max	
Input Rise and Fall Times (Figures 4 and 5) $\overline{C}$, $\overline{G}$ and $\overline{RESET}$	t_r, t_f	–	1.0*	–	0.666*	–	0.500*	μs
Input Pulse Width Low (Figure 4) (Asynchronous Input) $\overline{C}$, $\overline{G}$ and $\overline{RESET}$	PW_L	t_{cycE} + t_{su} + t_{hd}	–	t_{cycE} + t_{su} + t_{hd}	–	t_{cycE} + t_{su} + t_{hd}	–	ns
Input Pulse Width High (Figure 5) (Asynchronous Input) $\overline{C}$, $\overline{G}$	PW_H	t_{cycE} + t_{su} + t_{hd}	–	t_{cycE} + t_{su} + t_{hd}	–	t_{cycE} + t_{su} + t_{hd}	–	ns
Input Setup Time (Figure 6) (Synchronous Input) $\overline{C}$, $\overline{G}$ and $\overline{RESET}$	t_{su}	200	–	120	–	75	–	ns
Input Hold Time (Figure 6) (Synchronous Input) $\overline{C}$, $\overline{G}$ and $\overline{RESET}$	t_{hd}	50	–	50	–	50	–	ns
Input Synchronization Time (Figure 9) $\overline{C3}$ (÷8 Prescaler Mode Only)	t_{sync}	250	–	200	–	175	–	ns
Input Pulse Width $\overline{C3}$ (÷8 Prescaler Mode Only)	PW_L, PW_H	120	–	80	–	60	–	ns
Output Delay, O1-O3 (Figure 7) (V_{OH} = 2.4 V, Load B) TTL	t_{co}	–	700	–	460	–	340	ns
(V_{OH} = 2.4 V, Load D) MOS	t_{cm}	–	450	–	450	–	340	ns
(V_{OH} = 0.7 V_{DD}, Load D) CMOS	t_{cmos}	–	2.0	–	1.35	–	1.0	μs
Interrupt Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	μs

*t_r and t_f ≤ t_{cycE}

BUS TIMING CHARACTERISTICS (See Notes 1, 2, and 3)

Ident. Number	Characteristic	Symbol	MC6840		MC68A40		MC68B40		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (High Impedance).

FIGURE 3 — BUS TIMING

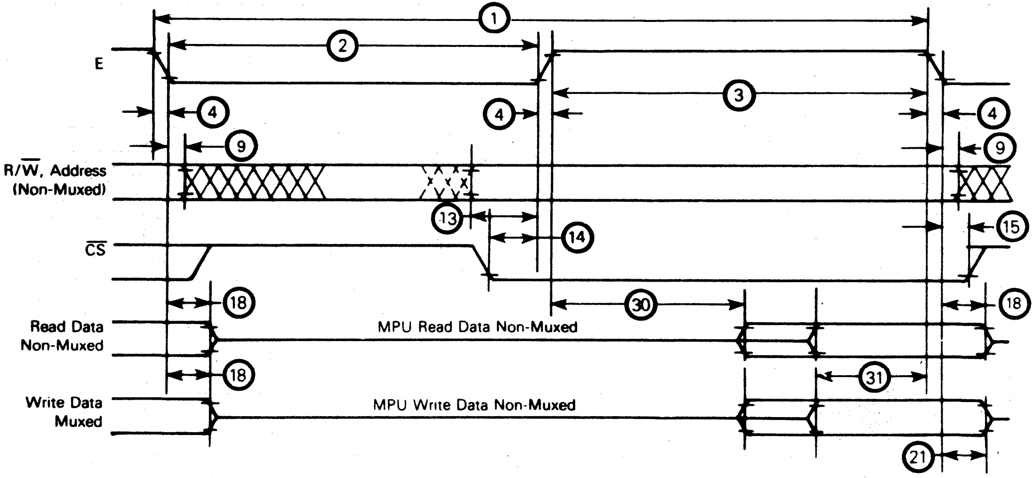


FIGURE 4 — INPUT PULSE WIDTH LOW

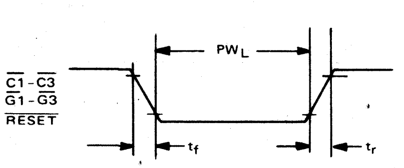
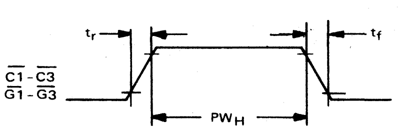


FIGURE 5 — INPUT PULSE WIDTH HIGH



NOTES:
1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 6 – INPUT SETUP AND HOLD TIMES

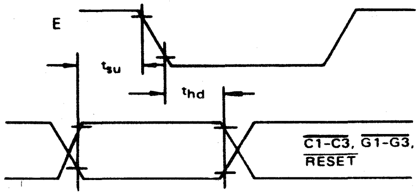
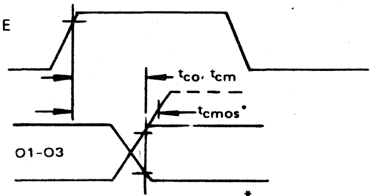


FIGURE 7 – OUTPUT DELAY



*t_{cmos} = 0.7 X V_{CC}

FIGURE 8 – $\overline{IRQ}$ RELEASE TIME

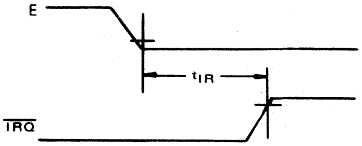


FIGURE 9 – $\overline{C3}$ INPUT SYNCHRONIZATION TIME (÷8 PRESCALER MODE ONLY)

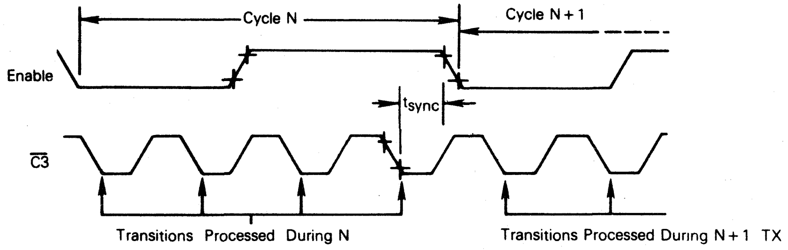
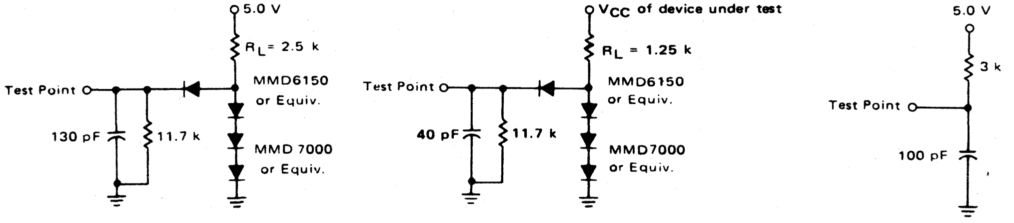


FIGURE 10 – BUS TIMING TEST LOADS

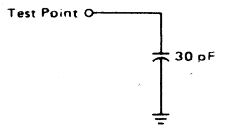
Load A (D0–D7)

Load B (O1, O2, O3) (TTL Load)

Load C ($\overline{IRQ}$ Only)



Load D (O1, O2, O3) (CMOS Load) (MOS)



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.

DEVICE OPERATION

The MC6840 is part of the M6800 microprocessor family and is fully bus compatible with M6800 systems. The three timers in the MC6840 operate independently and in several distinct modes to fit a wide variety of measurement and synthesis applications.

The MC6840 is an integrated set of three distinct counter/timers (Figure 1). It consists of three 16-bit data latches, three 16-bit counters (clocked independently), and the comparison and enable circuitry necessary to implement various measurement and synthesis functions. In addition, it contains interrupt drivers to alert the processor that a particular function has been completed.

In a typical application, a timer will be loaded by first storing two bytes of data into an associated Counter Latch. This data is then transferred into the counter via a Counter Initialization cycle. If the counter is enabled, the counter decrements on each subsequent clock period which may be an external clock, or Enable (E) until one of several predetermined conditions causes it to halt or recycle. The timers are thus programmable, cyclic in nature, controllable by external inputs or the MPU program, and accessible by the MPU at any time.

BUS INTERFACE

The Programmable Timer Module (PTM) interfaces to the M6800 Bus with an 8-bit bidirectional data bus, two Chip Select lines, a Read/Write line, a clock (Enable) line, and Interrupt Request line, an external Reset line, and three Register select lines. VMA should be utilized in conjunction with an MPU address line into a Chip Select of the PTM when using the MC6800/6802/6808.

BIDIRECTIONAL DATA (D0-D7) — The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and PTM. The data bus output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a PTM read operation (Read/Write and Enable lines high and PTM Chip Selects activated).

CHIP SELECT ($\overline{\text{CS0}}$, CS1) — These two signals are used to activate the Data Bus interface and allow transfer of data from the PTM. With $\overline{\text{CS0}}=0$ and $\text{CS1}=1$, the device is selected and data transfer will occur.

READ/WRITE (R/$\overline{\text{W}}$) — This signal is generated by the MPU to control the direction of data transfer on the Data Bus. With the PTM selected, a low state on the PTM R/$\overline{\text{W}}$ line enables the input buffers and data is transferred from the MPU to the PTM on the trailing edge of the E (Enable) clock. Alternately, (under the same conditions) R/$\overline{\text{W}}=1$ and Enable high allows data in the PTM to be read by the MPU.

ENABLE (E CLOCK) — The E clock signal synchronizes data transfer between the MPU and the PTM. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the PTM.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$) — The active low Interrupt Request signal is normally tied directly (or through priority interrupt circuitry) to the $\overline{\text{IRQ}}$ input of the MPU. This is an "open drain" output (no load device on the chip) which permits other similar interrupt request lines to be tied together in a wire-OR configuration.

The $\overline{\text{IRQ}}$ line is activated if, and only if, the Composite Interrupt Flag (Bit 7 of the Internal Status Register) is asserted. The conditions under which the $\overline{\text{IRQ}}$ line is activated are discussed in conjunction with the Status Register.

$\overline{\text{RESET}}$ — A low level at this input is clocked into the PTM by the E (Enable) input. Two Enable pulses are required to synchronize and process the signal. The PTM then recognizes the active "low" or inactive "high" on the third Enable pulse. If the $\overline{\text{RESET}}$ signal is asynchronous, an additional Enable period is required if setup times are not met. The $\overline{\text{RESET}}$ input must be stable High/Low for the minimum time stated in the AC Operating Characteristics.

Recognition of a low level at this input by the PTM causes the following action to occur:

a. All counter latches are preset to their maximum count values.

b. All Control Register bits are cleared with the exception of CR10 (internal reset bit) which is set.

c. All counters are preset to the contents of the latches.

d. All counter outputs are reset and all counter clocks are disabled.

e. All Status Register bits (interrupt flags) are cleared.

REGISTER SELECT LINES (RS0, RS1, RS2) — These inputs are used in conjunction with the R/$\overline{\text{W}}$ line to select the internal registers, counters and latches as shown in Table 1.

NOTE:

The PTM is accessed via MPU Load and Store operations in much the same manner as a memory device. The instructions available with the M6800 family of MPUs which perform read-modify-write operations on memory should not be used when the PTM is accessed. These instructions actually fetch a byte from memory, perform an operation, then restore it to the same address location. Since the PTM uses the R/$\overline{\text{W}}$ line as an additional register select input, the modified data will not be restored to the same register if these instructions are used.

CONTROL REGISTER

Each timer in the MC6840 has a corresponding write-only Control Register. Control Register #2 has a unique address space (RS0 = 1, RS = 0, RS2 = 0) and therefore may be written into at any time. The remaining Control Registers (#1 and #3) share the Address Space selected by a logic zero on all Register Select inputs.

CR20 — The least-significant bit of Control Register #2 (CR20) is used as an additional addressing bit for Control Registers #1 and #3. Thus, with all Register selects and R/W inputs at logic zero, Control Register #1 will be written into if CR20 is a logic one. Under the same conditions, Control Register #3 can also be written into after a $\overline{\text{RESET}}$ low condition has occurred, since all control register bits (except CR10) are cleared. Therefore, one may write in the sequence CR3, CR2, CR1.

TABLE 1 – REGISTER SELECTION

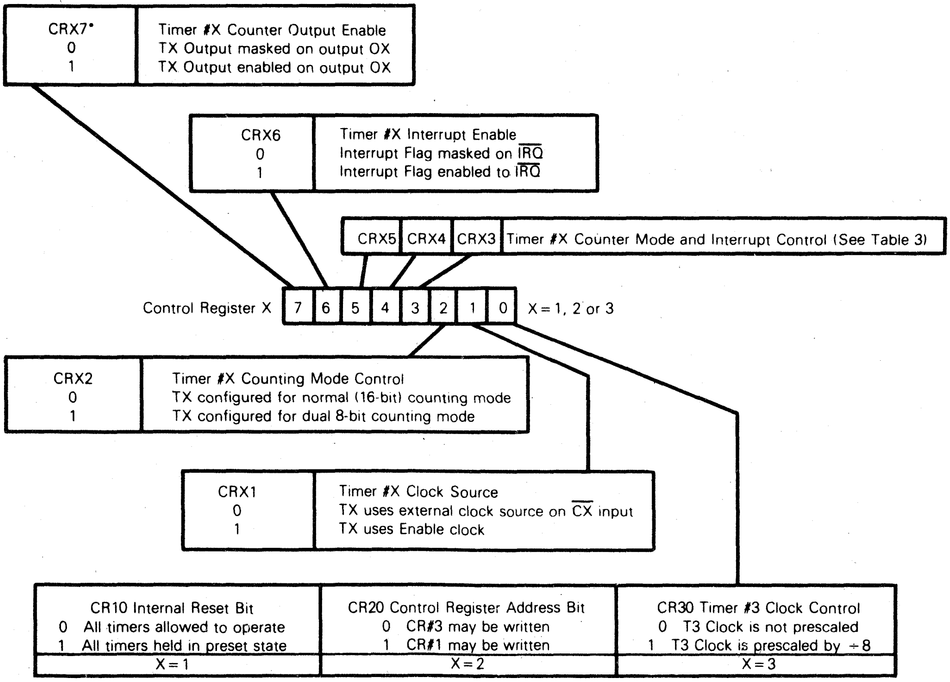
Register Select Inputs			Operations	
RS2	RS1	RS0	R/$\overline{W}$ = 0	R/$\overline{W}$ = 1
0	0	0	CR20 = 0 Write Control Register #3 CR20 = 1 Write Control Register #1	No Operation
0	0	1	Write Control Register #2	Read Status Register
0	1	0	Write MSB Buffer Register	Read Timer #1 Counter
0	1	1	Write Timer #1 Latches	Read LSB Buffer Register
1	0	0	Write MSB Buffer Register	Read Timer #2 Counter
1	0	1	Write Timer #2 Latches	Read LSB Buffer Register
1	1	0	Write MSB Buffer Register	Read Timer #3 Counter
1	1	1	Write Timer #3 Latches	Read LSB Buffer Register

CR10 — The least-significant bit of Control Register #1 is used as an Internal Reset bit. When this bit is a logic zero, all timers are allowed to operate in the modes prescribed by the remaining bits of the control registers. Writing a "one" into CR10 causes all counters to be preset with the contents of the corresponding counter latches, all counter clocks to be disabled, and the timer outputs and interrupt flags (Status Register) to be reset. Counter Latches and Control Registers are undisturbed by an Internal Reset and may be written into regardless of the state of CR10.

The least-signifcant bit of Control Register #3 is used as a selector for a ÷8 prescaler which is available with Timer #3 only. The prescaler, if selected, is effectively placed between the clock input circuitry and the input to Counter #3. It can therefore be used with either the internal clock (Enable) or an external clock source.

CR30 — The functions depicted in the foregoing discussions are tabulated in Table 2 for ease of reference.

TABLE 2 — CONTROL REGISTER BITS



CRX7*	Timer #X Counter Output Enable
0	TX Output masked on output OX
1	TX Output enabled on output OX

CRX6	Timer #X Interrupt Enable
0	Interrupt Flag masked on $\overline{IRQ}$
1	Interrupt Flag enabled to $\overline{IRQ}$

CRX5 CRX4 CRX3 Timer #X Counter Mode and Interrupt Control (See Table 3)

CRX2	Timer #X Counting Mode Control
0	TX configured for normal (16-bit) counting mode
1	TX configured for dual 8-bit counting mode

CRX1	Timer #X Clock Source
0	TX uses external clock source on $\overline{CX}$ input
1	TX uses Enable clock

CR10 Internal Reset Bit	CR20 Control Register Address Bit	CR30 Timer #3 Clock Control
0 All timers allowed to operate	0 CR#3 may be written	0 T3 Clock is not prescaled
1 All timers held in preset state	1 CR#1 may be written	1 T3 Clock is prescaled by ÷8
X = 1	X = 2	X = 3

Control Register Bits CR10, CR20, and CR30 are unique in that each selects a different function. The remaining bits (1 through 7) of each Control Register select common functions, with a particular Control Register affecting only its corresponding timer.

CRX1 – Bit 1 of Control Register #1 (CR11) selects whether an internal or external clock source is to be used with Timer #1. Similarly, CR21 selects the clock source for Timer #2, and CR31 performs this function for Timer #3. The function of each bit of Control Register "X" can therefore be defined as shown in the remaining section of Table 2.

CRX2 – Control Register Bit 2 selects whether the binary information contained in the Counter Latches (and subsequently loaded into the counter) is to be treated as a single 16-bit word or two 8-bit bytes. In the single 16-bit Counter Mode (CRX2 = 0) the counter will decrement to zero after N + 1 enabled (G = 0) clock periods, where N is defined as the 16-bit number in the Counter Latches. With CRX2 = 1, a similar Time Out will occur after (L + 1)•(M + 1) enabled clock periods, where L and M, respectively, refer to the LSB and MSB bytes in the Counter Latches.

CRX3-CRX7 – Control Register Bits 3, 4, and 5 are explained in detail in the Timer Operating Mode section. Bit 6 is an interrupt mask bit which will be explained more fully in conjunction with the Status Register, and bit 7 is used to enable the corresponding Timer Output. A summary of the control register programming modes is shown in Table 3.

STATUS REGISTER/INTERRUPT FLAGS

The MC6840 has an internal Read-Only Status Register which contains four Interrupt Flags. (The remaining four bits of the register are not used, and defaults to zeros when being read.) Bits 0, 1, and 2 are assigned to Timers 1, 2, and 3, respectively, as individual flag bits, while Bit 7 is a Composite Interrupt Flag. This flag bit will be asserted if any of the individual flag bits is set while Bit 6 of the corresponding Control Register is at a logic one. The conditions for asserting the composite Interrupt Flag bit can therefore be expressed as:

$$INT = I1 \cdot CR16 + I2 \cdot CR26 + I3 \cdot CR36$$

where INT = Composite Interrupt Flag (Bit 7)
I1 = Timer #1 Interrupt Flag (Bit 0)
I2 = Timer #2 Interrupt Flag (Bit 1)
I3 = Timer #3 Interrupt Flag (Bit 2)

An interrupt flag is cleared by a Timer Reset condition, i.e., External $\overline{\text{RESET}} = 0$ or Internal Reset Bit (CR10) = 1. It will also be cleared by a Read Timer Counter Command provided that the Status Register has previously been read while the interrupt flag was set. This condition on the Read Status Register-Read Timer Counter (RS-RT) sequence is designed to prevent missing interrupts which might occur after the status register is read, but prior to reading the Timer Counter.

An Individual Interrupt Flag is also cleared by a Write Timer Latches (W) command or a Counter Initialization (CI) sequence, provided that W or CI affects the Timer corresponding to the individual Interrupt Flag.

COUNTER LATCH INITIALIZATION

Each of the three independent timers consists of a 16-bit addressable counter and a 16-bit addressable latch. The counters are preset to the binary numbers stored in the latches. Counter initialization results in the transfer of the latch contents to the counter. See notes in Table 4 regarding the binary number N, L, or M placed into the Latches and their relationship to the output waveforms and counter Time-Outs.

Since the PTM data bus is 8-bits wide and the counters are 16-bits wide, a temporary register (MSB Buffer Register) is provided. This "write only" register is for the Most-Significant Byte of the desired latch data. Three addresses are provided for the MSB Buffer Register (as indicated in Table 1), but they all lead to the same Buffer. Data from the MSB Buffer will automatically be transferred into the Most-Significant Byte of Timer #X when a Write Timer #X Latches Command is performed. So it can be seen that the MC6840 has been designed to allow transfer of two bytes of data into the counter latches provided that the MSB is transferred first. The storage order must be observed to ensure proper latch operation.

In many applications, the source of the data will be an M6800 Family MPU. It should be noted that the 16-bit store operations of the M6800 family microprocessors (STS and STX) transfer data in the order required by the PTM. A Store Index Register Instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transferred directly into a selected counter latch with a single instruction.

A logic zero at the $\overline{\text{RESET}}$ input also initializes the counter latches. In this case, all latches will assume a maximum count of $65{,}535_{10}$. It is important to note that an Internal

TABLE 3 – PTM OPERATING MODE SELECTION

CRX3	CRX4	CRX5	
0	0	0	Continuous Operating Mode: Gate ↓ or Write to Latches or Reset Causes Counter Initialization
0	0	1	Frequency Comparison Mode: Interrupt If Gate ↑▁▁┌▔▔↓ is < Counter Time Out
0	1	0	Continuous Operating Mode: Gate ↓ or Reset Causes Counter Initialization
0	1	1	Pulse Width Comparison Mode: Interrupt if Gate ↑▁▁▁↑ is < Counter Time Out
1	0	0	Single Shot Mode: Gate ↓ or Write to Latches or Reset Causes Counter Initialization
1	0	1	Frequency Comparison Mode: Interrupt If Gate ↑▁▁┌▔▔↓ is > Counter Time Out
1	1	0	Single Shot Mode: Gate ↓ or Reset Causes Counter Initialization
1	1	1	Pulse Width Comparison Mode: Interrupt If Gate ↑▁▁▁↑ is > Counter Time Out

Reset (Bit zero of Control Register 1 Set) has no effect on the counter latches.

COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with subsequent clearing of the Individual Interrupt Flag associated with the counter. Counter Initialization always occurs when a reset condition ($\overline{\text{RESET}}$ = 0 or CR10 = 1) is recognized. It can also occur – depending on Timer Mode – with a Write Timer Latches command or recognition of a negative transition of the Gate input.

Counter recycling or re-initialization occurs when a negative transition of the clock input is recognized after the counter has reached an all-zero state. In this case, data is transferred from the Latches to the Counter.

ASYNCHRONOUS INPUT/OUTPUT LINES

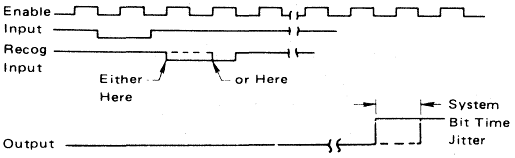
Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line. The inputs are high-impedance, TTL-compatible lines and ouputs are capable of driving two standard TTL loads.

CLOCK INPUTS ($\overline{\text{C1}}$, $\overline{\text{C2}}$, and $\overline{\text{C3}}$) – Input pins $\overline{\text{C1}}$, $\overline{\text{C2}}$, and $\overline{\text{C3}}$ will accept asynchronous TTL voltage level signals to decrement Timers 1, 2, and 3, respectively. The high and low levels of the external clocks must each be stable for at least one system clock period plus the sum of the setup and hold times for the clock inputs. The asynchronous clock rate can vary from dc to the limit imposed by the Enable Clock Setup, and Hold times.

The external clock inputs are clocked in by Enable pulses. Three Enable periods are used to synchronize and process the external clock. The fourth Enable pulse decrements the internal counter. This does not affect the input frequency, it merely creates a delay between a clock input transition and internal recognition of that transition by the PTM. All references to C inputs in this document relate to internal recognition of the input transition. Note that a clock high or low level which does not meet setup and hold time specifications may require an additional Enable pulse for recognition. When observing recurring events, a lack of synchronization will result in "jitter" being observed on the output of the PTM when using asynchronous clocks and gate input signals. There are two types of jitter. "System jitter" is the result of the input signals being out of synchronization with Enable, permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or the subsequent bit time.

"Input jitter" can be as great as the time between input signal negative going transitions plus the system jitter, if the first transition is recognized during one system cycle, and not recognized the next cycle, or vice versa. See Figure 11.

FIGURE 11 – INPUT JITTER



CLOCK INPUT $\overline{\text{C3}}$ (÷8 PRESCALER MODE) – External clock input $\overline{\text{C3}}$ represents a special case when Timer #3 is programmed to utilize its optional ÷8 prescaler mode.

The divide-by-8 prescaler contains an asynchronous ripple counter; thus, input setup (t_{su}) and hold times (t_{hd}) do not apply. As long as minimum input pulse widths are maintained, the counter will recognize and process all input clock ($\overline{\text{C3}}$) transitions. However, in order to guarantee that a clock transition is processed during the current E cycle, a certain amount of synchronization time (t_{sync}) is required between the $\overline{\text{C3}}$ transition and the falling edge of Enable (see Figure 9). If the synchronization time requirement is not met, it is possible that the $\overline{\text{C3}}$ transition will not be processed until the following E cycle.

The maximum input frequency and allowable duty cycles for the ÷8 prescaler mode are specified under the AC Operating Characteristics. Internally, the ÷8 prescaler output is treated in the same manner as the previously discussed clock inputs.

GATE INPUTS ($\overline{\text{G1}}$, $\overline{\text{G2}}$, $\overline{\text{G3}}$) – Input pins $\overline{\text{G1}}$, $\overline{\text{G2}}$, and $\overline{\text{G3}}$ accept asynchronous TTL-compatible signals which are used as triggers or clock gating functions to Timers 1, 2, and 3, respectively. The gating inputs are clocked into the PTM by the E (enable) clock in the same manner as the previously discussed clock inputs. That is, a Gate transition is recognized by the PTM on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the Gate input must be stable for at least one system clock period plus the sum of setup and hold times. All references to G transition in this document relate to internal recognition of the input transition.

The Gate inputs of all timers directly affect the internal 16-bit counter. The operation of $\overline{\text{G3}}$ is therefore independent of the ÷8 prescaler selection.

TIMER OUTPUTS (O1, O2, O3) – Timer outputs O1, O2, and O3 are capable of driving up to two TTL loads and produce a defined output waveform for either Continuous or Single-Shot Timer modes. Output waveform definition is accomplished by selecting either Single 16-bit or Dual 8-bit operating modes. The Single 16-bit mode will produce a square-wave output in the continuous mode and a single pulse in the single-shot mode. The Dual 8-bit mode will produce a variable duty cycle pulse in both the continuous and single-shot timer modes. One bit of each Control Register (CRX7) is used to enable the corresponding output. If this bit is cleared, the output will remain low (V_{OL}) regardless of the operating mode. If it is cleared while the output is high the output will go low during the first enable cycle following a write to the Control Register.

The Continuous and Single-Shot Timer Modes are the only ones for which output response is defined in this data sheet. Refer to the Programmable Timer Fundamentals and Applications manual for a discussion of the output signals in other modes. Signals appear at the outputs (unless CRX7 = 0) during Frequency and Pulse Width comparison modes, but the actual waveform is not predictable in typical applications.

TIMER OPERATING MODES

The MC6840 has been designed to operate effectively in a wide variety of applications. This is accomplished by using three bits of each control register (CRX3, CRX4, and CRX5) to define different operating modes of the Timers. These modes are divided into WAVE SYNTHESIS and WAVE MEASUREMENT modes, and are outlined in Table 4.

TABLE 4 – OPERATING MODES

Control Register			Timer Operating Mode	
CRX3	CRX4	CRX5		
0	•	0	Continuous	Synthesizer
0	•	1	Single-Shot	
1	0	•	Frequency Comparison	Measurement
1	1	•	Pulse Width Comparison	

•Defines Additional Timer Function Selection.

One of the WAVE SYNTHESIS modes is the Continuous Operating mode, which is useful for cyclic wave generation. Either symmetrical or variable duty-cycle waves can be generated in this mode. The other wave synthesis mode, the Single-Shot mode, is similar in use to the Continuous operating mode, however, a single pulse is generated, with a programmable preset width.

The WAVE MEASUREMENT modes include the Frequency Comparison and Pulse Width Comparison modes which are used to measure cyclic and singular pulse widths, respectively.

In addition to the four timer modes in Table 4, the remaining control register bit is used to modify counter initialization and enabling or interrupt conditions.

WAVE SYNTHESIS MODES

CONTINUOUS OPERATING MODE (TABLE 5) – The continuous mode will synthesize a continuous wave with a period proportional to the preset number in the particular timer latches. Any of the timers in the PTM may be programmed to operate in a continuous mode by writing zeroes into bits 3 and 5 of the corresponding control register. Assuming that the timer output is enabled (CRX7 = 1), either a square wave or a variable duty cycle waveform will be generated at the Timer Output, OX. The type of output is selected via Control Register Bit 2.

Either a Timer Reset (CR10 = 1 or External Reset = 0) condition or internal recognition of a negative transition of the Gate input results in Counter Initialization. A Write Timer latches command can be selected as a Counter Initialization signal by clearing CRX4.

The counter is enabled by an absence of a Timer Reset condition and a logic zero at the Gate input. In the 16-bit mode, the counter will decrement on the first clock cycle during or after the counter initialization cycle. It continues to decrement on each clock signal so long as G remains low and no reset condition exists. A Counter Time Out (the first clock after all counter bits = 0) results in the Individual Interrupt Flag being set and reinitialization of the counter.

In the Dual 8-bit mode (CRX2 = 1) [refer to the example in Figure 12 and Tables 5 and 6] the MSB decrements once for every full countdown of the LSB + 1. When the LSB = 0, the MSB is unchanged; on the next clock pulse the LSB is reset to the count in the LSB Latches, and the MSB is decremented by 1 (one). The output, if enabled, remains low during and after initialization and will remain low until the counter MSB is all zeroes. The output will go high at the beginning of the next clock pulse. The output remains high until both the LSB and MSB of the counter are all zeroes. At the beginning of the next clock pulse the defined Time Out (TO) will occur and the output will go low. In the Dual 8-bit mode the period of the output of the example in Figure 12 would span 20 clock pulses as opposed to 1546 clock pulses using the normal 16-bit mode.

A special time-out condition exists for the dual 8-bit mode (CRX2 = 1) if L = 0. In this case, the counter will revert to a mode similar to the single 16-bit mode, except Time Out occurs after M + 1* clock pulses. The output, if enabled, goes low during the Counter Initialization cycle and reverses state at each Time Out. The counter remains cyclical (is reinitialized at each Time Out) and the Individual Interrupt Flag is set when Time Out occurs. If M = L = 0, the internal counters do not change, but the output toggles at a rate of ½ the clock frequency.

TABLE 5 – CONTINUOUS OPERATING MODES

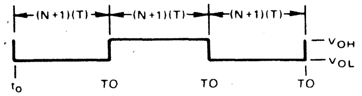
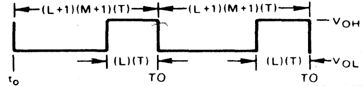
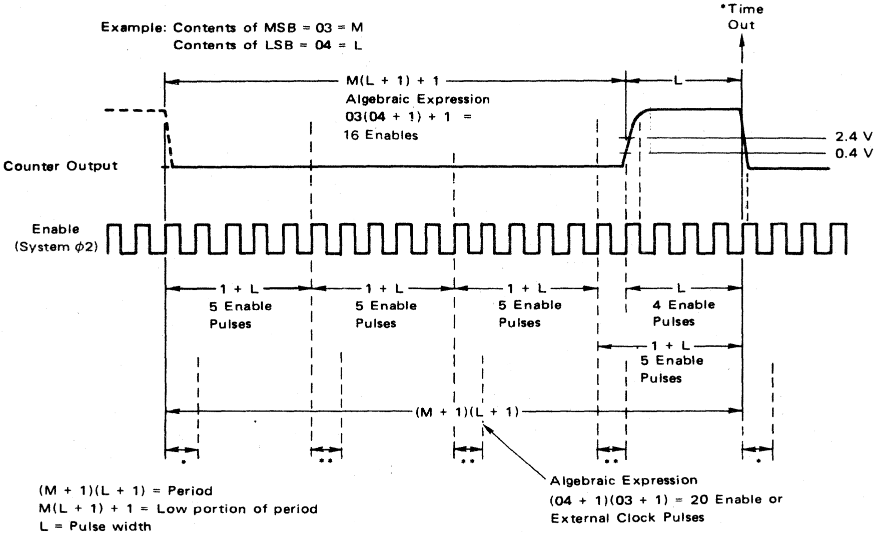
Synthesis Modes		CONTINUOUS MODE (CRX3 = 0, CRX5 = 0)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = 1)
0	0	$\overline{G}\downarrow+W+R$	
0	1	$\overline{G}\downarrow+R$	
1	0	$\overline{G}\downarrow+W+R$	
1	1	$\overline{G}\downarrow+R$	

FIGURE 12 – TIMER OUTPUT WAVEFORM EXAMPLE
(Continuous Dual 8-Bit Mode Using Internal Enable)



(M + 1)(L + 1) = Period
M(L + 1) + 1 = Low portion of period
L = Pulse width
*Preset LSB and MSB to Respective Latches on the negative transition of the Enable
**Preset LSB to LSB Latches and Decrement MSB by one on the negative transition of the Enable

The discussion of the Continuous Mode has assumed that the application requires an output signal. It should be noted that the Timer operates in the same manner with the output disabled (CRX7=0). A Read Timer Counter command is valid regardless of the state of CRX7.

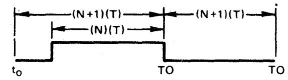
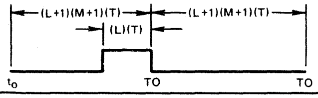
SINGLE-SHOT TIMER MODE – This mode is identical to the Continuous Mode with three exceptions. The first of these is obvious from the name – the output returns to a low level after the initial Time Out and remains low until another Counter Initialization cycle occurs.

As indicated in Table 6, the internal counting mechanism remains cyclical in the Single-Shot Mode. Each Time Out of the counter results in the setting of an Individual Interrupt Flag and re-initialization of the counter.

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the Gate input level reamining in the low state for the Single-Shot mode.

Another special condition is introduced in the Single-Shot mode. If L=M=0 (Dual 8-bit) or N=0 (Single 16-bit), the output goes low on the first clock received during or after Counter Initialization. The output remains low until the Operating Mode is changed or nonzero data is written into the Counter Latches. Time Outs continue to occur at the end of each clock period.

TABLE 6 – SINGLE-SHOT OPERATING MODES

Synthesis Modes		SINGLE-SHOT MODE (CRX3 = 0, CRX7 = 1, CRX5 = 1)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	Timer Output (OX)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	
1	1	$\overline{G}\downarrow$+R	

Symbols are as defined in Table 5.

The three differences between Single-Shot and Continous Timer Mode can be summarized as attributes of the Single-Shot mode:

1. Output is enabled for only one pulse until it is reinitialized.
2. Counter Enable is independent of Gate.
3. L = M = 0 or N = 0 disables output.

Aside from these differences, the two modes are identical.

WAVE MEASUREMENT MODES

TIME INTERVAL MODES — The Time Interval Modes are the Frequency (period) Measurement and Pulse Width Comparison Modes, and are provided for those applications which require more flexibility of interrupt generation and Counter Initialization. Individual Interrupt Flags are set in these modes as a function of both Counter Time Out and transitions of the Gate input. Counter Initialization is also affected by Interrupt Flag status.

A timer's output is normally not used in a Wave Measurement mode, but it is defined. If the output is enabled, it will operate as follows. During the period between reinitialization of the timer and the first Time Out, the output will be a logical zero. If the first Time Out is completed (regardless of its method of generation), the output will go high. If further TO's occur, the output will change state at each completion of a Time-Out.

The counter does operate in either Single 16-bit or Dual 8-bit modes as programmed by CRX2. Other features of the Wave Measurement Modes are outlined in Table 7.

Frequency Comparison Or Period Measurement Mode (CRX3 = 1, CRX4 = 0) — The Frequency Comparison Mode with CRX5 = 1 is straightforward. If Time Out occurs prior to the first negative transition of the Gate input after a Counter Initialization cycle, and Individual Interrupt Flag is set. The counter is disabled, and a Counter Initialization cycle cannot begin until the interrupt flag is cleared and a negative transition on G is detected.

If CRX5 = 0, as shown in Tables 7 and 8, an interrupt is generated if Gate input returns low prior to a Time Out. If a Counter Time Out occurs first, the counter is recycled and continues to decrement. A bit is set within the timer on the initial Time Out which precludes further individual interrupt generation until a new Counter Initialization cycle has been completed. When this internal bit is set, a negative transition of the Gate input starts a new Counter Initialization cycle. (The condition of $\overline{G\downarrow} \cdot \overline{I} \cdot TO$ is satisfied, since a Time Out has occurred and no individual Interrupt has been generated.)

Any of the timers within the PTM may be programmed to compare the period of a pulse (giving the frequency after calculations) at the Gate input with the time period requested for Counter Time Out. A negative transition of the Gate Input enables the counter and starts a Counter Initialization cycle — provided that other conditions, as noted in Table 8, are satisfied. The counter decrements on each clock signal recognized during or after Counter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs. It can be seen from Table 8 that an interrupt condition will be generated if CRX5 = 0 and the period of the pulse (single pulse or measured separately repetitive pulses) at the Gate input is less than the Counter Time Out period. If CRX5 = 1, an interrupt is generated if the reverse is true.

Assume now with CRX5 = 1 that a Counter Initialization has occurred and that the Gate input has returned low prior to Counter Time Out. Since there is no Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle. The process will continue with frequency comparison being performed on each Gate input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit.

Pulse Width Comparison Mode (CRX3 = 1, CRX4 = 1) — This mode is similar to the Frequency Comparison Mode except for a positive, rather than negative, transition of the Gate input termintes the count. With CRX5 = 0, an Individual Interrupt Flag will be generated if the zero level pulse applied to the Gate input is less than the time period required for Counter Time Out. With CRX5 = 1, the interrupt is generated when the reverse condition is true.

As can be seen in Table 8, a positive transition of the Gate input disables the counter. With CRX5 = 0, it is therefore possible to directly obtain the width of any pulse causing an interrupt. Similar data for other Time Interval Modes and conditions can be obtained, if two sections of the PTM are dedicated to the purpose.

FIGURE 7 — OUTPUT DELAY

CRX3 = 1			
CRX4	**CRX5**	**Application**	**Condition for Setting Individual Interrupt Flag**
0	0	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is less than Counter Time Out (TO)
0	1	Frequency Comparison	Interrupt Generated if Gate Input Period (1/F) is greater than Counter Time Out (TO)
1	0	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is less than Counter Time Out (TO)
1	1	Pulse Width Comparison	Interrupt Generated if Gate Input "Down Time" is greater than Counter Time Out (TO)

TABLE 8 — FREQUENCY COMPARISON MODE

Mode	Bit 3	Bit 4	Control Reg. Bit 5	Counter Initialization	Counter Enable Flip-Flop Set (CE)	Counter Enable Flip-Flop Reset (CE)	Interrupt Flag Set (I)
Frequency Comparison	1	0	0	$\overline{G}\downarrow \cdot I \pm (\overline{CE}+TO)+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	$W+R+I$	$\overline{G}\downarrow$ Before TO
	1	0	1	$\overline{G}\downarrow \cdot \overline{I}+R$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	$W+R+I$	TO Before $\overline{G}\downarrow$
Pulse Width Comparison	1	1	0	$\overline{G}\downarrow \cdot \overline{I}+\overline{R}$	$\overline{G}\downarrow \overline{W} \cdot \overline{R} \cdot \overline{I}$	$W+R+I+G$	$\overline{G}\uparrow$ Before TO
	1	1	1	$\overline{G}\downarrow \cdot \overline{I}+\overline{R}$	$\overline{G}\downarrow \cdot \overline{W} \cdot \overline{R} \cdot \overline{I}$	$W+R+I+G$	$\overline{G}\uparrow$ Before TO

$\overline{G}\downarrow$ = Negative transition of $\overline{\text{Gate}}$ input.
W = Write Timer Latches Command.
R = Timer Reset (CR10 = 1 or External $\overline{\text{RESET}}$ = 0)
N = 16-Bit Number in Counter Latch.
TO = Counter Time Out (All Zero Condition)
I = Interrupt for a given timer.

*All time intervals shown above assume the Gate ($\overline{G}$) and Clock ($\overline{C}$) signals are sycnhronized to the system clock (E) with the specified setup and hold time requirements.

MOTOROLA

MC6843

FLOPPY DISK CONTROLLER (FDC)

The MC6843 Floppy Disk Controller performs the complex MPU/Floppy interface function. The FDC was designed to optimize the balance between Hardware and Software in order to achieve integration of all key functions and maintain flexibility.

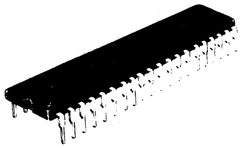
The FDC can interface a wide range of drives with a minimum of external hardware. Multiple drives can be controlled with the addition of external multiplexing rather than additional FDCs.

- Format Compatible with IBM 3740
- User Programmable Read/Write Format
- Ten Powerful Macro Commands
- Macro-End Interrupt Allows Parallel Processing of MPU and FDC
- Controls Multiple Floppies with External Multiplexing
- Direct Interface with M6800 Bus
- Programmable Step and Settling Times Enable Operation with a Wide Range of Floppy Drives
- Offers Both Program Controlled I/O (PCIO) and DMA Data Transfer Mode
- Free-Format Read or Write
- Single 5-Volt Power Supply
- All Registers Directly Accessible

MOS

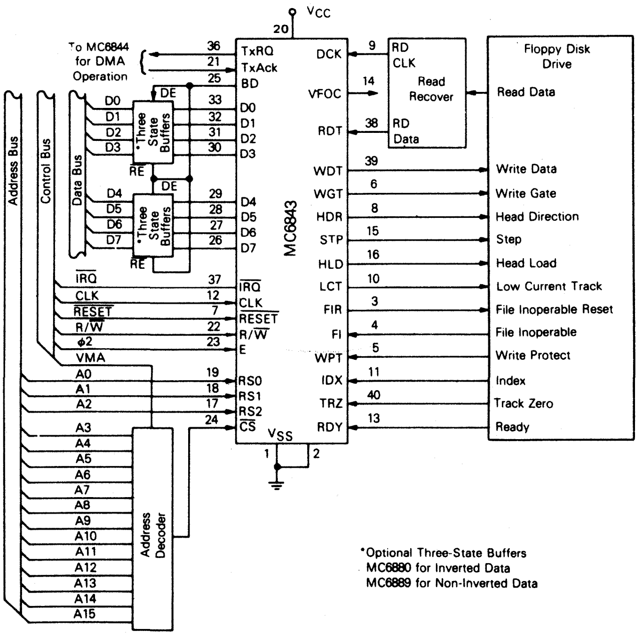
(N-CHANNEL, SILICON-GATE)

FLOPPY DISK CONTROLLER



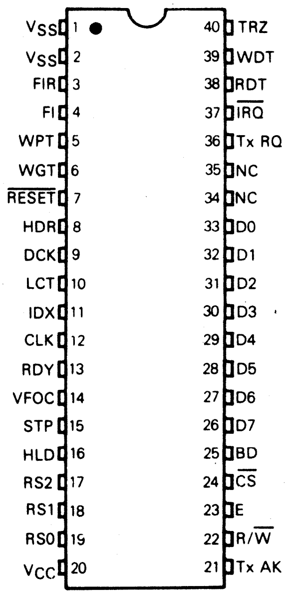
P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 — SYSTEM BLOCK DIAGRAM



*Optional Three-State Buffers
MC6880 for Inverted Data
MC6889 for Non-Inverted Data

FIGURE 2 — PIN ASSIGNMENT



MAXIMUM RATING

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	θ_{JA}	100	°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. 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:

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

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:

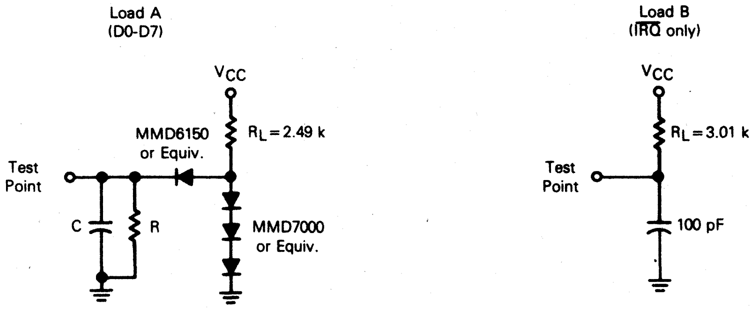
$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

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

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.

FIGURE 3 — TEST LOADS



C = 130 pF for D0-D7
= 30 pF for HLD, STP, HDR, LCT, WGT, FIR TxRQ, BD
R = 11.7 k for D0-D7
= 24 k for HLD, STP, HDR, LCR, LCT, WGT, FIR TxRQ, BD

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

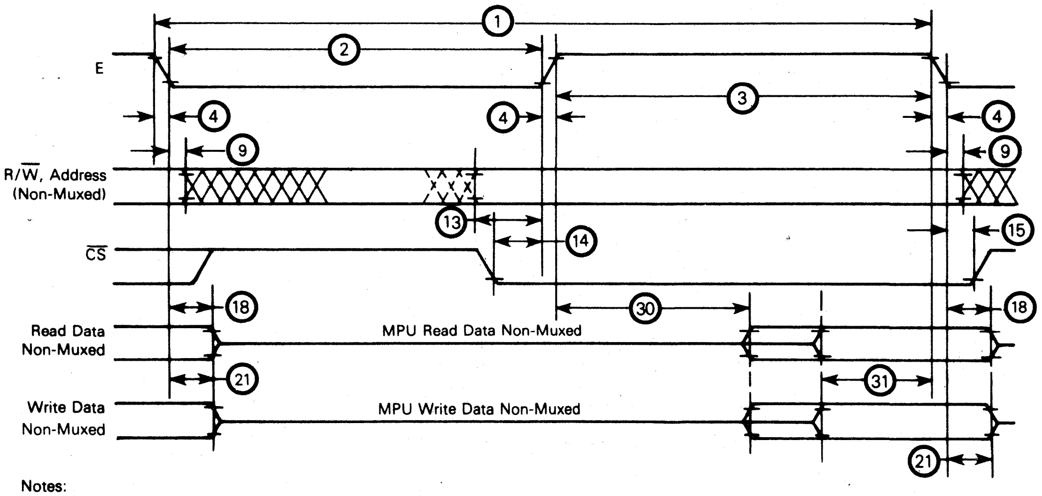
Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	$V_{SS}+2.0$	—	V_{CC}	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$	—	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.5 V)	I_{in}	—	1.0	2.5	μA
Three-State Input Leakage Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.5 V) D0-D7	I_{IZ}	−10	2.0	10	μA
Output High Voltage (I_{load} = −205 μA) D0-D7 (I_{load} = −100 μA) Other Outputs	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$	 — —	 — —	V
Output Low Voltage (I_{load} = 1.6 mA)	V_{OL}	—	—	$V_{SS}+0.4$	V
Three-State Output Leakage Current (V_{OH} = 2.4 V) IRQ	I_{OZ}	—	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$ to T_H)	P_{INT}	—	—	750	mW
Input Capacitance (V_{in} = 0 V, f = 1.0 MHz, T_A = 25°C) Enable D0-D7 All Others	C_{in}	 — — —	 — — —	 10 12.5 10	pF
Output Capacitance (V_{in} = 0 V, f = 1.0 MHz, T_A = 25°C) All Outputs	C_{out}	—	—	10	pF
Clock Pulse Width, Low (CLK)	PW_{CL}	400	—	—	ns
Clock Pulse Width, High (CLK)	PW_{CH}	400	—	—	ns
Master Clock Period (CLK)	t_{MC}	1.0	—	—	μs
Data Clock Pulse Width, Low (DCK)	PW_{DL}	1.3	1.95	—	μs
Data Clock Pulse Width, High (DCK)	PW_{DH}	1.3	1.95	—	μs
Data Clock Period (DCK)	t_{DC}	2.5	4.0	—	μs
Read Data to Data Clock Delay Time 1	t_{RDD1}	0.55	1.0	—	μs
Read Data to Data Clock Delay Time 2	t_{RDD2}	0.55	1.0	—	μs
Read Data Pulse Width, High	t_{RDH}	—	1.0	—	μs
Read Data Pulse Width, Low	t_{RDL}	—	1.0	—	μs
Index Pulse Width, High	PW_{IDX}	1.0	—	—	μs
Transfer Request Release Time	t_{TR}	—	—	450	ns
Interrupt Request Release Time	t_{IR}	—	—	1.2	μs
Bus Direction Delay Time	t_{DBD}	—	—	330	ns
Write Data Pulse Width, High (f_c = 1.0 MHz)	PW_{WD}	—	1.0	—	μs
Write Data Cycle Time (f_c = 1.0 MHz)	t_{cycWD}	—	2.0	—	μs
Step Pulse Width, High (f_c = 1.0 MHz)	PW_{STP}	—	32	—	μs
Step Cycle Time* (f_c = 1.0 MHz)	t_{cycSTP}	1.0	—	15	ms
Write Gate to Write Data Delay (SSW, SWD, MSW)	t_{GD1}	0.7	1.0	1.3	μs
Write Gate Hold Time	t_{GH}	0	—	0.3	μs
Write Gate to Write Data Delay (FFW)	t_{GD2}	0.2	—	2.0	μs
CLK to $\overline{IRQ}$ Delay	t_{IRQC}	—	—	1.2	μs
CLK TO ISR0-3 Delay	t_{ISRD}	—	—	0.7	μs
Index Pulse to STRB Bit 3 Delay	t_{IRQI}	—	—	1.8	μs
Index Pulse to $\overline{IRQ}$ Delay	t_{STRB3}	—	—	1.0	μs
Data Clock to Transfer Request Delay	t_{DTx}	400	—	700	ns
Signal Rise and Fall Times	t_r, t_f	—	—	25	ns

*Step (STP) cycle time is programmable.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

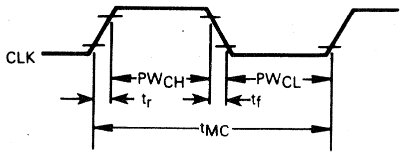
Ident. Number	Characteristic	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1.0	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	ns
9	Non-Muxed Address Hold Time	t_{AH}	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	ns
14	Chip-Select Setup Time Before E	t_{CS}	80	–	ns
15	Chip-Select Hold Time	t_{CSH}	10	–	ns
18	Peripheral Read Data Hold Time Provided	t_{DHR}	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	ns

FIGURE 4 — BUS TIMING CHARACTERISTICS (See Notes 1 and 2)



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

FIGURE 5 — MASTER CLOCK (CLK)



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 6 — READ DATA TIMING

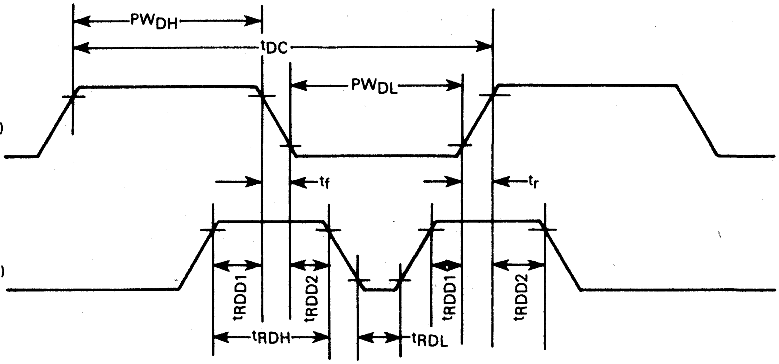


FIGURE 7 — INDEX TIMING

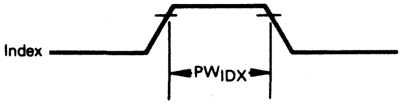


FIGURE 8 — IRQ RELEASE TIME

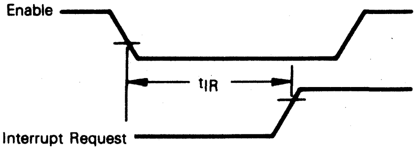


FIGURE 9 — DATA BUS DIRECTION TIMING

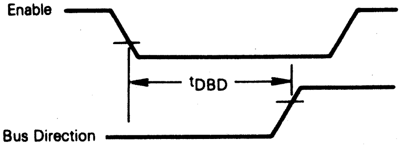
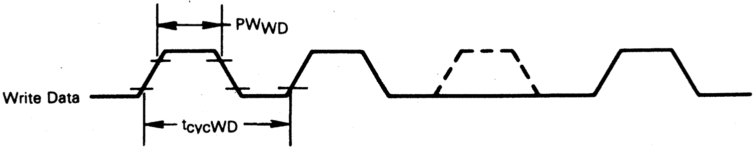


FIGURE 10 — WRITE DATA TIMING



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 11 — STEP TIMING (PROGRAMMABLE)

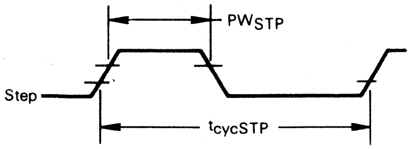


FIGURE 12 — TxRQ RELEASE TIMING

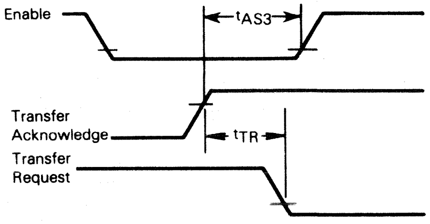


FIGURE 13 — DELAY TIME FROM DATA CLOCK TO TRANSFER REQUEST (t_{DTx})

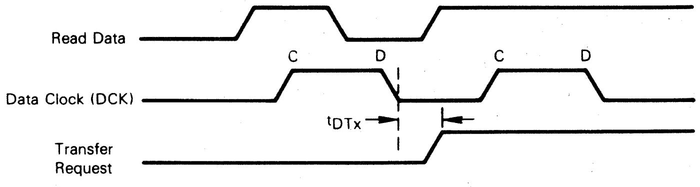
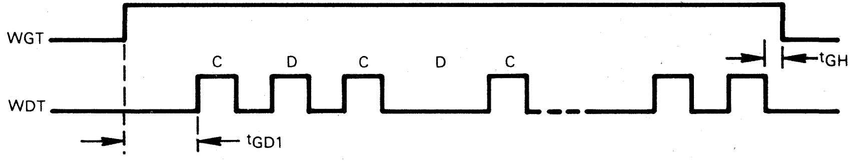


FIGURE 14 — WRITE DATA versus WRITE GATE TIMING

a — SSW, SWD and MSW commands (Single Sector Write, Single Sector Write with Deleted Address Mark, and Multiple Sector Write)



b — FFW Command (Free Format Write)

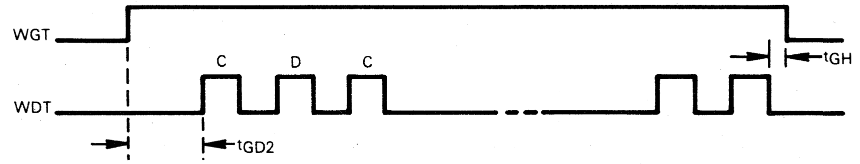
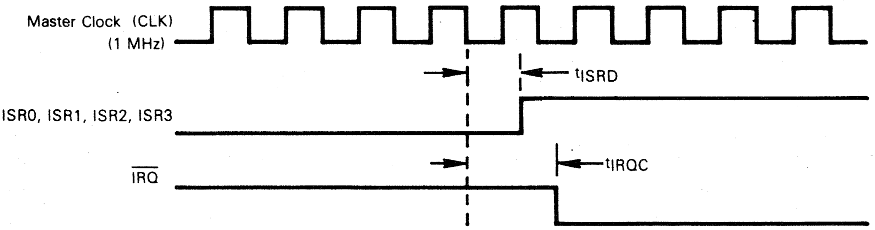
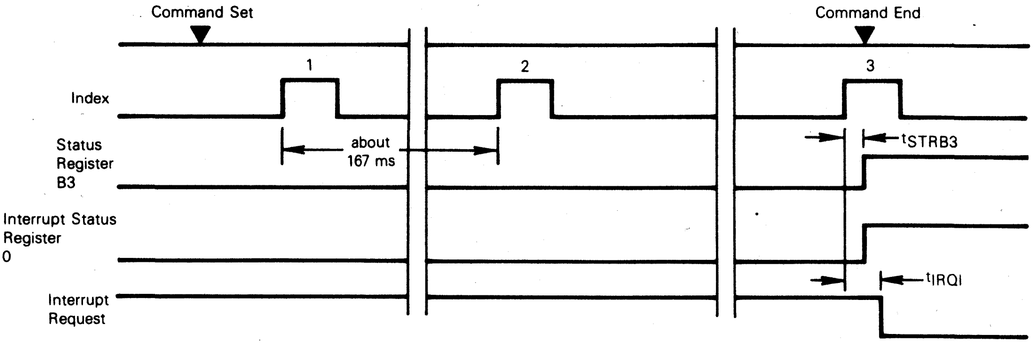


FIGURE 15 — INTERRUPT STATUS REGISTER AND INTERRUPT REQUEST TIMING



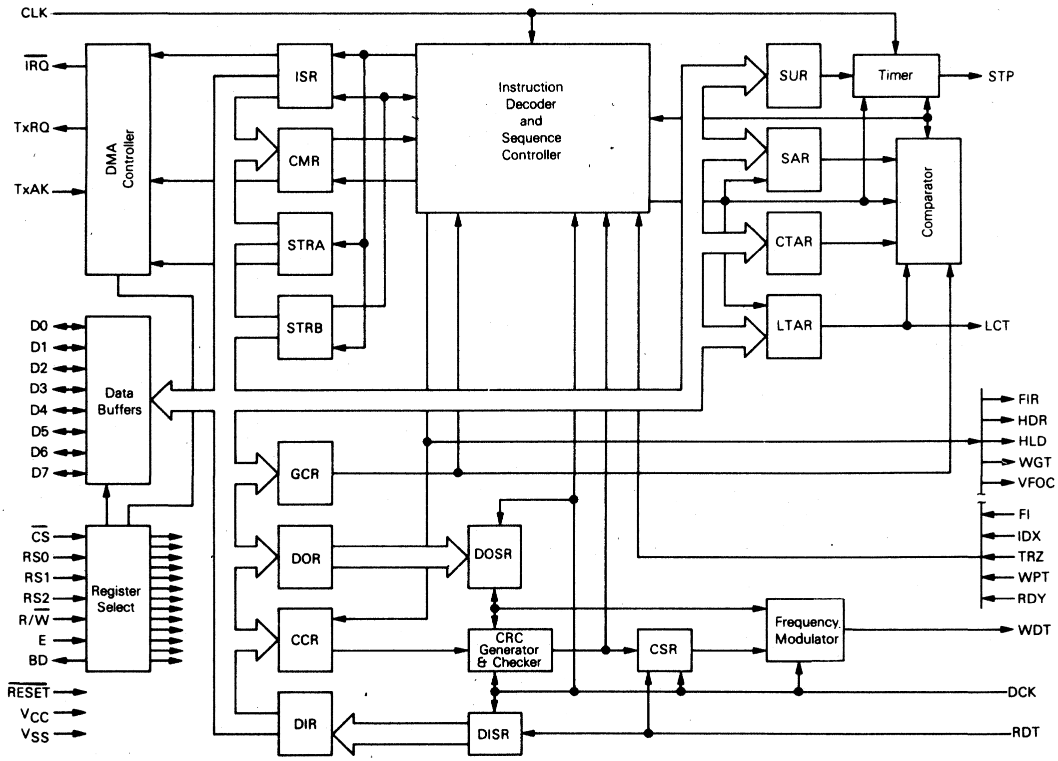
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

FIGURE 16 — TRACK NOT EQUAL ERROR TIMING



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 17 — INTERNAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The MC6843 FDC is a single-density controller which is IBM compatible. Data from the drive is clocked into the FDC by an external phase lock loop oscillator. Internal synchronization to the data stream is handled automatically. A 1 MHz clock is used as a timing signal for the internal functions of the FDC, such as head load and step, as well as shifting data serially to the drive. Status bits are provided to indicate various error conditions and status of the drive. DMA or polled I/O modes are available.

Register Section

The register section consists of twelve user-accessible registers used for controlling a floppy disk drive. All twelve are connected by the internal data bus to allow the processor access to them.

Data Output Register (DOR) – The DOR is an 8-bit register which holds the data to be written onto the disk. The information is stored here by the bus interface.

Data Input Register (DIR) – The data words read from the disk are stored in the 8-bit DIR until read by the bus interface.

Current-Track Address Register (CTAR) – CTAR is a 7-bit register containing the address of the track over which the R/W head is currently positioned.

Command Register (CMR) – The macro commands are written to the 8-bit CMR to begin their execution.

Interrupt Status Register (ISR) – The four bits of the ISR represent the four conditions that can cause an interrupt to occur.

Set-Up Register (SUR) – Variable Seek and Settling times are programmed by the SUR. Four bits are used to program the track-to-track seek time and four bits are used to program the head settling time for the floppy disk drive used with the FDC.

Status Register A (STRA) – The eight bits of STRA are used to indicate the state of the floppy disk interface.

Sector Address Register (SAR) – SAR contains the 5-bit sector address associated with the current data transfer.

Status Register B (STRB) – The eight error flags of STRB are used to signify error conditions detected by the FDC or generated by the floppy disk drive.

General Count Register (GCR) – The seven bits of GCR contain the destination track address when a SEK (seek) macro command is being executed. If a multi-sector Read or Write macro command is being executed, GCR contains the number of sectors to be read or written.

CRC Control Register (CCR) – The two bits of the CCR are used to enable the CRC and shift the CRC for the Free-Format Commands.

Logical-Track Address Register (LTAR) – The 7-bit track address used for read and write operations is stored in the LTAR by the bus interface.

Serializing Section

The serializing section handles the serial-to-parallel and parallel-to-serial conversions for Read/Write operations, as well as CRC generation/checking and the generation/detection of the clock pattern. The Data Output Shift Register (DOSR), Data Input Shift Register (DISR), CRC Generator/Checker, and Clock Shift Register (CSR) comprise the serializing section of the FDC.

Bus Interface

The Bus Interface section provides the timing and control logic that allows the FDC to operate with the M6800 bus, and is comprised of the Data Buffers, DMA Control, and the Register Select circuitry.

Control

The internal timing and control signals which sequence the FDC are derived from the macro instructions by the control section.

PIN DESCRIPTION

POWER PINS

V_{CC}: Input

+5 volt (±5%) power input.

V_{SS}: Input

Power Supply Ground.

BUS PINS

Reset Input – The $\overline{\text{RESET}}$ input is used to initialize the FDC. When $\overline{\text{RESET}}$ becomes Low, the state of the outputs is defined by the table below:

Output	State of Output	Output	State of Output
FIR	Low	HLD	Low
WGT	Low	TxRQ	Low
HDR	Low	IRQ	High
STP	Low	WDT	Low

Registers which are affected by $\overline{\text{RESET}}$ are shown in Table 7.

Interrupt Request ($\overline{\text{IRQ}}$) Output – The $\overline{\text{IRQ}}$ line is an open-drain output that becomes a low level when the FDC requests an interrupt. Interrupt requests are controlled by the interrupt enables in CMR (Command Register) with the function causing the interrupt shown in ISR (Interrupt Status Register).

Data Bus 0-Data Bus 7 (D0-D7) Bidirectional – The eight bidirectional data lines allow the transfer of data between the FDC and the controlling system. The output buffers are three-state drivers that are enabled when the FDC is transferring data to the data bus.

Enable (E) Input — The E input to the FDC causes data transfers to occur between the FDC and the system controlling the FDC (MC6800 MPU, DMA Controller, etc.). E must be a logic '1' (high level) for any transfer to be enabled on D0-D7. The E input is normally connected to system $\phi 2$.

Chip-Select (CS) Input — The $\overline{CS}$ input, in conjunction with the E input, is used to enable data transfers on D0-D7. E must be a high level and $\overline{CS}$ must be a low level to enable the transfer. The TxAK input being a high level (logic '1') performs a function similar to $\overline{CS}$ being a low level.

Read/Write (R/$\overline{W}$) Input — The R/$\overline{W}$ input is issued by the system controlling the FDC (MC6800 MPU, DMA Controller, etc.) to signify if a read or write operation is to be performed on the FDC. When TxAK is a low level, R/$\overline{W}$ is used in conjunction with $\overline{CS}$ and RS0-RS2 to determine which register is accessed by the bus as shown in Table 1. When TxAK is a high level, R/$\overline{W}$ is used to select either the DOR or DIR to the data bus (see description of TxAK input).

Register Select 0-Register Select 2 (RS0-RS2) Input — RS0-RS2, in conjunction with the R/$\overline{W}$ input, are used to select one of the user accessible registers in the FDC as shown in Table 1.

TABLE 1 — ADDRESS CODES FOR USER ACCESSIBLE REGISTERS

TxAK	RS2	RS1	RS0	R/$\overline{W}$	Registers
0	0	0	0	0	DOR (Data Out Register)
				1	DIR (Data In Register)
0	0	0	1	1/0	CTAR (Current Track Address Register)
0	0	1	0	0	CMR (Command Register)
				1	ISR (Interrupt Status Register)
0	0	1	1	0	SUR (Set Up Register)
				1	STRA (Status Register A)
0	1	0	0	0	SAR (Sector Address Register)
				1	STRB (Status Register B)
0	1	0	1	0	GCR (General Count Register)
0	1	1	0	0	CCR (CRC Control Register)
0	1	1	1	0	LTAR (Logical Track Address Register)

Transfer Request (TxRQ) Output — TxRQ is used in the DMA mode to request a data transfer by the DMAC. TxRQ is a high level if the FDC is in the DMA mode (CMR bit 5 is set) when a data transfer request occurs (STRA bit 1 is set). It is reset to a low level (logic '0') when TxAK becomes a high level (logic '1'). Data transfer errors will occur if TxAK does not reset TxRQ before the next data transfer is required.

Transfer Acknowledge (TxAK) Input — TxAK is generated by the system controlling the FDC (MC6800 MPU, DMA Controller, etc.) and is a response to a TxRQ issued by the FDC. A high level (logic '1') on TxAK causes the FDC to neglect the state of RS0-RS2 and $\overline{CS}$ causing the FDC to select the DOR (Data Output Register) or DIR (Data Input Register) to the data bus (D0-D7) as shown in Table 2.

TABLE 2 — REGISTER SELECTION FOR DMA TRANSFERS

TxAK	RS0-RS2	$\overline{CS}$	R/$\overline{W}$	Register Selected
1	X	X	1	DOR
1	X	X	0	DIR

This mode of operation is normally used for DMA (Direct Memory Access) transfer with the FDC.

When TxAK is a low level the registers are selected by $\overline{CS}$, R/$\overline{W}$ and RS0-RS2 as shown in Table 1.

Bus Direction (BD) Output — The BD output is provided to control bidirectional buffers on the data bus (D0-D7) as shown in Figure 1. Its polarity is shown by Table 3.

TABLE 3 — BUS DIRECTION (BD) STATES

TxAK	$\overline{CS}$	BD
1	X	$\overline{R/\overline{W}}$
0	1	0
0	0	R/$\overline{W}$

(Operation of BD, as defined by this chart, allows the FDC to function with the DMA Controller MC6844).

I/O AND CONTROL PINS

Master Clock (CLK) Input — The CLK input is used to generate various timing sequences internal to the FDC. The head settling and seek time, as well as the data and data clock timing, are generated from the CLK input signal.

Head Load (HLD) Output — HLD is used to notify the disk drive that the R/W head should be loaded (placed in contact with the media). When the FDC is ready for the head to load, HLD is a high level (logic '1'). A low level (logic '0') on HLD indicates the head should be unloaded.

Step (STP) Output — The STP output, in conjunction with HDR, is used to control head movement. A 32 μs wide positive (logic '1') pulse is generated on STP, to move the R/W head one track in the direction defined by the HDR output. The period of the STP signal is programmable by the SUR (Set-Up Register). The number of pulses generated on STP is the difference between the contents of the CTAR (Current Track Address Register) and the GCR (General Count Register) which contains the track address to which the head is to be moved.

Head Direction (HDR) Output — The HDR signal controls the direction of head movement. A high level (logic '1') signifies the head should step to the inside (toward the hub) of the disk. A low level (logic '0') indicates the direction of head movement should be to the outside of the disk.

Low-Current Track (LCT) Output — The LCT signal is used to control the level of write current used by the disk drive. LCT is a low level (logic '0') when the write head is positioned over tracks 0-43. If it is over tracks 44-76, LCT is a high level (logic '1'). LCT is determined from the contents of the Current Track Address Register (CTAR).

Write Gate (WGT) Output — When a write operation is being performed, WGT is a logic '1' (high level). For a read operation, WGT is a low level (logic '0').

File-Inoperable Reset (FIR) Output — FIR is an output from the FDC to the floppy disk drive to reset it from an inoperable status. If the FI input is a '1', a 1 μs pulse is generated on the FIR output whenever Status Register B is read.

File Inoperable (FI) Input — FI is an input to the FDC from the drive. A high level indicates the drive is in an inoperable state. Its current state can be examined by reading bit 5 of Status Register B (STRB).

Track Zero (TRZ) Input — The TRZ input is reflected by bit 3 of STRA (Status Register A). The TRZ input must be a high level (logic '1') when the R/W head of the drive is positioned over track zero. A logic '1' on this input inhibits step pulses during a Seek Track Zero command.

Index (IDX) Input — The index input is received from the floppy disk drive and is used to sense the index hold in the disk media. The IDX signal is used to initialize the internal FDC timing. The state of the IDX input is reflected by bit 6 of Status Register A (STRA). A high level (logic '1') is to indicate the index hole is under the index sensor. The index input is used to count the number of disk revolutions while searching for the address ID field (see description of STRB bit 3).

Ready (RDY) Input — The ready input is received from the disk drive and can be read as bit 2 of STRA (Status Register A). A high level (logic '1') indicates the drive is ready and allows the FDC to operate the drive.

Write Protect (WPT) Input — WPT is an input indicating when the media is Write Protected. A high level during an FDC write operation results in a Write Error (STRB bit 6) but the FDC continues to perform the write function. The state of the WPT input can be read by examining bit 4 of the Status Register A (STRA).

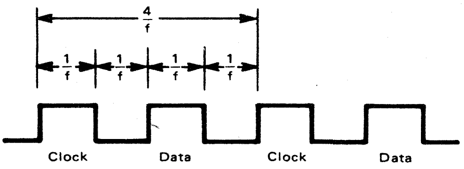
DATA PINS

Data Clock (DCK) Input — Data from the drive is clocked into the FDC on both positive and negative edges of the DCK input. This signal is generated from the Read Recovery Circuit.

Read-Data (RDT) Input — RDT is the serial data input from the Read Recovery Circuit. The data stream includes both the clock and the data bits and must be presynchronized to the Data Clock (DCK).

Write-Data (WDT) Output — WDT is the double frequency modulated data output from the FDC. The time between clock bits is 4/f where f is the frequency of the CLK input. The pulse width for both clock and data is 1/f (see Figure 18). For the normal CLK frequency of 1 MHz the write period is 4 μs, the clock pulse width is 1 μs and the data pulse width is 1 μs. Figure 18 shows the relationship between the WDT output and the frequency of the CLK inputs.

FIGURE 18 — WDT OUTPUT TIMING



f = Frequency of the CLK Input. To insure IBM3740 compatibility the clock frequency must be 1 MHz.

Variable-Frequency Oscillator Control (VFOC) Output — VFOC is used as a sync signal during system diagnostics. Waveforms are shown in Figure 19.

FORMAT

The format used by the MC6843, shown in Figure 20, is compatible with the soft sector format of the IBM 3740.

FIGURE 19 — VARIABLE FREQUENCY OSCILLATOR CONTROL WAVEFORM
(Relation Between WGT and VFOC)

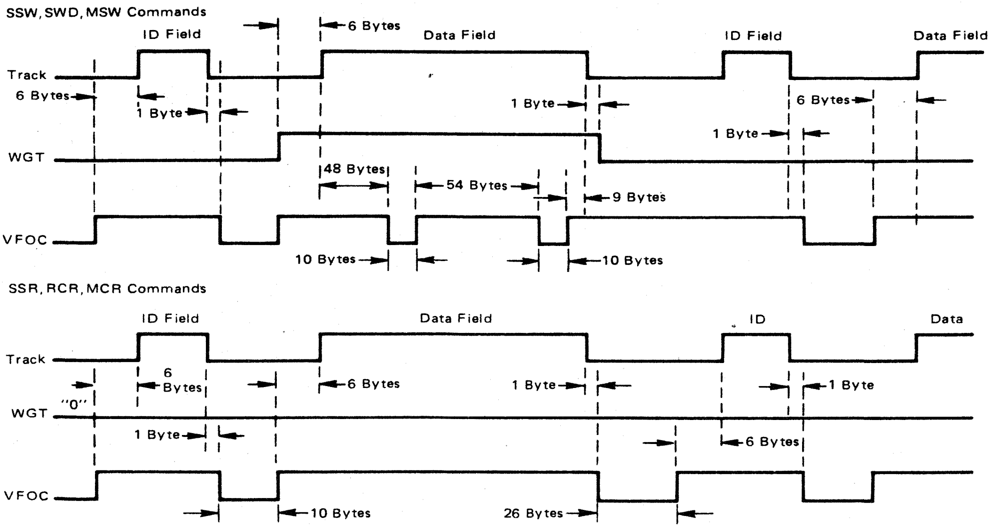
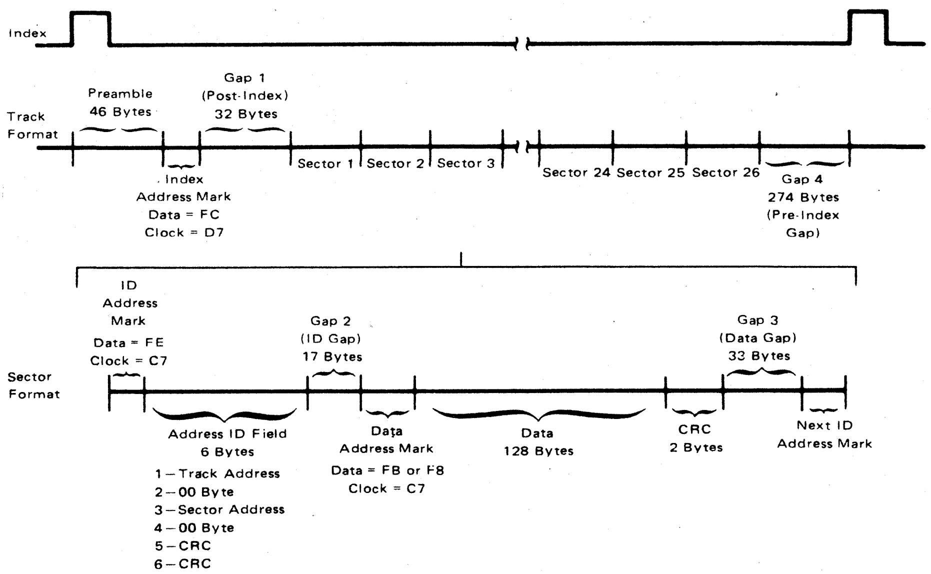


FIGURE 20 — SOFT SECTOR FORMAT



MACRO COMMAND SET

The macro command set shown in Table 4 is discussed in the following paragraphs.

Seek Track Zero (STZ)

The STZ command causes the R/W head to be released from the surface of the disk (HLD is reset) and positioned above track 00. The FDC issues step pulses on the STP output until the TRZ input becomes a high level or until 83 pulses have been sent to the drive. When the TRZ input becomes high, the step pulses are inhibited on the STP output but the FDC remains busy until all 83 have been generated internally.

If the TRZ input remains low (logic '0') after all 83 pulses have been generated, the Seek Error flag (STRB bit 4) is set.

After all 83 pulses have been generated, the head is loaded (HLD becomes a '1'). After the settling time specified in the SUR has expired, the Settling Time Complete flag is set (ISR bit 1), Busy (STRA-7) is reset, and CTAR and GCR are cleared. The head remains in contact with the disk. A command such as RCR (Read CRC) may be issued following a STZ if the head must be released.

Seek (SEK)

The SEK command is used to position the R/W head over the track on which a Read/Write operation is to be performed. The contents of the GCR are taken as the destination address and the contents of the CTAR is the source address; therefore, the number of pulses (N) on the STP output are given by:

$$N = |(CTAR) - (GCR)|$$

HDR is a '1' for (GCR) > (CTAR) otherwise it is a '0'.

When a SEK command is issued, Busy is set, the head is raised from the disk, HDR is set, and N number of pulses ap-

TABLE 4 — MACRO COMMAND SET

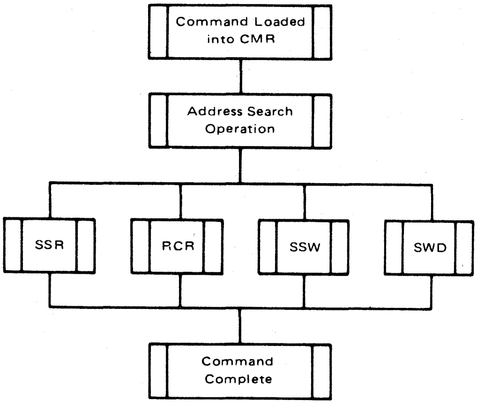
			CMR Bits				Hex
			Bit 3	Bit 2	Bit 1	Bit 0	Code
1	STZ	Seek Track Zero	0	0	1	0	2
2	SEK	Seek	0	0	1	1	3
3	SSR	Single Sector Read	0	1	0	0	4
4	SSW	Single Sector Write	0	1	0	1	5
5	RCR	Read CRC	0	1	1	0	6
6	SWD	Single Sector Write with Delete Data Mark	0	1	1	1	7
7	MSW	Multi Sector Write	1	1	0	1	D
8	MSR	Multi Sector Read	1	1	0	0	C
9	FFW	Free Format Write	1	0	1	1	B
10	FFR	Free Format Read	1	0	1	0	A

number of pulses appear on the STP output. After the last step pulse is used, the head is placed in contact with the disk. Once the head settling time has expired, the Settling Time complete flag (ISR bit 1) is set, Busy is reset, and the contents of the GCR are transferred to the CTAR.

SINGLE-SECTOR READ/WRITE COMMANDS

The single-sector Read/Write commands (SSR, RCR, SSW, and SWD) are used to Read/Write data from a single 128 byte sector on the disk. As shown in Figure 21 these types of instructions can be divided into two sections. The first section, which is common to all instructions, is the address search operation, while the second section is unique to the requirements of each instruction.

FIGURE 21 — BASIC SINGLE SECTOR COMMAND FLOW CHART



Address Search Operation

The flow chart of Figure 22 shows the operation of the address search.

Single-Sector Read (SSR)

The single-sector read command follows the address search procedure as defined in the previous flowchart. If the search is successful, status sense request is set and the operation continues as described by the flowchart of Figure 23.

Read CRC (RCR)

The RCR command is used to verify that correct data was written on a disk. The operation is the same as for the SSR command with the exception that the data-transfer request (STRA bit 0) is not set. The SSR interrupt can be disabled by using the DMA mode.

Single-Sector Write (SSW)

Single-sector write is used to write 128 bytes of data on the disk. After the command is issued, the address search is performed. The remainder of the instruction's operation is shown in Figure 24.

Single-Sector Write with Delete-Data Mark (SWD)

The operational flow of SWD is exactly like that of SSW. For SWD, the data pattern of the Data-Address Mark becomes F8 instead of FB. The clock pattern remains C7.

Multi-Sector Commands (MSR/MSW)

MSR is used for sequential reading of two or more sectors. If S sectors are to be read, S1 must be written into the GCR before the command is issued.

The basic operation for the MSR and MSW is the same as that for the SSR and SSW repsectively. The basic operation begins with an address search operation, which is followed by a single-sector read or write operation. This completes the operation on the first sector. The SAR is incremented, the GCR is decremented, and if no overflow is detected from the GCR (i.e., GCR becomes negative) the sequence is repeated until S number of sectors are read or written.

The completion of an MSR or MSW is like that of an SSR or SSW command. First MCC is set, after the settling time has expired, Busy is reset, and the head is released.

If a delete-data mark is detected during an MSR command, STRA bit 1 (Delete-Data Mark Detected) remains set throughout the commands operation.

When a multi-sector instruction is issued, the sum of the SAR and GCR must be less than 27. If SAR + GCR > 26, an address error (STRB bit 3 set) will occur after the contents of SAR becomes greater than 26.

Free-Format Write (FFW)

The FFW has two modes of operation which are selected by FWF (Free-Format Write Flag) which is data bit 4 of the CMR.

When the FWF = '0', the data bits of the DOR are written directly to the disk without first writing the preamble, address mark, etc. The contents of the DOR are FM modulated with a clock pattern of all ones.

If FWF = '1' the odd bits of the DOR are used as clock bits and even bits are used for data bits. In this mode, the DOSR clock is twice a normal write operation and one byte of DOR is one nibble (four bits of data) on the disk.

The two modes of the FFW command allow formatting a disk with either the IBM 3470 format or a user defined format.

After the FFW command is loaded into the CMR, WGT becomes a high level, the contents of DOR are transferred to the DOSR, data transfer request (STRA bit 0) is set, and the serial bit pattern is shifted out on the WDT line. Therefore, DOR must be loaded before the FFW command is issued. Data from the DOR is continually transferred to the DOSR and shifted out on WDT until the CMR has been written with an all zero pattern. When CMR becomes zero, WGT becomes a low level, but MCC is not set and the R/W head is left in contact with the disk.

Free-Format Read (FFR)

FFR is used to input all data (including Address Marks) from a disk. Once the FFR command is set into the CMR, the head is loaded and after the settling time has expired the serial data from the FDC is brought into the DISR. After 8 bits have accumulated, it is transferred to the DIR and Data-Transfer Request (STRA bit 0) is set.

FIGURE 22 — OPERATIONAL FLOW OF THE ADDRESS SEARCH SEQUENCE

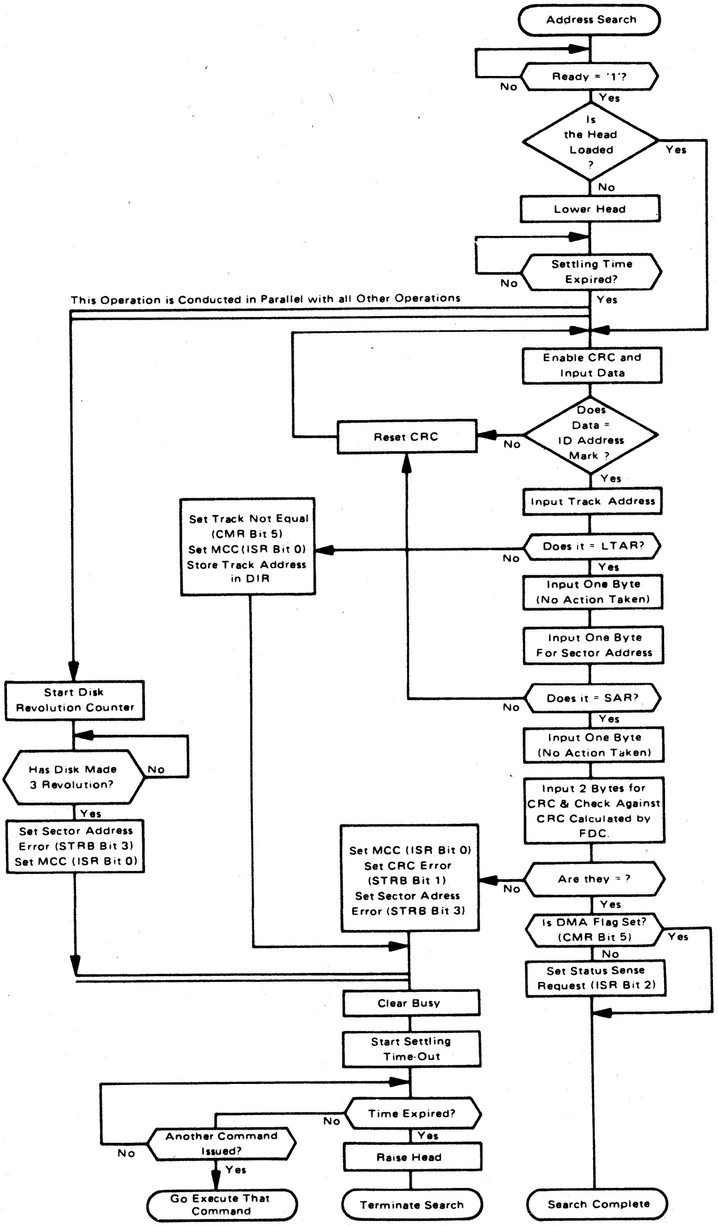


FIGURE 23 — OPERATIONAL FLOW OF THE SSR COMMAND

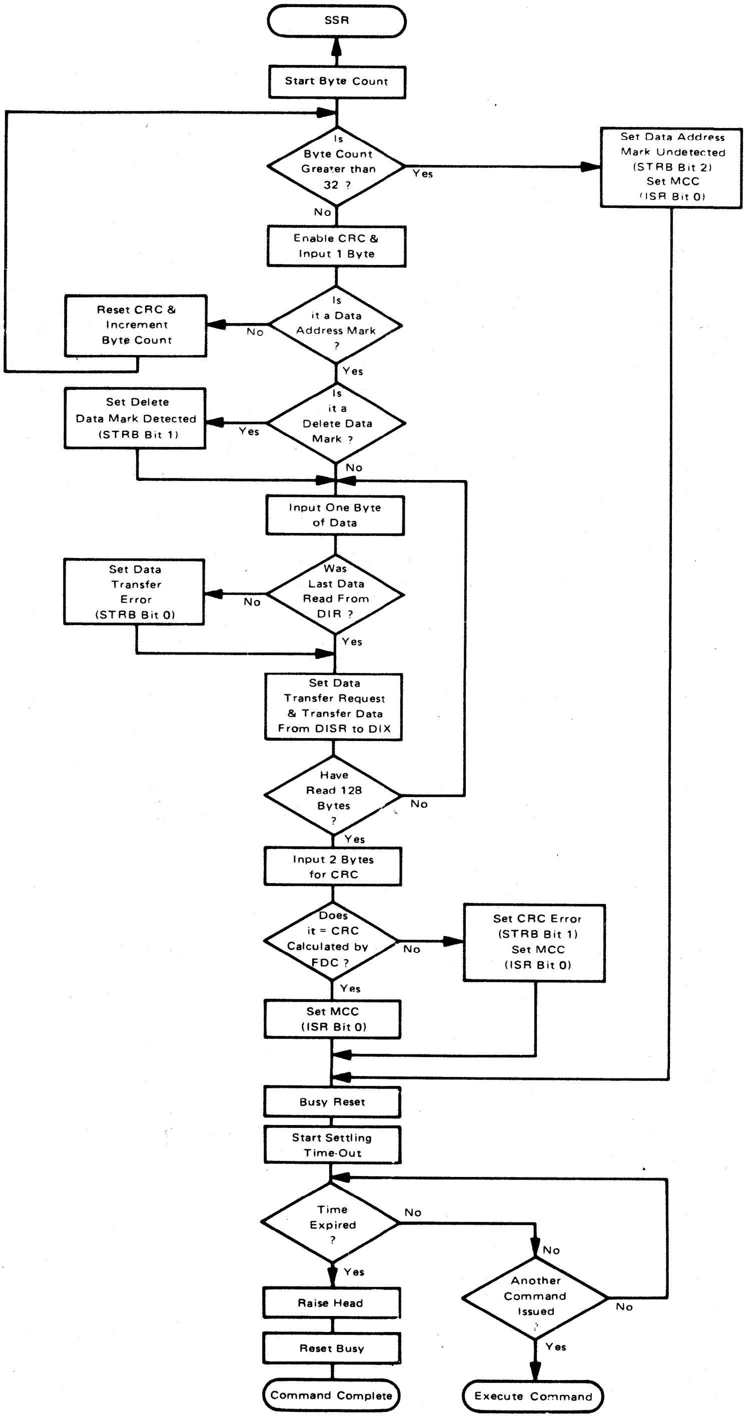
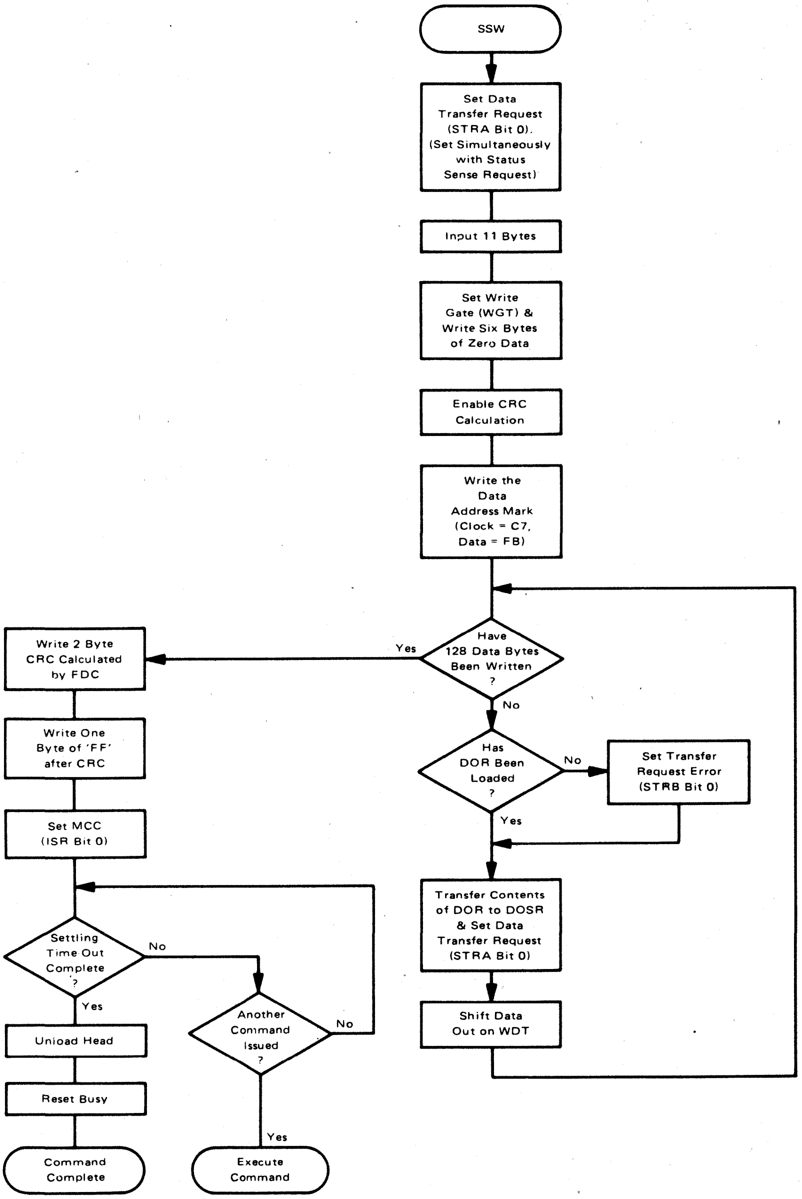


FIGURE 24 — OPERATIONAL FLOW OF THE SSW COMMAND



This operation continues until a zero pattern is stored in the CMR, terminating the FFR command. As in the case of the FFW command, MCC is not set and the head remains in contact with the disk.

The first data that enters the DISR is not necessarily the first bit of a data word since the head may be lowered at any place on the disk. To prevent the FDC from remaining unsynchronized to the data, the FFR command will synchronize to either an ID address mark (FE) or a Data-Address Mark (FB or F8).

REGISTER DEFINITIONS

DATA OUTPUT REGISTER (DOR)

Hex address 0, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8 Bits of Data Used for a Disk Write Operation							

When one of the four write macro commands (SSW, SWD, MSW, and FFW) is executed, the information contained in the DOR is loaded into the DOSR, and is shifted out on the WDT line using a double frequency (FM) format.

DATA INPUT REGISTER (DIR)

Hex address 0, read only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8 Bits of Data Used for a Disk Read Operation							

One of the three read macro commands (SSR, MSR, FFR) executed, will cause the information on the RDT input to be clocked into the DISR. When eight clock pulses have occurred, the eight bits of information in the DISR are transferred to the DIR where it can be read by the bus interface.

CURRENT TRACK ADDRESS (CTAR)

Hex address 1, read/write

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used	7 Bit Track Address of Current Head Position						

The address of the track over which the R/W head is currently positioned is contained in the CTAR. At the end of a SEK command, the contents of the GCR are transferred to the CTAR. CTAR is cleared at the completion of a STZ command. CTAR is a read/write register so that the head position can be updated when several drives are connected to one FDC. Bit 7 is read as a '0'.

COMMAND REGISTER (CMR)

Hex address 2, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3*	Bit 2*	Bit 1*	Bit 0*
Function Interrupt Mask	ISR3 Interrupt Mask	DMA Flag	FWF	Macro Command			

*Bits 0-4 are clared by $\overline{\text{RESET}}$.

The commands that control the FDC are loaded into the lower four bits of the CMR. Information that controls the data transfer mode and interrupt conditions are loaded into bits 4 through 7.

Bit 0-Bit 3: Macro Command

The Macro Command to be executed by the FDC is written to bits 0-3.

Bit 4: Free-Format Write Flag (FWF)

If a Free-Format Write command is issued, the state of bit 4 of the CMR determines what clock source will be used. The FWF is defined in the FFW (Free-Format Write) command explanation.

Bit 5: DMA Flag

If bit 5 is a '1' the FDC is in the DMA mode. Bit 5 being a '1' inhibits setting of Status Sense Request (ISR bit 2) thereby preventing its associated interrupt. A logic '1' DMA flag also enables the TxRQ output allowing it to request DMA transfers when the Data Transfer Request flag (STRA bit 0) is set.

A logic '0' DMA flag indicates the program controlled I/O (PC I/O) mode is to be used.

Bit 6: ISR3 Mask

CMR bit 6 (ISR3 Mask) is used to control the operation of ISR bit 3. A logic '1' in CMR bit 6 inhibits ISR bit 3 from being set when STRB becomes non-zero. If CMR bit 6 (ISR3 Mask) is a '0' the ISR bit 3 will be set if any bit in STRB becomes set. The setting of ISR bit 3 will cause an interrupt if CMR bit 7 is a '0'.

Bit 7: Function Interrupt Mask

When CMR bit 7 is a logic '1' all interrupts are inhibited except Status Sense Request (ISR bit 2) which can only be inhibited by the DMA flag (CMR bit 5). A logic '0' in CMR bit 7 enables interrupts from ISR0 (Macro Command Complete) and ISR1 (Settling Time Complete), and if the ISR3 Mask is '0', from ISR3.

TABLE 5

Interrupt Status Register (Bits Causing Interrupts)	Command Register Masks That Affect Interrupts		
	CMR7 (Function Interrupt Mask)	**CMR6 (ISR3 Mask)**	**CMR5 (DMA Flag)**
ISR0 (Macro Command Complete)	M	X	X
ISR1 (Settling Time Complete)	M	X	X
ISR2 (Status Sense Request)	X	X	M
ISR3 (STRB Conditions)	M	M	X

X = No effect
M = Bits that are used as masks

INTERRUPT STATUS REGISTER (ISR)

Hex address 2, read only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used (Read as '0')				STRB*	Status* Sense Request	Settling* Time Complete	Macro* Command

*Cleared by $\overline{\text{RESET}}$

Bit 0: Macro Command Complete

When an SSR, RCR, SSW, SWD, MSR or MSW Macro Command has completed execution, bit 0 bcomes set (logic '1'). If the function interrupts are enabled (bit 7 of CMR is a logic '0'), the conclusion of a Macro command's execution will cause an interrupt.

Bit 1: Settling Time Complete

Settling time complete is set on SEK and STZ commands to indicate the head has been loaded and the settling time specified in SUR has expired. Since MCC is not set for the SEK or STZ command, settling time complete can be used as an interrupt to signify the SEK or STZ command has finished. Settling Time Complete is not set for any of the R/W commands.

Bit 2: Status Sense Request

For an SSR, SSW, SWD, MSR, or MSW Command, Status Sense Request indicates that the specified address ID field has been detected and verified by a CRC check. This is used as an early indication that data transfers will occur after 18 more byte times. For MSR and MSW commands, it is set for each sector.

In the PC I/O mode, an interrupt occurs when Status Sense Request becomes a logic '1' regardless of the state of the CMR interrupt mask. In the DMA mode, (DMA flag of CMR is set) Status Sense Request is unchanged and does not generate an interrupt when the address ID field has been verified.

Bit 3: STRB

STRB is an 'OR' of all of the bits of Status Register B and is disabled by the STRB interrupt mask in the CMR (CMR bit 6). The equation:

$$STRB = CMR6 \cdot (STRB + STRB1 + STRB2 + STRB3 + STRB4 + STRB5 + STRB6 + STRB7)$$

describes the operation of Bit 3 of the ISR.

ISR0, ISR1, and ISR2 are cleared when the Interrupt Status Register is read, but ISR3 is cleared only after Status Register B has been read.

SET-UP REGISTER (SUR)

Hex address 3, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Track to Track Seek Time				Head Settling Time			

The SUR is not affected by a reset operation; therefore, once it is initialized, the information remains until power is removed from the FDC.

Bit 0-Bit 3: Head Settling Time

The head settling time is used to generate a delay after the head is placed in contact with the disk. This allows the head to stop bouncing before any operations are performed. The delay is programmed by bits 0-3 and is specified by the equation:

$$\text{Delay} = \frac{4096}{f} \cdot B$$

B = Number contained in bits 0-3 of SUR

f = Frequency of CLK input

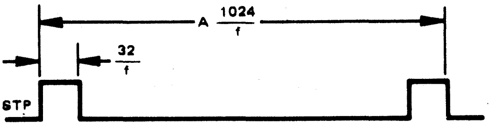
For IBM 3740 compatibility f = 1 MHz and the timing range is 4.096 ms for a '0001' to 61.44 ms for a '1111'. A '0000' code prevents Settling Time complete from being set and the FDC must be Reset.

Bit 4-Bit 7: Track-to-Track Seek Time

The frequency of STP is determined by bit 4-bit 7 of SUR as shown below. If the track-to-track seek time is 0 the period of STP is 64/f.

A = Number specified in bits 4-7 of SUR

f = Frequency of clock input



A = Number specified in bits 4-7 of SUR.
f = Frequency of clock input.

For IBM compatible operation, f is 1 MHz. This results in an STP pulse width of 32 µs and an STP interval of 1.024 ms for 0001 in bits 7-4 to 15.36 ms for 1111.

STATUS REGISTER A (STRA)

Hex address 3, read only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Busy*	Index	Track* Not Equal	Write Protect	Track Zero	Drive Ready	Delete* Data Mark Detected	Data* Transfer Request

*Cleared by $\overline{\text{RESET}}$

Bit 0: Data Transfer Request

For a write operation (SSW, SWD, MSW, FFW) the transfer request bit indicates that the DOR is ready to accept the next data word to be written on the disk. If data is not written into the DOR before the last data bit in the DOSR is shifted out to the WDT line, the data transfer error bit (bit 0 of STRB) will be set. After a write command has been issued, the first transfer request occurs simultaneously with the Status Sense Request. For a write operation, transfer request is reset after the DOR has been written from the data bus.

During a read operation (SSR, MSR, FFR) the transfer request bit signifies data from the DISR has been transferred to the DIR. The DIR must be read before the DISR is full again or the data transfer error bit (bit 0 of STRB) will be set. For read operations, transfer request is reset by a read of the DIR.

Bit 1: Delete Data Mark Detected

A Single-Sector Read operation that detects a delete data code (F8), instead of a general code (FB) as a Data Address Mark, will set the Delete Data Mark Detected bit. For the MSR command, bit 1 is set the first time an 'F8' code is found and remains set throughout the execution of the command. Bit 1 is reset whenever an SSR, SSW, SWD, MSR, MSW, or RCR command is issued.

Bit 2: Drive Ready

The Drive Ready bit indicates the state of the Ready input from the floppy disk drive. If a command is issued with Ready at logic '0', its execution will be inhibited until Ready becomes a logic '1'. If ready becomes a '0' during the execution of a command the Hard Error Flag (STRB bit 7) is set.

Bit 3: Track Zero

The state of the Track Zero input from the floppy disk drive is reflected in this bit of STRA. A logic '1' on the Track Zero input inhibits step pulses during an STZ command.

Bit 4: Write Protect

The Write Protect input from the floppy disk drive is reflected by bit 4 of STRA. A high level (logic '1') on the WPT input during the execution of any write command results in a write error (bit 6 of STRB set).

Bit 5: Track Not Equal

If the track address read from the address ID field does not coincide with the address in the LTAR, the Track Not Equal bit is set. Track Not Equal applies to all non-free format read/write commands, and is reset after a non-free format read/write command is issued.

Bit 6: Index

The state of the index input appears in bit 6 of STRA. The index input is used to count the number of disk revolutions while the FDC is looking for the address ID field (see operation of STRB bit 3) during the address search phase of a non-free format read/write command.

Bit 7: Busy

When Busy is a logic '1', the FDC is executing a command and no new commands can be issued.

SECTOR ADDRESS REGISTER (SAR)

Hex address 4, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used			5 Bit Sector Address				

Before a data transfer macro command (SSW, SWD, SSR, RCR, MSW, MSR) is issued, the address of the sector on which the operation is to be performed must be written into the SAR. The address in the sector address byte of an Address ID field of the disk is compared with the contents of the SAR. During an MSW or MSR command, the SAR is incremented after each sector is read or written. When execution is complete, the SAR contains the address of the last sector on which an operation was performed plus one. At the completion of an STZ or SEK command, SAR is cleared.

STATUS REGISTER B (STRB)

Hex address 4, read only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Hard* Error	Write* Error	File Inoperable	Seek* Error	Sector* Address Undetected	Data* Mark Undetected	CRC* Error	Data* Transfer Error

*Cleared by $\overline{\text{RESET}}$

The bits of the STRB represent possible error conditions that may occur during execution of macro commands. Whenever STRB is reset, ISR bit 3 is also reset.

Bit 0: Data Transfer Error

Data Transfer indicates an underflow or overflow of data. If a Write operation is being performed, it signifies that data was not presented to the DOR before the DOSR became empty. In this case, the current contents of the DOR are transferred to the DOSR and the write operation continues. The data transfer error remains set until data is written into the DOR. The operation of the CRC is unchanged.

For read commands, a data transfer error indicates that data in the DIR was not read before the next data word from the disk was transferred to the DIR. The read operation continues until sufficient data has been read from the disk to satisfy the requirements of the command (128 bytes for SSR). The error indication remains set until STRB is read, and the transfer request remains set until data is read from the DIR.

Bit 1: CRC Error

A CRC error occurs when the CRC read from the disk does not match that calculated by the FDC on the data it reads from the disk. A CRC error can occur in three different situations; checking the address ID field, checking the data field, and checking the FFR data. (See operation of CCR.)

If the CRC error occurs during the check of an address ID field, Sector Address Undetected (STRB bit 3) will also be indicated (see Table 6). A CRC error of a data field is indicated by a CRC error and no sector address error.

Bit 2: Data Mark Undetected

If a valid mark is not detected in the data block of a sector, it is indicated by a Data Mark Undetected error. Data Mark Undetected is reset after a non-free format Read/Write command is issued.

Bit 3: Sector Address Undetected

The sector address bit can be set on two conditions; not finding the sector address and a CRC error on an address ID field.

If the disk makes three revolutions during an address search operation and the sector address specified in the sector address register is not found in any of the address ID fields, a sector address undetected condition is indicated.

A CRC error that occurs on an address ID field will set bit 3 also. Table 6 shows how bits 1 and 3 are related.

TABLE 6 — RELATIONSHIP OF CRC ERROR AND SECTOR ADDRESS UNDETECTED

CRC Error (STRB 1)	Sector Address Undetected (STRB 3)	Condition
0	0	No Error
0	1	Sector Address not Detected
1	0	CRC Error on a Data Field
1	1	CRC Error on Address ID Field

Bit 4: Seek Error

An STZ (Seek Track Zero) command that never receives a track zero indication on the track zero input will result in a Seek Error (see description of STZ command).

Bit 5: File Inoperable

The state of the File Inoperable input appears in bit 5. If the File Inoperable input is a '1', a pulse of width 1/f (where f = Frequency of the clock input) is issued on the FIR output when STRB is read. FI is not latched but the input is gated to the bus when STRB is read.

Bit 6: Write Error

If the WPT input becomes a high level (logic '1') during the execution of a write command the write error bit is set.

Bit 7: Hard Error

If the Ready input becomes a '0' during the operation of a command (Busy is set), a hard error indication will result.

GENERAL COUNT REGISTER (GCR)

Hex address 5, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used	7 Bit Count for Track Number on SEK Command and Sector Count for MSR or MSW Command						

The GCR contains the destination track address for the R/$\overline{W}$ head on an SEK Macro Command. The contents of the GCR are transferred to the CTAR at the end of the SEK Command. For multi-sector read or write operations (MSR, MSW), the GCR contains the number of sectors to be read minus one. During the MSR or MSW execution the GCR is decremented after each sector is read or written.

CRC CONTROL REGISTER (CCR)

Hex address 6, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used						Shift CRC	CRC Enable

The CCR information is used only in the free format commands; for all other commands this register is masked and has no function.

Bit 0: CRC Enable

During an FFW command, CRC Enable is set by software and CRC generation takes effect on the next transfer of data from DOR to DOSR (see Figure 25). The CRC generation continues until Shift CRC (CCR bit 1) is set.

For an FFR command, CRC Enable is set by software and CRC generation takes effect on the next data read from DIR. The calculation continues for all data bytes read from DIR until CRC Enable is reset. The bytes read previous to resetting CRC Enable are considered the CRC information bytes and the CRC check is made against them.

Bit 1: Shift CRC

Bit 1 is valid only for the FFW command. After setting, it takes effect on the next transfer of data from DOR to DOSR (see Figure 26). Setting Shift CRC terminates the CRC calculation and causes the CRC calculated on all the data written into DOR up to the setting of bit 1, to be shifted out the WDT output. The CRC calculation will not include any data written to DOR after Shift CRC is set.

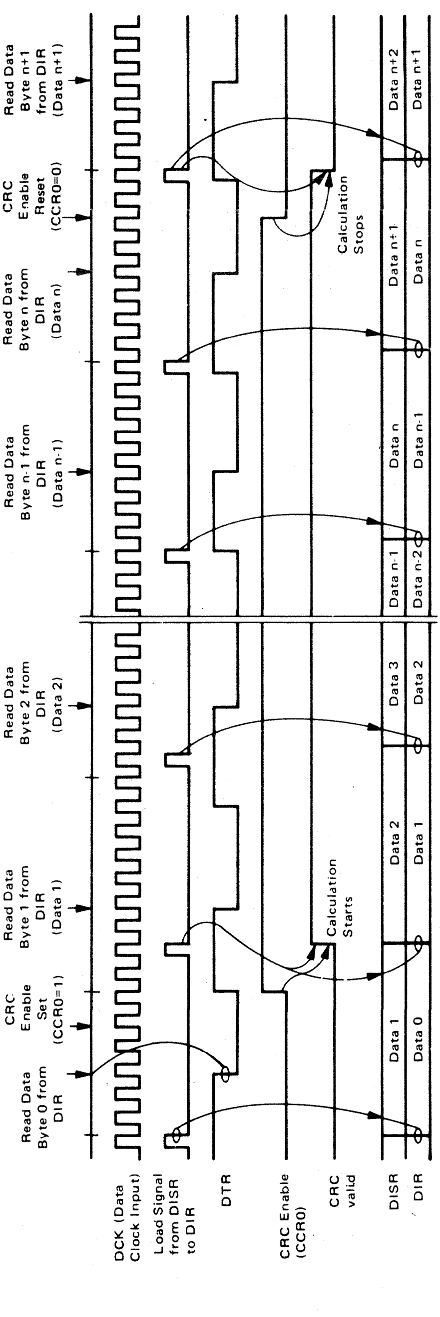
LTAR (LOGICAL TRACK ADDRESS)

Hex address 7, write only

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used	7 Bit Logical Track Address						

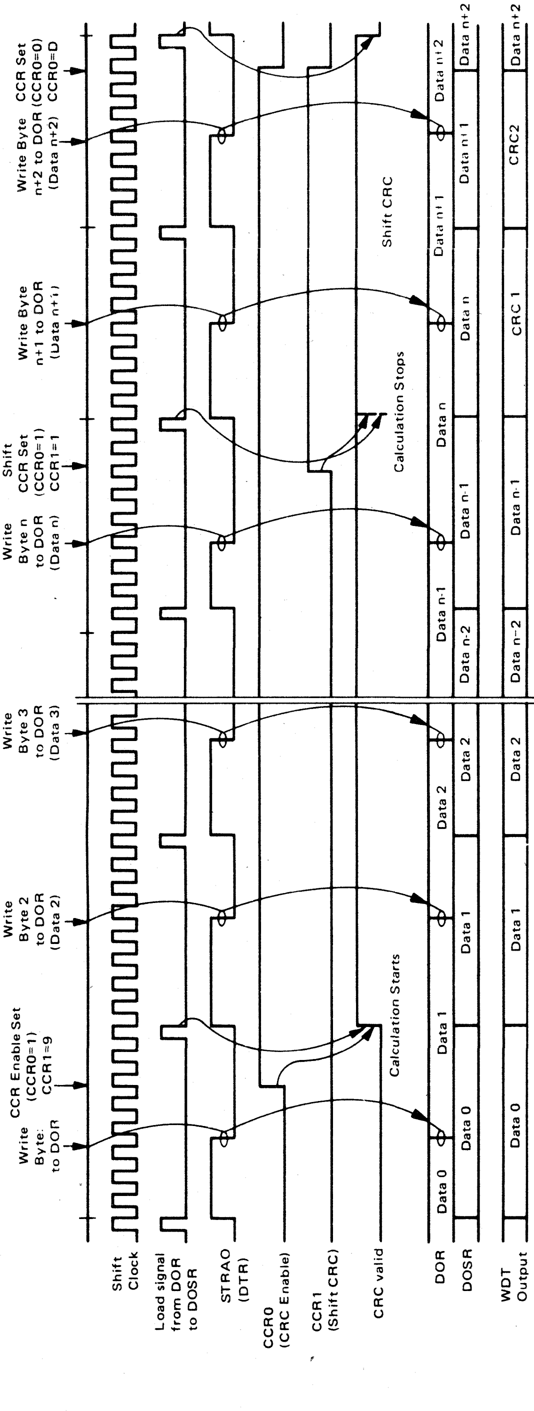
When a read or write macro command (SSW, SWD, SSR, RCR, MSW, MSR) is issued, the address of the track on which the operation is to be performed must be written into the LTAR. The address in the track address byte of an Address ID field of the disk is compared with the contents of the LTAR. The contents of LTAR are not affected by the execution of any of the commands.

FIGURE 25 — CCR CONTROL REGISTER TIMING FOR AN FFR COMMAND (READ)



CRC Calculation includes Data Byte 1 through Data Byte n.

FIGURE 26 — CCR CONTROL REGISTER TIMING FOR AN FFW COMMAND (WRITE)



The CRC Calculation includes Data Byte 1 through Data Byte n-1.

TABLE 7 — PROGRAMMING REFERENCE DATA

Table 7 is a summary of the information in the data sheet and can be used as a reference when programming the MC6843.

Registers	Hex Address	R/$\overline{W}$ Mode	Data Bits							
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DOR	0	WO	8 Bits of Data Used for a Disk Write Operation							
DIR	0	RO	8 Bits of Data Used for a Disk Read Operation							
CTAR	1	R/W	Not Used	7 Bit Track Address of Current Head Position						
CMR	2	WO	Function Interrupt Mask	ISR3 Interrupt Mask	DMA Flag	FWF	* Macro Command	*	*	*
ISR	2	RO	Not Used				STRB *	Status Sense Request *	Settling Time Complete *	Macro Command Complete *
SUR	3	WO	Track to Track Seek Time				Head Settling Time			
STRA	3	RO	Busy *	Index	Track Not Equal *	Write Protect	Track Zero	Drive Ready	Delete Data Mark Detected *	Data Transfer Request *
SAR	4	WO	Not Used			5 Bit Sector Address				
STRB	4	RO	Hard Error *	Write Error *	File Inoperable	Seek Error *	Sector Address Undetected *	Data Mark Undetected *	CRC Error *	Data Transfer *
GCR	5	WO	Not Used	7 Bit Count for Track Number on SEK or Sector Count for MSR or MSW.						
CCR	6	WO	Not Used						Shift CRC	CRC Enable
LTAR	7	WO	Not Used	7 Bit Search Track Address						

RO — Read Only
WO — Write Only
R/W — Read/Write

*Cleared by Reset

MACRO COMMANDS

Hex Code	Instruction	Hex Code	Instruction
2	STZ	A	FFR
3	SEK	B	FFW
4	SSR	C	MSR
5	SSW	D	MSW
6	RCR		
7	SWD		

TABLE 8

Table 8 is a list of all error flags showing what conditions will cause the error, the instructions for which they are valid, and what conditions reset them.

Name	Flag	Set Condition	Reset Condtiion	Command
Track Not Equal	STRA5	The track address that is read from the disk does not coincide with the content of the LTAR.	Upon issuance of SSW, SSR, SWD, RCR, MSR, MSW command	SSW, SSR, SWD, RCR, MSR, MSW
Data Transfer Error	STRB0	During the data transfer between the drive and the MPU or memory, an overrun or underflow occurs.	Reading of STRB	SSW, SSR, SWD, RCR, MSR, MSW, FFR, FFW
CRC Error	STRB1	In checking the CRC of an ID field or a Data field, a CRC error occurs.	Reading of STRB	SSW, SSR, SWD, RCR, MSR, MSW, (FFR)
Data Mark Undetected	STRB2	If data address mark (FB or F8) is not detected within 32 bytes after the address ID field has been detected.	Upon issuance of SSW, SSR, SWD, RCR, MSR, MSW Command	SSR, RCR, MSR
Sector Address Undetected	STRB3	(1) The sector address that coincides with the contents of the SAR does not exist on the track. (2) A CRC error occurs in checking the ID field.	Reading of STRB	SSW, SSR, SWD, RCR, MSR, MSW
Seek Error	STRB4	During a STZ command, the TKZ input remains low after 83 pulses have been issued on the STP output.	Reading of STRB	STZ
FI	STRB5	File Inoperable input is high.	Reading STRB causes the FIR output to go High. This should reset the FI input.	SSW, SWD, MSW, FFW
Write Error	STRB6	The WPT input is high, and a write operation is executed.	Reading of STRB with either WGT or WPT reset.	SSW, SWD, MSW, FFW
Hard Error	STRB7	During the execution of command (Busy is 1) the RDY input becomes low.	Reading of STRB	All commands

MOTOROLA

MC6844
(1.0 MHz)
MC68A44
(1.5 MHz)
MC68B44
(2.0 MHz)

DIRECT MEMORY ACCESS CONTROLLER (DMAC)

The MC6844 Direct Memory Access Controller (DMAC) performs the function of transferring data directly between memory and peripheral device controllers. It directly transfers the data by controlling the address and data bus in place of an MPU in a bus organized system.

The bus interface of the MC6844 includes select, read/write, interrupt, transfer request/grant, a data port, and an address port which allow data transfer over an 8-bit bidirectional data bus. The funtional configuration of the DMAC is programmed via the data bus. The internal structure provides for control and handling of four individual channels, each of which is separately configured. Programmable control registers provide control for data transfer location and data block length, individual channel control and transfer mode configuration, priority of channel servicing, data chaining, and interrupt control. Status and control lines provide control to peripheral controllers.

The mode of transfer for each channel can be programmed as one of two single-byte transfer modes or a burst transfer mode.

Typical MC6844 applications are a Floppy Disk Controller (FDC) and an Advanced Data Link Controller (ADLC) DMA interface.

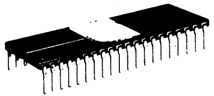
MC6844 features include:

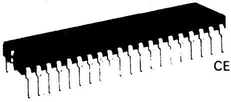
- Four DMA Channels, Each Having a 16-Bit Address Register and a 16-Bit Byte Count Register
- 2 M Byte/Sec Maximum Data Transfer Rate
- Selection of Fixed or Rotating Priority Service Control
- Separate Control Bits for Each Channel
- Data Chain Function
- Address Increment or Decrement Update
- Programmable Interrupts and DMA End to Peripheral Controllers

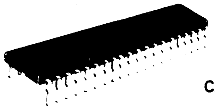
MOS

(N-CHANNEL, SILICON-GATE)

DIRECT MEMORY ACCESS CONTROLLER (DMAC)


CASE 715-04
(CERAMIC)


S SUFFIX
CERDIP PACKAGE
CASE 734


CASE 711-03
(PLASTIC)

PIN ASSIGNMENT

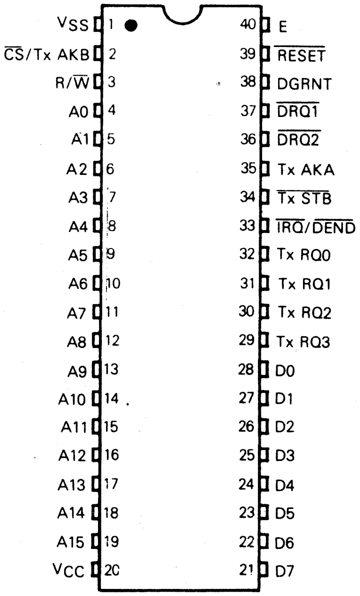


FIGURE 1 – M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM

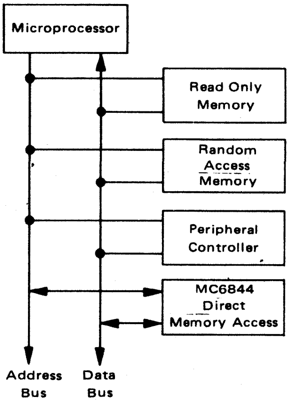
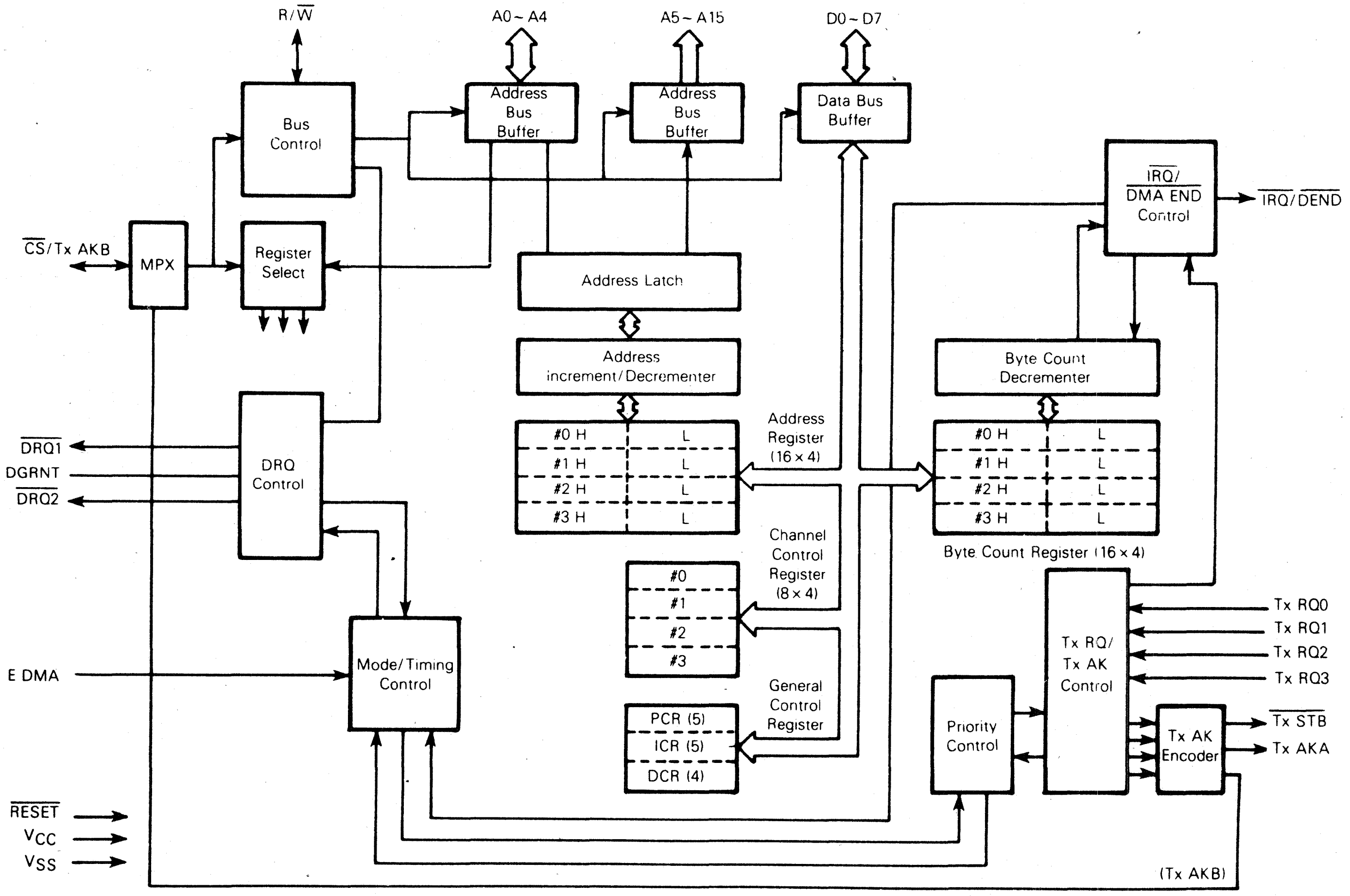


FIGURE 2 — BLOCK DIAGRAM OF DMAC



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 MC6844, MC68A44, MC68B44 MC6844C, MC68A44C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. 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 Ceramic Cerdip	θ_{JA}	 100 50 60	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

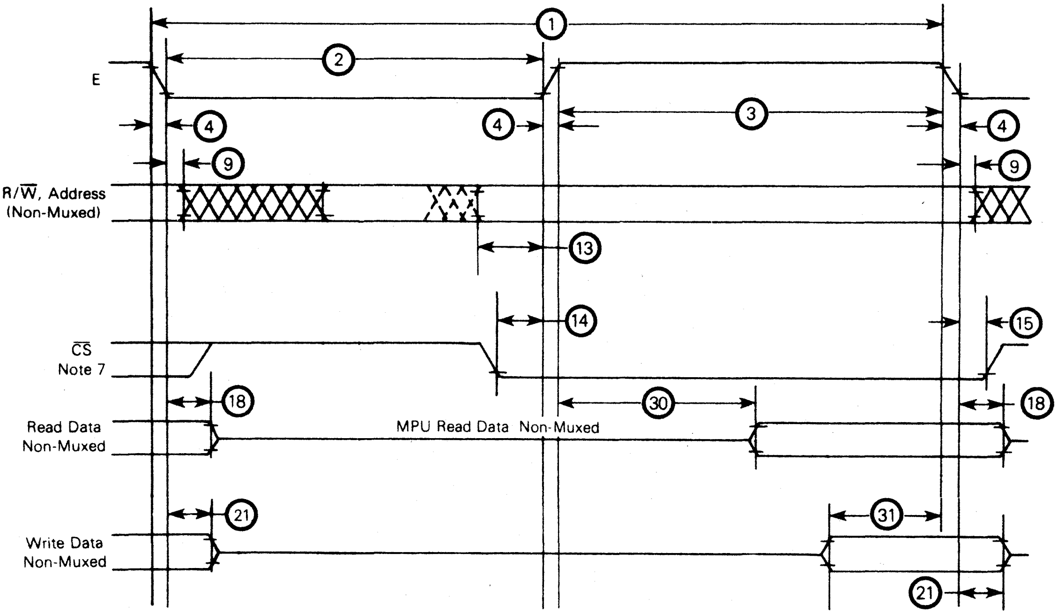
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$/Tx AKB Other Inputs	V_{IL}	$V_{SS}-0.3$ $V_{SS}-0.3$	– –	$V_{SS}+0.6$ $V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	Tx RQ0-3, E, $\overline{RESET}$, DGRNT	I_{in}	–	–	2.5	μA
Three-State Leakage Current (V_{in} = 0.4 to 2.4 V)	A0-A15, R/$\overline{W}$ D0-D7	I_{TSI}	−10	–	10	μA
Output High Voltage (I_{Load} = −205 μA (I_{Load} = −145 μA) (I_{Load} = −100 μA)	 D0-D7 A0-A15, R/$\overline{W}$ All Others	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$ $V_{SS}+2.4$	 – – –	 – – –	V
Output Low Voltage (I_{Load} = 1.6 mA)	All Others	V_{OL}	–	–	$V_{SS}+0.4$	V
Source Current (V_{in} = 0 V, Figure 10)	$\overline{CS}$/Tx AKB	I_{CSS}	–	10	16	mA
Internal Power Dissipation (Measured at $T_A = T_L$ to T_H)		P_{INT}	–	500	750	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	E D0-D7, CS, A0-A4, R/$\overline{W}$ All Others	C_{in}	– – –	– – –	20 12.5 10	pF
		C_{out}	–	–	12	pF

MPU MODE TIMING (See Notes 1 and 2)

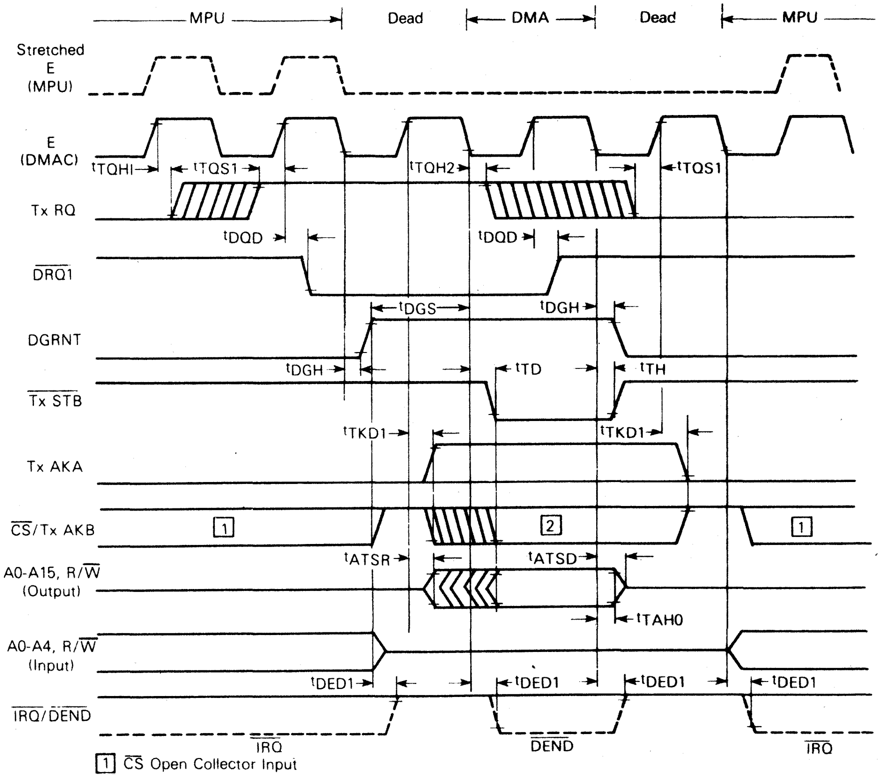
Ident. Number	Characteristic	Symbol	MC6844		MC68A44		MC68B44		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	TBD		ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	–	20	–	20	–	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	TBD		ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 3 — MPU MODE TIMING



NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

FIGURE 4 — MODE 1 TIMING
(TSC STEAL MODE)



1 CS Open Collector Input
2 Tx AKB Output

FIGURE 5 — MODE 2 TIMING
(HALT STEAL MODE)

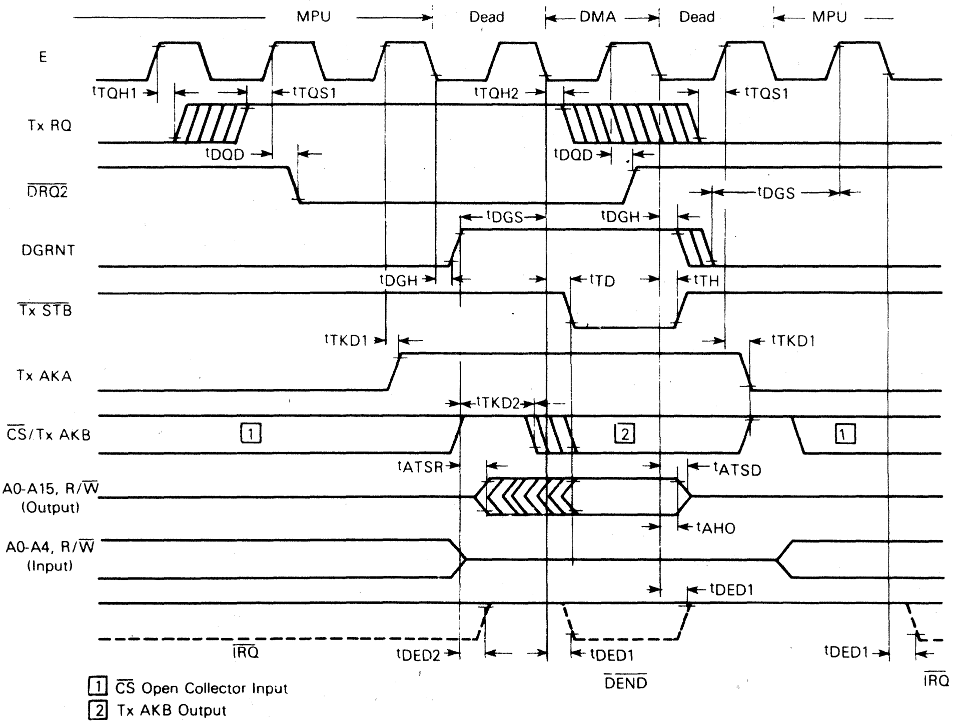
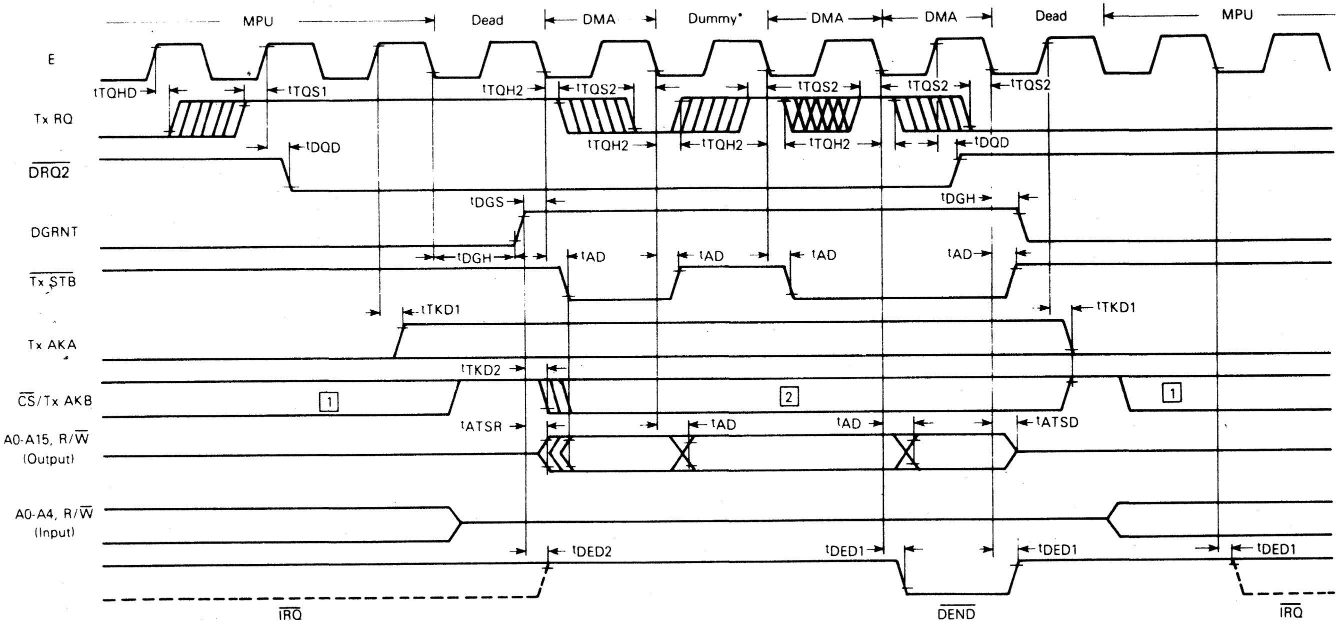


FIGURE 6 — MODE 3 TIMING (HALT BURST MODE)



1 $\overline{CS}$ Open Collector Input

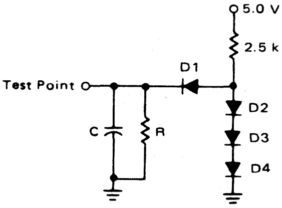
2 Tx AKB Output

*No transfer (dummy cycle) because Tx RQ was negated at start of E cycle.

DMA TIMING (Load Condition Figure 7)

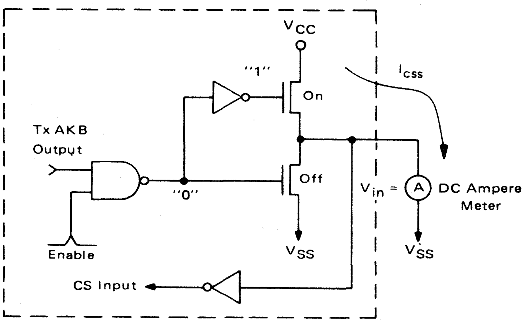
Characteristic		Symbol	MC6844		MC68A44		MC68B44		Unit
			Min	Max	Min	Max	Min	Max	
Tx RQ Setup Time									
E Rising Edge		t_{TQS1}	120	–	120	–	120	–	ns
E Falling Edge		t_{TQS2}	210	–	210	–	170	–	
Tx RQ Hold Time									
E Rising Edge		t_{TQH1}	20	–	10	–	10	–	ns
E Falling Edge		t_{TQH2}	20	–	10	–	10	–	
DGRNT Setup Time		t_{DGS}	155	–	125	–	115	–	ns
DGRNT Hold Time		t_{DGH}	10	–	10	–	10	–	ns
Address Output Delay Time	A0-A15, R/$\overline{W}$	t_{AD}	–	270	–	180	–	150	ns
Address Output Hold Time	A0-A15, R/$\overline{W}$	t_{AHO}	30	–	20	–	20	–	ns
Address Three-State Delay Time	A0-A15, R/$\overline{W}$	t_{ATSD}	–	720	–	460	–	370	ns
Address Three-State Recovery Time		t_{ATSR}	–	430	–	280	–	210	ns
Delay Time	$\overline{DRQ1}$, $\overline{DRQ2}$	t_{DQD}	–	375	–	250	–	200	ns
Tx AK Delay Time									
E Rising Edge		t_{TKD1}	–	400	–	310	–	250	ns
DGRNT Rising Edge		t_{TKD2}	–	190	–	160	–	145	
IRQ/DEND Delay Time									
E Falling Edge		t_{DED1}	–	300	–	250	–	230	ns
DGRNT Rising Edge		t_{DED2}	–	190	–	160	–	145	
Tx STB Output Delay Time		t_{TD}	–	270	–	180	–	150	ns
Tx STB Output Hold Time		t_{TH}	–	20	–	20	–	20	ns

FIGURE 7 — TEST LOADS



Test Pin	C = pF	R = kΩ
D0-D7	130	11.7
A0-A15, R/$\overline{W}$	90	16.5
$\overline{CS}$/Tx AKB	50	24
Others	30	24

FIGURE 8 — CS/Tx AKB SOURCE CURRENT TEST CIRCUIT



INTRODUCTION

The MC6844 DMAC has four DMA channels which can be independently configured by software using fifteen addressable registers. Eight of the addressable registers are 16-bit registers, and seven are 8-bit registers. Associated with each channel are a 16-bit Address Register, a 16-bit Byte Control Register, and an 8-bit Channel Control Register.The DMAC also has three 8-bit registers which affect all of the channels: the Priority Control Register, the Interrupt Control Register, and the Data Chain Register. A block diagram of the DMAC is presented in Figure 2.

SOFTWARE INITIALIZATION

A channel is initialized for DMA by loading the channel address register with the desired starting DMA address and the channel byte control register with the number of bytes to be transferred. In addition, the channel control register must be initialized for the direction of data transfer, for address register increment or decrement after each byte transfer, and for DMA transfer mode.

Each channel can be initialized for one of three transfer modes: Mode 1, Mode 2, or Mode 3. Two read-only status bits in the channel control register indicate when the channel is busy transferring a block of data and when the DMA transfer of a block of data is complete.

The priority control register, the interrupt control register, and the data chain registers must also be initialized.

The priority control register enables/disables each channel and determines whether channel service requests are serviced in a fixed or a rotating priority. The interrupt control register controls assertion of $\overline{\mathrm{IRQ}}$ interrupt by each channel at the end of a data block transfer and sets a flag when $\overline{\mathrm{IRQ}}$ is asserted. The data chain register controls selection of two or four channel operation, selection of data chaining operation, and the channel to be updated in the data chaining mode.

When data chaining is enabled, the contents of the channel 3 address and byte count registers are stored into the corresponding registers of the channel selected for chaining after the channel data block transfer is completed.This feature allows for repetitively reading or writing a block of memory.

HARDWARE INITIALIZATION

At power-on reset (POR) and anytime $\overline{\mathrm{RESET}}$ is asserted, all device registers except the address and byte count registers are cleared. Therefore, the state of the DMAC after reset is as follows:

- all DMA channels are disabled,
- all interrupts are disabled,
- all flags are cleared,
- address register decrement is selected for each channel,
- mode 2 is selected for each channel,
- peripheral controller wirte-to-memory is selected for each channel,
- two-channel operation is selected, and
- data chaining is disabled.

DMAC BUS CONTROL

During DMA operation, the DMAC controls the system address and data buses and generates system R/$\overline{\mathrm{W}}$. The DMAC also generates $\overline{\mathrm{Tx\ STB}}$, which can be used to derive system VMA; Tx AKA and Tx AKB, which can be used to identify which DMA channel is in service; $\overline{\mathrm{DRQ1}}$ and $\overline{\mathrm{DRQ2}}$, which are used for handshaking with the system MPU; $\overline{\mathrm{DEND}}$, which is asserted when the last byte of a data block is being transferred; and $\overline{\mathrm{IRQ}}$, which when enabled will interrupt the system MPU when a data block transfer is completed. Data itself does not pass through the DMAC, but is transferred between memory and peripheral under control of the DMAC.

TRANSFER MODES

Each DMAC channel can be programmed to operate in one of three modes.* Two of the modes, mode 1 and mode 2, are single-byte transfer modes in which the DMAC returns the bus to the MPU after each DMA transfer by negating the appropriate DMA Request ($\overline{\mathrm{DRQ1}}$ or $\overline{\mathrm{DRQ2}}$). These modes are intended to be used in applications requiring the MPU to regain control of the bus after each byte transfer. Timing information for modes 1 and 2 is presented in Figures 4 and 5.

Mode 3 is a block transfer mode in which the DMAC retains control of the bus until the last byte of the DMA data block has been transferred (byte control register 0), if DGRNT remains asserted during the entire block transfer. In mode 3, byte transfers are possible at the DMAC clock frequency by asserting Tx RQ each cycle. This mode offers the highest DMA transfer rate. Mode 3 timing is presented in Figure 6.

A flowchart of DMAC operation in each mode is presented in Figure 9.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power to the DMAC. The power supply should provide +5 V ±5% to V_{CC}. V_{SS} should be tied to ground. Total power dissipation will not exceed P_D milliwatts.

$\overline{\mathrm{RESET}}$

This input is used to place the DMAC into a known state and provide for an orderly startup procedure. Assertion of $\overline{\mathrm{RESET}}$ clears all internal registers except the address and the byte count registers (see Hardware Initialization).

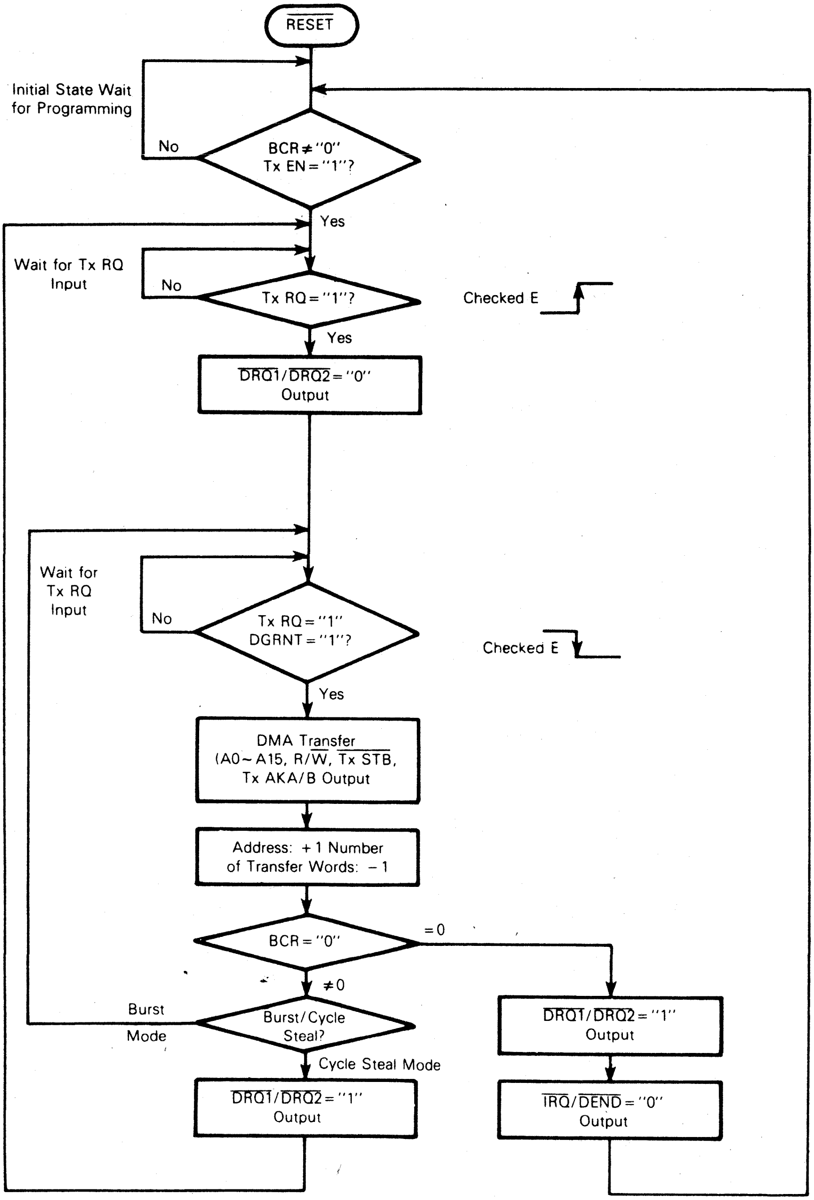
E (ENABLE)

This TTL-compatible input is used to clock the DMAC with the MPU E clock. In systems that perform single-byte transfers by stretching the MPU clock rather than by halting the MPU, the system must be designed to provide a non-stretched E clock to this pin. Clock modules such as the MC6875 are available which provide a separate stretchable E clock to externally-driven MPUs and a non-stretched clock to the DMAC.

---

*Modes 1, 2, and 3 are also called TSC Steal, HALT Steal, and HALT Burst modes.

FIGURE 9 — FLOWCHART OF DMAC OPERATION



READ/WRITE (R/W)

This TTL-compatible bidirectional line is a high-impedance input when the DMAC is off the system bus (MPU mode), and an output when the DMAC is controlling the bus (DMA mode). In the MPU mode, this input is used to control the direction of data transfer through the DMAC data bus interface to allow MPU reads and writes to internal registers. In the DMA mode, Read/Write is an output to the system bus, with its state controlled by bit 0 of the appropriate channel control register.

ADDRESS A0-A15

Address lines A0-A4 are bidirectional. In the MPU mode, these lines are inputs used by the MPU to address DMAC registers. In the DMA mode, these lines and lines A5-A15 are outputs which assert the contents of the address register of the channel being serviced. Address lines A0-A15 are TTL compatible.

DATA D0-D7

These bidirectional TTL-compatible lines are used for data transfer between the MPU and the DMAC. These lines remain in the high-impedance state except when the MPU reads DMAC registers.

INTERRUPT REQUEST/DMA END ($\overline{\text{IRQ}}$/$\overline{\text{DEND}}$)

Interrupt Request/$\overline{\text{DMA End}}$ is a TTL-compatible, time-multiplexed, active low output used to interrupt the MPU and signal a peripheral controller when a DMAC data block transfer has ended. $\overline{\text{DEND}}$ is asserted during the transfer of the last data byte of a block transfer for one E clock cycle (see Figures 4, 5, and 6). $\overline{\text{IRQ}}$ is asserted after the last byte transfer of a block transfer if enabled by setting the proper DEND IRQ enable bit in the interrupt control register (see Table 2). Once asserted, $\overline{\text{IRQ}}$ is negated by reading the channel control register of the channel asserting the interrupt.

TRANSFER REQUEST (Tx RQ0-3)

Associated with each channel is a high-impedance input pin used by a peripheral controller to request DMA service by the channel. The Tx RQ pins are sampled by the DMAC in an order of priority determined by the software-programmable state of the priority control register. The Tx RQ pins for channels programmed for mode 1 or mode 2 operation (single-byte transfer modes) are sampled on the rising edge of E. If Tx RQ for one of these channels is asserted when sampled, the DMAC will perform one DMA byte transfer for the channel before sampling the Tx RQ pin of the channel next in the priority. The Tx RQ pins for channels programmed for mode 3 operation (block transfer mode) are sampled on the rising edge of E for the first DMA byte transfer only. If a Tx RQ for one of these channels is asserted when sampled, the first byte of the channel data block is transferred, then the Tx RQ pin is sampled on falling edges of E for subsequent byte transfers (see Figure 6). Once a channel programmed for mode 3 operation begins DMA, that channel has priority of servicing until the channel completes its entire block transfer.

DMA REQUEST 1-2 ($\overline{\text{DRQ1}}$, $\overline{\text{DRQ2}}$)

These active low TTL-compatible outputs are used by the DMAC to handshake with the MPU in requesting the system bus for DMA operation. $\overline{\text{DRQ1}}$ is asserted to indicate that a channel configured for mode 1 operation requires servicing, and $\overline{\text{DRQ2}}$ is asserted to indicate that a channel configured for mode 2 or mode 3 operation requires servicing. Once asserted, each output remains asserted until the DMAC completes one DMA byte transfer in mode 1 and mode 2 DMA, or an entire byte block transfer in mode 3 DMA.

DMA GRANT (DGRNT)

This high-impedance input is used to enable MC6844 DMA operation and should be asserted only after the MPU has relinquished the system bus to the DMAC. Typically, DGRNT will be asserted by the MPU in response to a DMA request, indicating that the system bus is available for DMA.

TRANSFER STROBE ($\overline{\text{Tx STB}}$)

$\overline{\text{Tx STB}}$ is asserted during each DMA transfer cycle and can be used as a transfer acknowledge for peripheral controllers and as a system VMA. $\overline{\text{Tx STB}}$ is a TTL-compatible output.

TRANSFER ACKNOWLEDGE A (Tx AKA)

Transfer Acknowledge A is asserted during DMA operation and can be used with Tx AKB to identify the DMA channel being serviced, as shown in Table 1.

CHIP SELECT/TRANSFER ACKNOWLEDGE B ($\overline{\text{CS}}$/Tx AKB)

This bidirectional pin serves two functions. During MPU operation it is a chip-select input which when asserted allows MPU access to the DMAC registers. During DMA transfers this pin is for Tx AKB output, used with Tx AKA to identify the DMA channel being serviced (see Table 1).

TABLE 1 — ENCODING ORDER

$\overline{\text{CS}}$/Tx AKB	Tx AKA	Channel #
0	0	0
0	1	1
1	0	2
1	1	3

DMAC REGISTERS

All DMAC registers are read/write regsiters, although some of the register status bits are read-only. Table 2 presents a summary of the DMAC control registers, and Table 3 lists address and byte count register addresses.

ADDRESS REGISTERS

Associated with each DMA channel is an address register which stores the 16-bit address to be asserted on the system

TABLE 2 — DMAC CONTROL REGISTERS

Register	Address (Hex)	Register Content							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Control	1x*	DMA End Flag (DEND)	Busy/$\overline{\text{Ready}}$ Flag	Not Used	Not Used	Address Up/$\overline{\text{Down}}$	MCA	MCB	Read/$\overline{\text{Write}}$ (R/$\overline{\text{W}}$)
Priority Control	14	Rotate Control	Not Used	Not Used	Not Used	Request Enable #3 (RE3)	Request Enable #2 (RE2)	Request Enable #1 (RE1)	Request Enable #0 (RE0)
Interrupt Control	15	DEND IRQ Flag	Not Used	Not Used	Not Used	DEND IRQ Enable #3 (DIE3)	DEND IRQ Enable #2 (DIE2)	DEND IRQ Enable #1 (DIE1)	DEND IRQ Enable #0 (DIE0)
Data Chain	16	Not Used	Not Used	Not Used	Not Used	Two/Four Channel Select (2/4)	Data Chain Channel Select B	Data Chain Channel Select A	Data Chain Enable

*The x represents the binary equivalent of the channel desired.

TABLE 3 — ADDRESS AND BYTE COUNT REGISTERS

Register	Channel	Address (Hex)
Address High	0	0
Address Low	0	1
Byte Count High	0	2
Byte Count Low	0	3
Address High	1	4
Address Low	1	5
Byte Count High	1	6
Byte Count Low	1	7
Address High	2	8
Address Low	2	9
Byte Count High	2	A
Byte Count Low	2	B
Address High	3	C
Address Low	3	D
Byte Count High	3	E
Byte Count Low	3	F

address bus during the next DMA cycle of the channel. After each DMA byte transfer, the address register will increment or decrement according to the state of bit 3 of the appropriate channel control register. The starting address of a DMA data block should be stored in the address register of a channel to be used before beginning DMA operation with the channel.

BYTE COUNT REGISTERS

Each channel has a 16-bit byte count register which stores the number of DMA cycles remaining in a channel DMA block. This register should be loaded with the number of bytes to be transferred by a channel before the channel begins DMA. The byte count register is decremented at the beginning of a DMA cycle.

CHANNEL CONTROL REGISTERS

A channel control register associated with each channel is used to control the channel mode of operation, the state of the R/$\overline{\text{W}}$ line during DMA, and whether the channel address register will increment or decrement after each DMA cycle. The channel control registers contain two read-only status flags which report the status of the channel. The channel control register bits are defined as follows:

Bit 0 R/$\overline{\text{W}}$ Read/Write. The direction of DMA transfer is determined by the state of this bit. When this bit is a "1", R/W will be asserted high by the DMAC during DMA, and memory will be read by the peripheral controller. When this bit is a "0", R/W will be asserted low by the DMAC during DMA and data transfer will be from the peripheral controller to memory.

Bit 1 MCB Mode Control B. This bit is used to select the channel DMA mode. When this bit is a "1", mode 3 operation is selected. When this bit is clear, either mode 1 or mode 2 operation is selected according to the state of channel control register bit 2. Table 4 shows the DMA mode options.

TABLE 4 — DMA MODE SELECT

MCA	MCB	DMA Transfer Mode
0	0	Mode 2
0	1	Mode 3
1	0	Mode 1
1	1	Undefined

Bit 2 MCA	Mode Control A. This bit is used with MCB to select the channel DMA mode. When MCB is set, this bit must be clear and mode 3 operation is selected. Setting both MCA and MCB to a "1" places the DMAC into an undefined mode of operation. With MCB clear, setting MCA to a "1" places the channel into mode 1 and clearing MCA places the channel into mode 2 (see Table 2).
Bit 3	Address Up/Down. Bit 3 controls address register increment/decrement during DMA. If this bit is set to a "1", the address register increments with each DMA cycle; if it is clear, the address register decrements with each DMA cycle.
Bits 4-5	Not used.
Bit 6	Busy/Ready Flag. The Busy/Ready flag is read-only status bit that indicates a DMA block transfer is in progress in the channel. After initializing the channel for a block transfer (address register, byte count register, etc.), this flag sets when Tx RQ is recognized and clears during the last block byte transfer.
Bit 7 DEND	DMA End Flag (DEND). The DEND flag is used to indicate when a DMA transfer is complete. This flag is set during the transfer of the last byte of a DMA block and is cleared by reading the channel control register. This flag will generate an $\overline{\text{IRQ}}$ interrupt if enabled in the interrupt control register.

PRIORITY CONTROL REGISTER

The Priority Control Register is used to individually enable each DMA channel and to select the channel service priority scheme, with bits defined as follows:

Bits 0-3 RE0-3	Request Enable 0-3. Each DMA channel is individually enabled by setting the appropriate RE bit (RE0 for channel 0 etc.) in the priority control register. A clear channel RE bit inhibits recognition of Tx RQ for the channel.
Bits 4-6	Not used.
Bit 7	Rotate Control. One of two channel service priority schemes can be selected by bit 7. When this bit is "0", the fixed priority of servicing is selected in which channel 0 has highest priority, channel 1 has the next highest priority, channel 2 the next highest priority, and channel 3 the last priority. When this bit is set to a "1", the rotating priority of servicing is selected. Rotating priority is initially the same as fixed priority, in that the lower numbered channels initially have the higher priroities. However, once a channel is serviced in the rotating priority mode, that channel is given last priority of servicing. In this scheme the channel last serviced gets the last priority.

INTERRUPT CONTROL REGISTER

The interrupt control register allows the user to selectively enable each channel $\overline{\text{IRQ}}$ interrupt. When enabled, an IRQ is generated when a DMA block transfer is complete. The interrupt control register also has a flag to indicate that the DMAC $\overline{\text{IRQ}}$ is asserted. Interrupt control register bits are defined as follows:

Bits 0-3 DIE0-3	DEND IRQ Enable. These bits enable individual channel $\overline{\text{IRQ}}$ interrupts when set to "1", and mask these interrupts when cleared. The register bit number is the same as the channel number controlled by the bit. An $\overline{\text{IRQ}}$ is asserted only when a DMA block transfer is completed.
Bits 4-6	Not used.
Bit 7	DEND IRQ Flag. This read-only bit is set to a "1" when the DMAC $\overline{\text{IRQ}}$ is asserted, indicating the end of a channel block transfer (DEND assertion) with interrupt enabled. This flag is cleared and $\overline{\text{IRQ}}$ is negated by a read of the channel control register of the channel causing the $\overline{\text{IRQ}}$ interrupt.

DATA CHAIN REGISTER

Repetitive reading or writing of a block of memory can best be performed using the data chain function. This function transfers the contents of the channel 3 address and byte count registers into the respective registers of the channel selected for data chaining. These contents are transferred during the E cycle following the transfer of the last byte of a block by the selected channel. The data chain register is defined as follows:

Bit 0 DCE	Data Chain Enable. Data chaining is enabled when this bit is set to a "1". When this bit is clear, data chaining is disabled.
Bit 1-2 DCA/B	Data Chain Select A, B. The state of these two bits determine which channel will be updated when data chaining is enabled, as listed in Table 5.
Bit 3	Two/Four Channel Select. The DMAC will operate with either two channels or four channels, depending on the state of this bit. When this bit is set to a "1", the four-channel mode is selected, and all four channels are selectable. When this bit is clear, the two-channel mode is selected and only channels 0 and 1 are selectable.
Bits 4-7	Not used.

TABLE 5 — CHANNEL SELECT

DCB Bit 2	DCA Bit 1	Channel #
0	0	0
0	1	1
1	0	2
1	1	Undefined

APPLICATIONS

The MC6844 DMAC can be interfaced to a wide variety of MPUs, including the Motorola MC68000. This section offers examples of MC6844 interface circuits that can be used as starting points in designing the DMAC into a particular system.

IRQ, DEND, Tx AK GENERATION

Derivation of IRQ (Interrupt Request), DEND (DMA End), and Tx AK (Transfer Acknowledge) for one, two, and four-channel DMA is shown in Figure 10. IRQ, if enabled, is asserted by the DMA to interrupt the MPU whenever a DMA block transfer is completed. Tx AK is asserted during each DMA cycle and is used to handshake with a peripheral controller each time a DMA byte transfer occurs. DEND is used to handshake with a peripheral controller each time a DMA block transfer is complete.

Each circuit uses DMA GRANT to demultiplex the IRQ/DEND DMAC output to ensure that the system IRQ is asserted at the proper time, only during MCU operation. Whenever DMA GRANT is high, IRQ is negated.

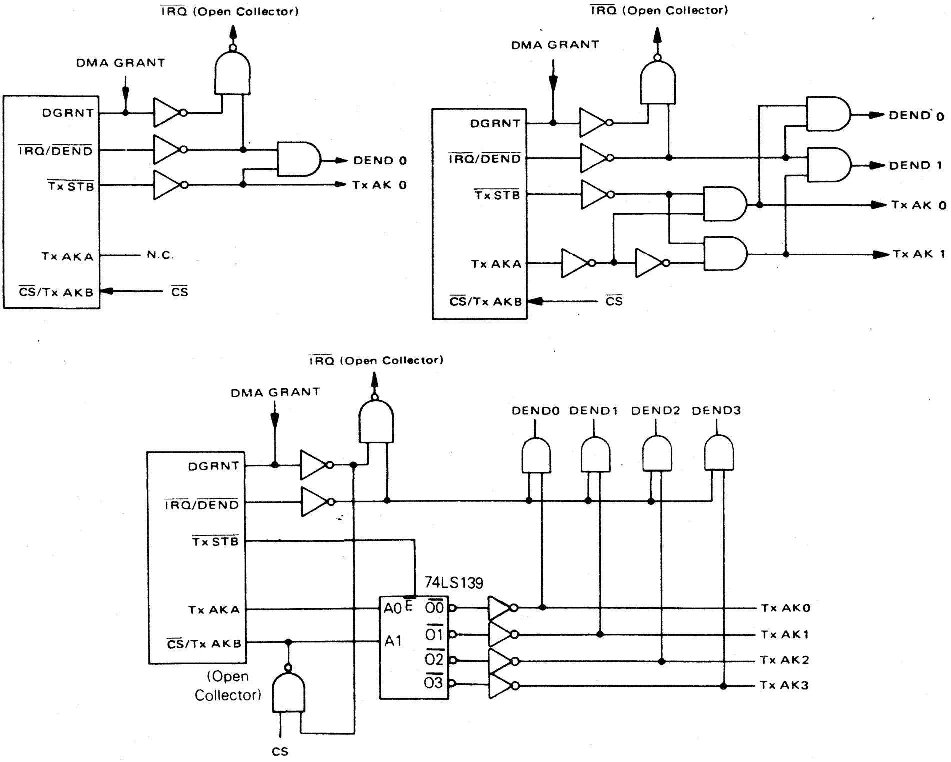
The circuits also generate DEND and Tx AK for the proper channel, gated by Tx STB.

The one-channel DMA mode requires no channel decoding, so for this mode Tx AK is derived from Tx STB directly, and Tx STB is used to demultiplex the IRQ/DEND output for DEND generation.

The two-channel mode circuit is similar to the one-channel circuit, but uses Tx AKA to identify the active channel and generate the appropriate channel signal (see Table 1).

The four-channel circuit is functionally similar to the two-channel circuit but uses a 74LS139 to decode Tx AKA and Tx AKB for channel identification. The DMAC CS/Tx AKB pin is bidirectional during four-channel operation, so an open collector gate must be used to drive CS in order to avoid drive contention.

FIGURE 10 — IRQ, DEND, Tx AK GENERATION



MC68000 BUS ARBITRATION INTERFACE

Figure 11 shows an MC6844/MC68000 interface for DMAC mode 2 or mode 3 operation. The MC68000 Advanced Information Data Sheet should be consulted for complete understanding of the circuit.

The MC6844 must be initialized for transfer mode, byte count, DMA starting address, etc.

Initially DGRNT is low, BGACK output is high, and Tx STB is high. The MC6844 responds to a Tx RQ by asserting DRQH. Assertion of Tx RQ also asserts MC68000 BR. For DMA transfer, two conditions must be met: 1) DMAC DRQH must be asserted and 2) all bus masters must relinquish the system bus. Once DRQH is asserted it remains asserted low until DMA byte-transfer in the halt-steal mode or until the last byte of a DMA memory block is being transferred in the halt-burst mode. A relinquishing of the bus by all bus masters is indicated by negated BGACK, AS, and DTACK after the MC68000 asserts BG in response to a bus request.

When both conditions are met, the NAND flip-flop is set by assertion of LS138 O3, asserting DGRNT and BGACK. The DMAC then performs a byte transfer in the halt-steal mode or a block of byte transfers in the halt-burst mode.

The NAND flip-flop is cleared on the rising edge of Tx STB after asserting during each DMA cycle in the halt-steal mode, and during the last DMA cycle of a DMA block in the halt-burst mode (see MC6844 timing diagrams).

Note that BR to the MC68000 is negated when BGACK is asserted, satisfying an MC68000 requirement.

MC6800 BUS ARBITRATION INTERFACE

A typical system design, using the MC6800/MC6844, is shown in Figure 12. A clock generator/driver is used which will stretch the MPU clock during DMA operation while generating a non-stretched clock for system memory. Priority logic is used to give highest priority to refresh request, since memory refresh and DMA transfers must not occur during the same E cycles.

During mode 2 or 3 DMA operation, the clock generator has no control over DMA Grant. To prevent DMA operation in mode 1 during a memory refresh cycle, system E must be gated with refresh grant. DGRNT must be the ORed output of bus available (BA) and DMA grant from the clock generator in order to support all 3 DMA modes of operation.

During the DMA cycle, a system VMA signal must be generated by the DMAC. This is done by ORing Tx STB and the MPU VMA line.

MC6844/MC6809 BUS ARBITRATION INTERFACE

An MC6844/MC6809 interface is presented in Figure 13. This circuit ensures that MC6809 DMA/BREQ is asserted only during Q high, an MC6809 requirement. The circuit will also generate a system VMA (valid memory address), often referred to as DMA VMA.

The MC6809 does not generate a VMA output since the only invalid address asserted by the MPU is $FFFF with R/W asserted high. Therefore, an MC6809 system does not normally need a VMA circuit. When using the MC6844 for DMA in an MC6809 system, however, a VMA circuit is required since the address lines are floating during dead cycles between the MPU and DMA modes. Devices on the bus must be deselected during this time.

Initially, in the MPU mode, DRQ1/2 is negated (high level), and the Q output of U3 is high. The output of the exclusive OR gate U4 is therefore a low, inhibiting clocking of U3 by forcing the output of U5 to remain a low. When DRQ1/2 is asserted low, the output of U4 changes to a high. If the MC6809 Q output is high at this time, the output of U5 changes to a high, clocking U3. If the MC6809 Q output is low at this time, the output of U5 will be driven high on the next rising edge of Q, clocking U3. When U3 is clocked, the Q output of U3 changes to a low asserting MC6809 DMA/BREQ. The output of U4 at this time is a low, since both of the U4 inputs are low.

FIGURE 11 — MC68000/MC6844 INTERFACE

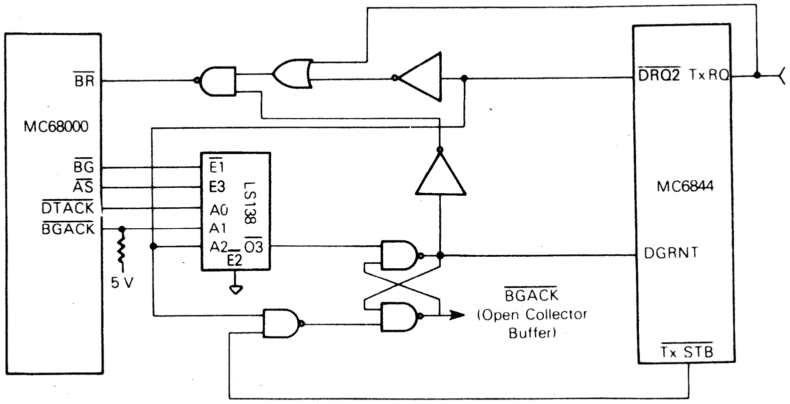


FIGURE 12 — MC6800/MC6844 INTERFACE

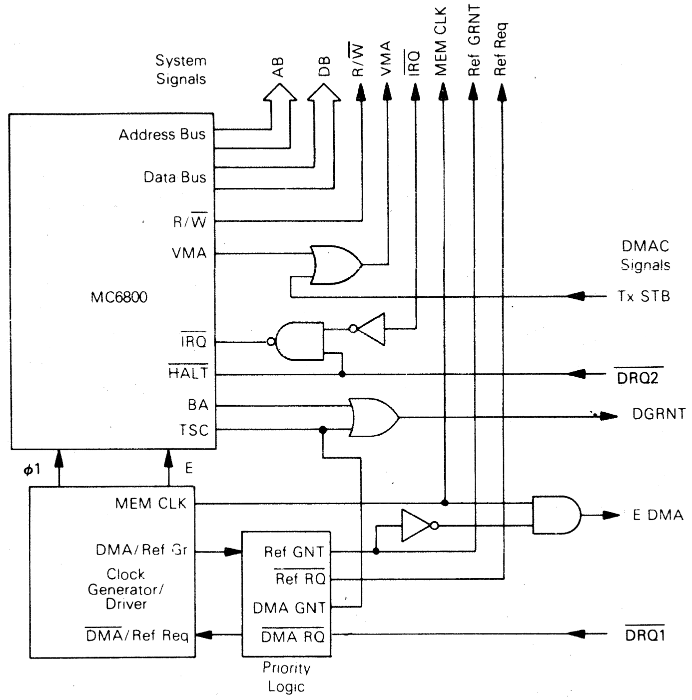
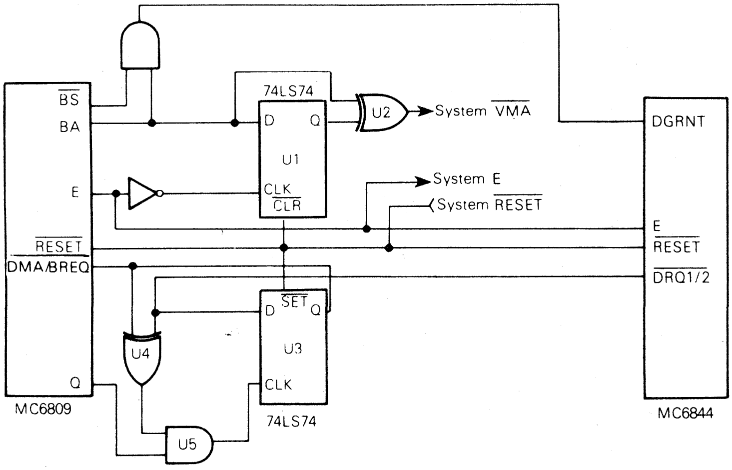


FIGURE 13 — MC6844/MC6809 INTERFACE



After the DMA transfer, DRQ1/2 is negated by the MC6844, forcing the output of U4 to a high. Once again, U3 will be clocked only when the MC6809 Q output is high.

VMA is generated by U1 and U2. Initially, in the MPU mode, U1 is clear, with a low Q output. The BA (bus available) output of the MC6809 is also a low. Therefore, the output of U2 (VMA) is low (VMA asserted). When the MC6809 asserts BA for DMA, the output of U2 becomes high, indicating that the address on the system address bus is invalid during this dead cycle between MPU and DMA modes. On the next falling edge of E, U1 is clocked high forcing the output of U2 low during this DMA cycle. When BA is negated after DMA, the output of U2 is forced high until the next falling edge of E, indicating invalid address during this dead cycle.

MOTOROLA

MC6845 (1.0 MHz) **MC6845☆1** (1.0 MHz)
MC68A45 (1.5 MHz) **MC68A45☆1** (1.5 MHz)
MC68B45 (2.0 MHz) **MC68B45☆1** (2.0 MHz)

MOS

(N-CHANNEL, SILICON-GATE)

CRT CONTROLLER (CRTC)

CRT CONTROLLER (CRTC)

The MC6845 CRT Controller performs the interface between an MPU and a raster-scan CRT display. It is intended for use in MPU-based controllers for CRT terminals in stand-alone or cluster configurations.

The CRTC is optimized for the hardware/software balance required for maximum flexibility. All keyboard functions, reads, writes, cursor movements, and editing are under processor control. The CRTC provides video timing and refresh memory addressing.

- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games; Information Displays
- Alphanumeric, Semi-Graphic, and Full-Graphic Capability
- Fully Programmable Via Processor Data Bus. Timing May Be Generated for Almost Any Alphanumeric Screen Format, e.g., 80 × 24, 72 × 64, 132 × 20
- Single +5 V Supply
- M6800 Compatible Bus Interface
- TTL-Compatible Inputs and Outputs
- Start Address Register Provides Hardware Scroll (by Page, Line, or Character)
- Programmable Cursor Register Allows Control of Cursor Format and Blink Rate
- Light Pen Register
- Refresh (Screen) Memory May be Multiplexed Between the CRTC and the MPU Thus Removing the Requirements for Line Buffers or External DMA Devices
- Programmable Interlace or Non-Interlace Scan Modes
- 14-Bit Refresh Address Allows Up to 16K of Refresh Memory for Use in Character or Semi-Graphic Displays
- 5-Bit Row Address Allows Up to 32 Scan-Line Character Blocks
- By Utilizing Both the Refresh Addresses and the Row Addresses, a 512K Address Space is Available for Use in Graphics Systems
- Refresh Addresses are Provided During Retrace, Allowing the CRTC to Provide Row Addresses to Refresh Dynamic RAMs
- Programmable Skew for Cursor and Display Enable (DE)
- Pin Compatible with the MC6835

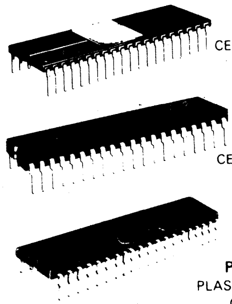
MAXIMUM RATINGS

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 MC6845, MC68A45, MC68B45 MC6845C, MC68A45C, MC68B45C	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	Rating
Thermal Resistance			
Plastic Package	θ_{JA}	100	°C/W
Cerdip Package		60	
Ceramic Package		50	

*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 $V_{out}) \leq V_{CC}$.



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

☆ = Package Suffix

FIGURE 1 — PIN ASSIGNMENTS

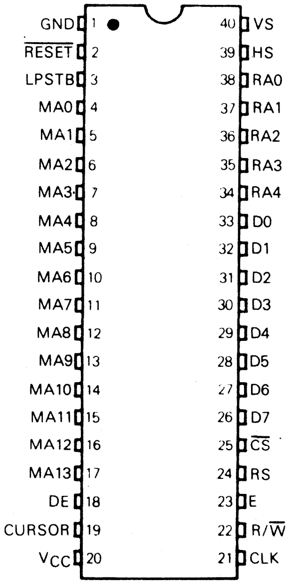
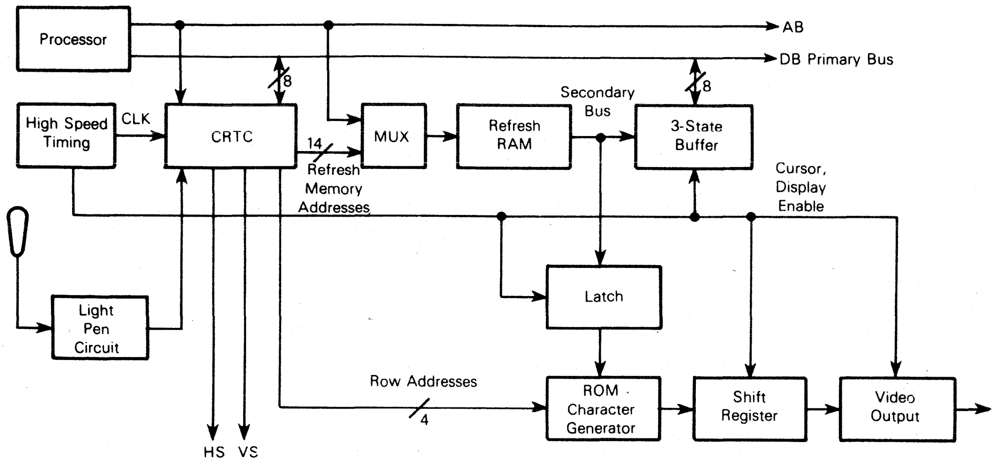


FIGURE 2 — TYPICAL CRT CONTROLLER APPLICATION



RECOMMENDED OPERATING CONDITIONS

Characteristics	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0, T_A = 0 to 70°C unless otheriwse noted, see Figures 3-5)

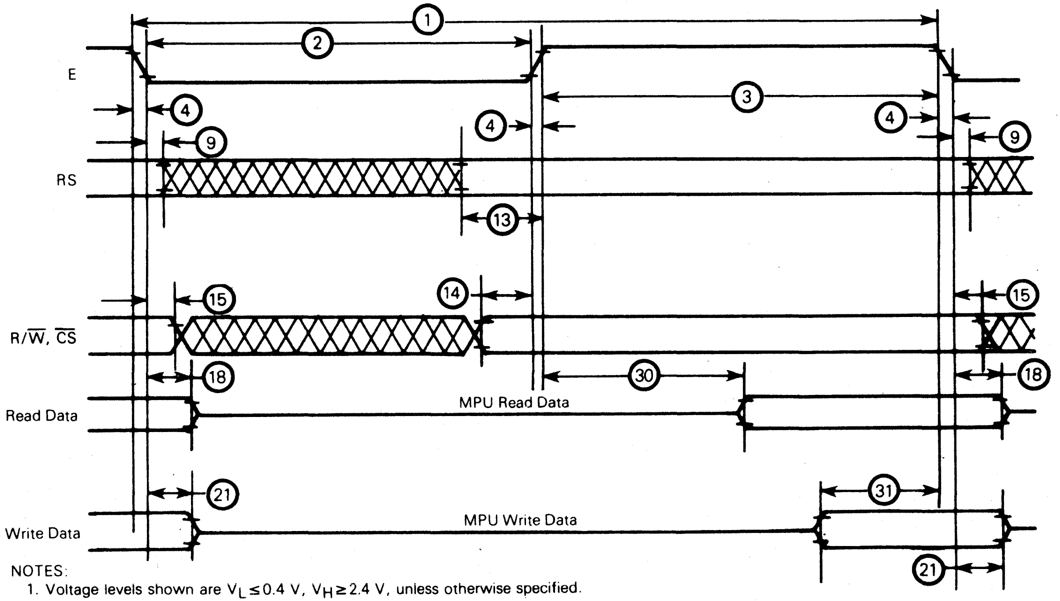
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage		V_{IL}	–0.3	–	0.8	V
Input Leakage Current		I_{in}	–	0.1	2.5	μA
Three-State (V_{CC} = 5.25 V) (V_{in} = 0.4 to 2.4 V)		I_{TSI}	–10	–	10	μA
Output High Voltage (I_{Load} = –205 μA) (I_{Load} = –100 μA)	 D0-D7 Other Outputs	V_{OH}	 2.4 2.4	 3.0 3.0	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA)		V_{OL}	–	0.3	0.4	V
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	600	750	mW
Input Capacitance	D0-D7 All Others	C_{in}	– –	– –	12.5 10	pF
Output Capacitance	All Outputs	C_{out}	–	–	10	pF

BUS TIMING CHARACTERISTICS (See Notes 1 and 2) (Reference Figures 3 and 4)

Ident. Number	Characteristic	Symbol	MC6845 MC6845☆1		MC68A45 MC68A45☆1		MC68B45 MC68B45☆1		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time (RS)	t_{AH}	10	–	10	–	10	–	ns
13	RS Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	R/$\overline{W}$ and $\overline{CS}$ Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	R/$\overline{W}$ and $\overline{CS}$ Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	20	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Peripheral Output Data Delay Time	t_{DDR}	–	290	–	180	0	150	ns
31	Peripheral Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

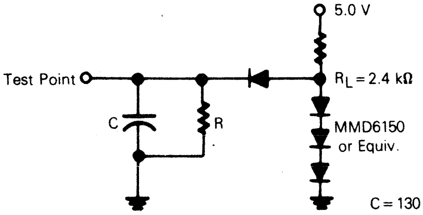
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max (high impedance).

FIGURE 3 — MC6845 BUS TIMING



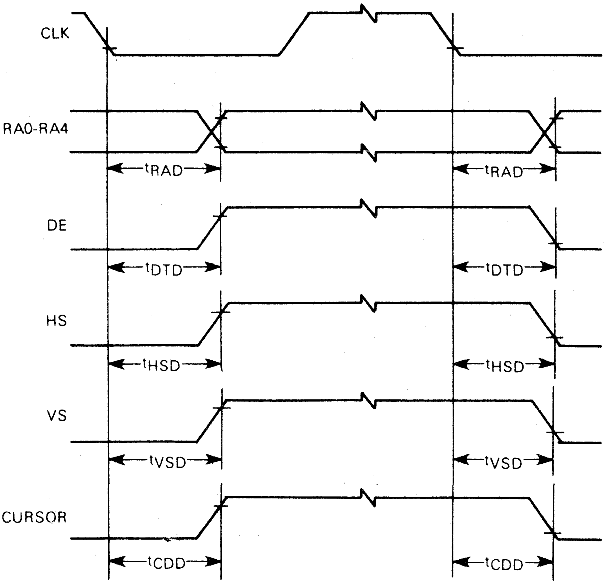
NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 4 — BUS TIMING TEST LOAD



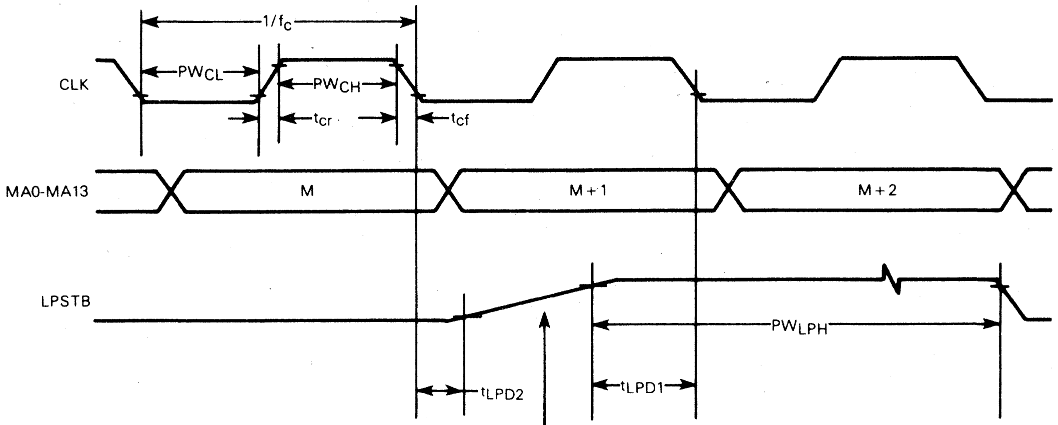
C = 130 pF for D0-D7
= 30 pF for MA0-MA13, RA0-RA4,
DE, HS, VS, and CURSOR
R = 11 kΩ for D0-D7
= 24 kΩ for All Other Outputs

FIGURE 5 — CRTC TIMING CHART



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 6 — CRTC-CLK, MA0-MA13, AND LPSTB TIMING



When the CRTC detects the rising edge of LPSTB in this period, the CRTC sets the Refresh Memory Address 'M+2' into the LIGHT PEN REGISTER.

t_{LPD1}, t_{LPD2}: Period of uncertainty for the Refresh Memory Address.

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.

CRTC TIMING CHARACTERISTICS (Reference Figures 5 and 6)

Characteristic	Symbol	Min	Max	Unit
Minimum Clock Pulse Width, Low	P_{WCL}	160	–	ns
Minimum Clock Pulse Width, High	P_{WCH}	200	–	ns
Clock Frequency	f_c	–	2.5	MHz
Rise and Fall Time for Clock Input	t_{cr}, t_{cf}	–	20	ns
Memory Address Delay Time	t_{MAD}	–	160	ns
Raster Address Delay Time	t_{RAD}	–	160	ns
Display Timing Delay Time	t_{DTD}	–	300	ns
Horizontal Sync Delay Time	t_{HSD}	–	300	ns
Vertical Sync Delay Time	t_{VSD}	–	300	ns
Cursor Display Timing Delay Time	t_{CDD}	–	300	ns
Light Pen Strobe Minimum Pulse Width	PW_{LPH}	100	–	ns
Light Pen Strobe Disable Time	t_{LPD1}	–	120	ns
	t_{LPD2}	–	0	ns

NOTE: The light pen strobe must fall to low level before VS pulse rises.

CRTC INTERFACE SYSTEM DESCRIPTION

The CRT controller generates the signals necessary to interface a digital system to a raster scan CRT display. In this type of display, an electron beam starts in the upper left hand corner, moves quickly across the screen and returns. This action is called a horizontal scan. After each horizontal scan the beam is incrementally moved down in the vertical direction until it has reached the bottom. At this point one frame has been displayed, as the beam has made many horizontal scans and one vertical scan.

Two types of raster scanning are used in CRTs, interlace and non-interlace, shown in Figures 7 and 8. Non-interlace scanning consists of one field per frame. The scan lines in Figure 7 are shown as solid lines and the retrace patterns are indicated by the dotted lines. Increasing the number of frames per second will decrease the flicker. Ordinarily, either a 50 or 60 frame per second refresh rate is used to minimize beating between the CRT and the power line frequency. This prevents the displayed data from weaving.

Interlace scanning is used in broadcast TV and on data monitors where high density or high resolution data must be displayed. Two fields, or vertical scans are made down the screen for each single picture or frame. The first field (even field) starts in the upper left hand corner; the second (odd field) in the upper center. Both fields overlap as shown in Figure 8, thus interlacing the two fields into a single frame.

In order to display the characters on the CRT screen the frames must be continually repeated. The data to be displayed is stored in the refresh (screen) memory by the MPU controlling the data processing system. The data is usually written in ASCII code, so it cannot be directly displayed as characters. A character generator ROM is typically used to convert the ASCII codes into the "dot" pattern for every character.

The most common method of generating characters is to create a matrix of dots "x" dots (columns) wide and "y" dots (rows) high. Each character is created by selectively filling in the dots. As "x" and "y" get larger a more detailed character may be created. Two common dot matrices are 5×7 and 7×9. Many variations of these standards will allow Chinese, Japanese, or Arabic letters instead of English. Since characters require some space between them, a character block larger than the character is typically used, as shown in Figure 9. The figure also shows the corresponding timing and levels for a video signal that would generate the characters.

Referring to Figure 2, the CRT controller generates the refresh addresses (MA0-MA13), row addresses (RA0-RA4), and the video timing (vertical sync — VS, horizontal sync — HS, and display enable — DE). Other functions include an internal cursor register which generates a cursor output when its contents compare to the current refresh address. A light pen strobe input signal allows capture of the refresh address in an internal light pen register.

All timing in the CRTC is derived from the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high-speed logic (TTL) to generate the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high-speed logic (TTL) to generate the CLK signal. The high-speed logic must also generate the timing and control signals necessary for the shift register, latch, and MUX control.

The processor communicates with the CRTC through an 8-bit data bus by reading or writing into the 19 registers.

The refresh memory address is multiplexed between the processor and the CRTC. Data appears on a secondary bus separate from the processor's bus. The secondary data bus concept in no way precludes using the refresh RAM for other purposes. It looks like any other RAM to the processor. A number of approaches are possible for solving contentions for the refresh memory:

1. Processor always gets priority. (Generally, "hash" occurs as MPU and CRTC clocks are not synchronized.)
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize the processor with memory wait cycles (states).
4. Synchronize the processor to the character rate as shown in Figure 10. The M6800 processor family works very well in this configuration as constant cycle lengths are present. This method provides no overhead for the processor as there is never a contention for a memory access. All accesses are transparent.

The present version of the CRTC is being upgraded to improve functionality. This data sheet contains the information describing both the MC6845 (present CRTC) and the MC6845☆1 (upgraded CRTC). Complete compatibility between both versions is maintained by programming all register bits, which are undefined/unused, in the MC6845 with zero's.

FIGURE 7 — RASTER SCAN SYSTEM (NON-INTERLACE)

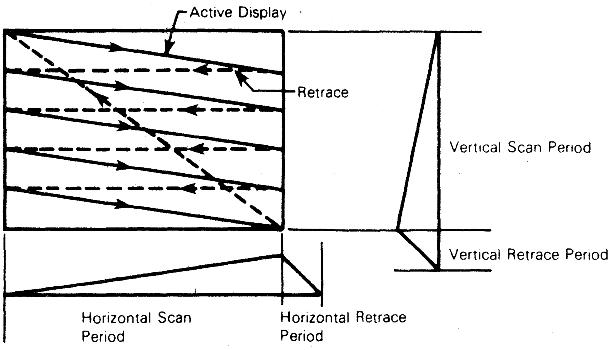


FIGURE 8 — RASTER SCAN SYSTEM (INTERLACE)

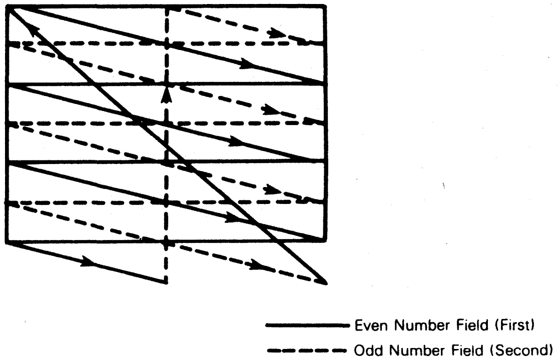


FIGURE 9 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL

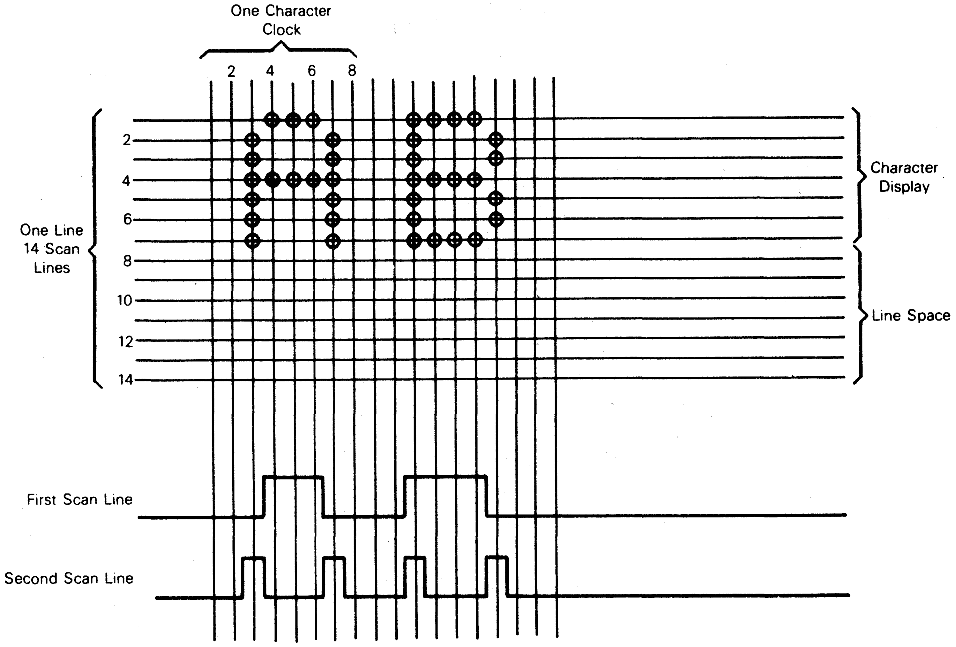
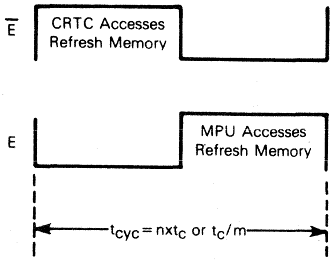


FIGURE 10 — TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING M6800 FAMILY MPU



Where: m, n are integers; t_c is character period

TABLE 1 — CRTC OPERATING MODE

$\overline{\text{RESET}}$	LPSTB	Operating Mode
0	0	Reset
0	1	Test Mode
1	0	Normal Mode
1	1	Normal Mode

The test mode configures the memory addresses as two independent 7-bit counters to minimize test time.

PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{\text{CS}}$, RS, E, and R/$\overline{\text{W}}$ for control signals.

Data Bus (D0-D7) — The bidirectional data lines (D0-D7) allow data transfers between the internal CRTC register file and the processor. Data bus output drivers are high-impedance state until the processor performs a CRTC read operation.

Enable (E) — The Enable signal is a high-impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock. The high-to-low transition is the active edge.

Chip Select ($\overline{\text{CS}}$) — The CS line is a high-impedance TTL/MOS compatible input which selects the CRTC, when low, to read or write to the internal register file. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) — The RS line is a high-impedance TTL/MOS compatible input which selects either the address register (RS = "0") or one of the data register (RS = "1") or the internal register file.

Read/Write (R/$\overline{\text{W}}$) — The R/$\overline{\text{W}}$ line is a high-impedance TTL/MOS compatible input which determines whether the internal register file gets written or read. A write is defined as a low level.

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and display enable (DE) signals.

NOTE

Care should be exercised when interfacing to CRT monitors, as many monitors claiming to be "TTL compatible" have transistor input circuits which require the CRTC or TTL devices buffering signals from the CRTC/video circuits to exceed the maximum-rated drive currents.

Vertical Sync (VS) and Horizontal Sync (HS) — These TTL-compatible outputs are active high signals which drive the monitor directly or are fed to the video processing circuitry to generate a composite video signal. The VS signal determines the vertical position of the displayed text while the HS signal determines the horizontal position of the displayed text.

Display Enable (DE) — This TTL-compatible output is an active high signal which indicates the CRTC is providing addressing in the active display area.

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides memory addresses (MA0-MA13) to scan the refresh RAM. Row addresses (RA0-RA4) are also provided for use with character generator ROMs. In a graphics system, both the memory addresses and the row addresses would be used to scan the refresh RAM. Both the memory addresses and the row addresses continue to run during vertical retrace thus allowing the CRTC to provide the refresh addresses required to refresh dynamic RAMs.

Refresh Memory Addresses (MA0-MA13) — These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs are capable of driving one standard TTL load and 30 pF.

Row Addresses (RA0-RA4) — These five outputs from the internal row address counter are used to address the character generator ROM. These outputs are capable of driving one standard TTL load and 30 pF.

OTHER PINS

Cursor — This TTL-compatible output indicates a valid cursor address to external video processing logic. It is an active high signal.

Clock (CLK) — The CLK is a TTL/MOS-compatible input used to synchronize all CRT functions except for the processor interface. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high-to-low.

Light Pen Strobe (LPSTB) — A low-to-high transition on this high-impedance TTL/MOS-compatible input latches the current Refresh Address in the light pen register. The latching of the refresh address is internally synchronized to the character clock (CLK).

V_{CC}, V_{SS} — These inputs supply +5 Vdc ±5% to the CRTC.

$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input is used to reset the CRTC. A low level on the $\overline{\text{RESET}}$ input forces the CRTC into the following state:

(a) All counters in the CRTC are cleared and the device stops the display operation.
(b) All the outputs are driven low.
(c) The control registers of the CRTC are not affected and remain unchanged.

Functionality of $\overline{\text{RESET}}$ differs from that of other M6800 parts in the following functions:

(a) The $\overline{\text{RESET}}$ input and the LPSTB input are encoded as shown in Table 1.
(b) After $\overline{\text{RESET}}$ has gone low and (LPSTB = "0"), MA0-MA13 and RA0-RA4 will be driven low on the falling edge of CLK. $\overline{\text{RESET}}$ must remain low for at least one cycle of the character clock (CLK).
(c) The CRTC resumes the display operation immediately after the release of $\overline{\text{RESET}}$. DE is not active until after the first VS pulse occurs.

CRTC DESCRIPTION
(Figure 11 CRTC Block Diagram)

The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17. For horizontal timing generation, comparisons result in: 1) horizontal sync pulse (HS) of a frequency, position, and width determined by the registers; 2) horizontal display signal of a frequency, position, and duration determined by the registers.

The horizontal counter produces H clock which drives the scan line counter and vertical control. The contents of the Raster Counter are continuously compared to the maximum scan line address register. A coincidence resets the raster counter and clocks the vertical counter.

Comparisons of vertical counter contents and vertical registers result in: 1) vertical sync pulse (VS) of a frequency, width and position determined by the registers; 2) vertical display of a frequency and position determined by the registers.

The vertical control logic has other functions.

1. Generate row selects, RA0-RA4, from the raster count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the vertical total adjust register.

The linear address generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the start address register, hardware scrolling through 16K characters is possible. The linear address generator repeates the same sequence of addresses for each scan line of a character row.

The cursor logic determines the cursor location, size, and blink rate on the screen. All are programmable.

The light pen strobe going high causes the current contents of the address counter to be latched in the light pen register. The contents of the light pen register are subsequently read by the processor.

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals — R/$\overline{\text{W}}$, $\overline{\text{CS}}$, RS, and E.

REGISTER FILE DESCRIPTIONS

The nineteen registers of the CRTC may be accessed through the data bus. Only two memory locations are required as one location is used as a pointer to address one of the remaining eighteen registers. These eighteen registers control horizontal timing, vertical timing, interlace operation, row address operation, and define the cursor, cursor address, start address, and light pen register. The register addresses and sizes are shown in Table 2.

ADDRESS REGISTER

The address register is a 5-bit write-only register used as an "indirect"or "pointer" register. It contains the address of one of the other eighteen registers. When both RS and $\overline{\text{CS}}$ are low, the address register is selected. When $\overline{\text{CS}}$ is low and RS is high, the register pointed to by the address register is selected.

TIMING REGISTERS R0-R9

Figure 12 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in character clock time units with respect to the reference as shown in Figure 13. The point of reference for the vertical registers is the top character position displayed. Vertical registers are programmed in scan line times with respect to the reference as shown in Figure 14.

Horizontal Total Register (R0) — This 8-bit write-only register determines the horizontal sync (HS) frequency by defining the HS period in character times. It is the total of the displayed characters plus the non-displayed character times (retrace) minus one.

FIGURE 11 — CRTC BLOCK DIAGRAM

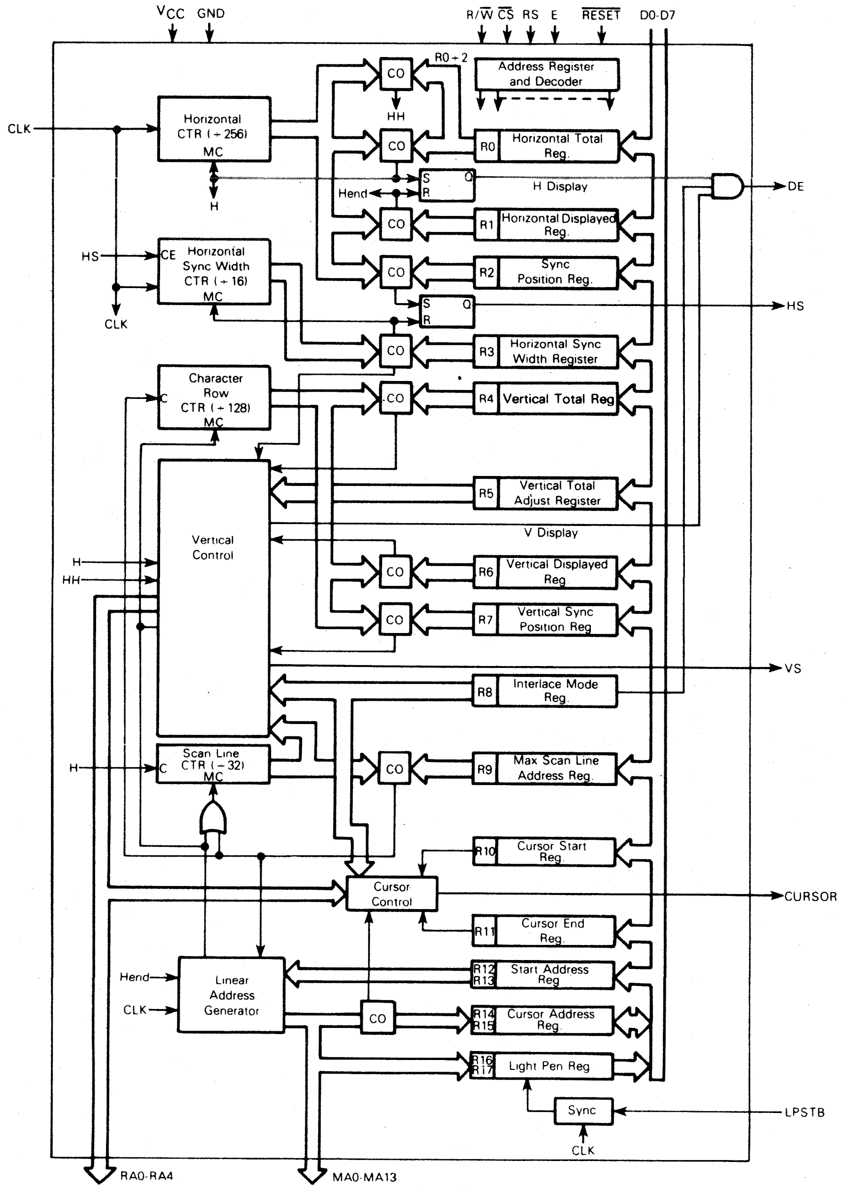


TABLE 2 — CRTC INTERNAL REGISTER ASSIGNMENT

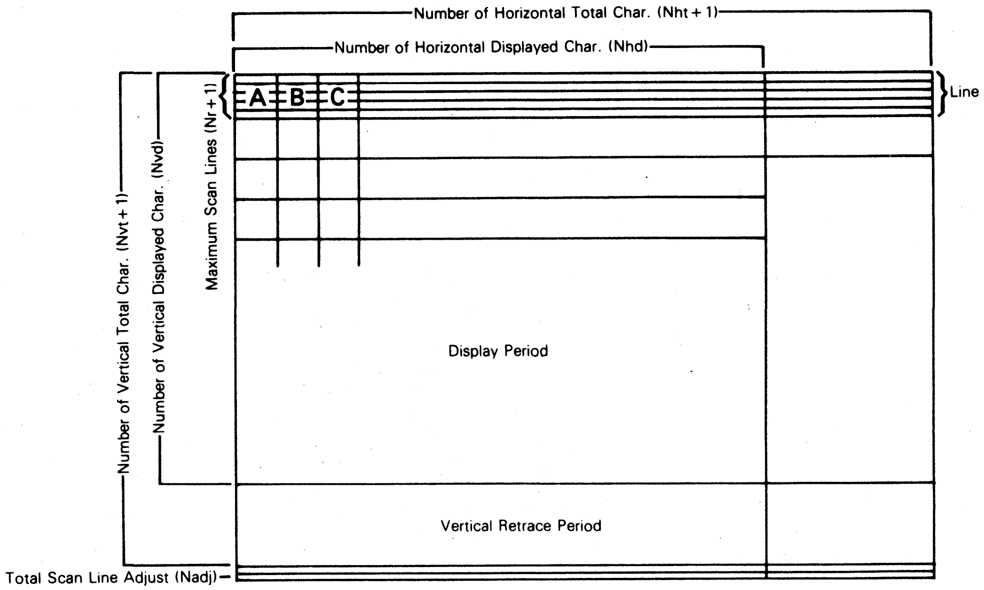
(Features of the MC6845-1 have 1 subscript)

$\overline{CS}$	RS	Address Register					Register #	Register File	Program Unit	Read	Write	Number of Bits							
		4	3	2	1	0						7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X	—	—	—	—	╲	╲	╲	╲	╲	╲	╲	╲
0	0	X	X	X	X	X	AR	Address Register	—	No	Yes	╲	╲	╲					
0	1	0	0	0	0	0	R0	Horizontal Total	Char.	No	Yes								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Char.	No	Yes								
0	1	0	0	0	1	0	R2	H. Sync Position	Char.	No	Yes								
0	1	0	0	0	1	1	R3	Sync Width	—	No	Yes	V_1	V_1	V_1	V_1	H	H	H	H
0	1	0	0	1	0	0	R4	Vertical Total	Char. Row	No	Yes	╲							
0	1	0	0	1	0	1	R5	V. Total Adjust	Scan Line	No	Yes	╲	╲	╲					
0	1	0	0	1	1	0	R6	Vertical Displayed	Char. Row	No	Yes	╲							
0	1	0	0	1	1	1	R7	V. Sync Position	Char. Row	No	Yes	╲							
0	1	0	1	0	0	0	R8	Interlace Mode and Skew	Note 1	No	Yes	C_1	C_1	D_1	D_1	╲	╲	I	I
0	1	0	1	0	0	1	R9	Max Scan Line Address	Scan Line	No	Yes	╲	╲	╲					
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line	No	Yes	╲	B	P			(Note 2)		
0	1	0	1	0	1	1	R11	Cursor End	Scan Line	No	Yes	╲	╲	╲					
0	1	0	1	1	0	0	R12	Start Address (H)	—	Yes	Yes	0	0						
0	1	0	1	1	0	1	R13	Start Address (L)	—	Yes	Yes								
0	1	0	1	1	1	0	R14	Cursor (H)	—	Yes	Yes	0	0						
0	1	0	1	1	1	1	R15	Cursor (L)	—	Yes	Yes								
0	1	1	0	0	0	0	R16	Light Pen (H)	—	Yes	No	0	0						
0	1	1	0	0	0	1	R17	Light Pen (L)	—	Yes	No								

NOTES:

1. The skew control is shown in Table 3 and interlace is shown in Table 4.
2. Bit 5 of the Cursor Start Raster Register is used for blink period control, and Bit 6 is used to select blink or non-blink.

FIGURE 12 — ILLUSTRATION OF THE CRT SCREEN FORMAT



Note 1: Timing values are described in Table 8.

Horizontal Displayed Register (R1) – This 8-bit write-only register determines the number of displayed characters per line. Any 8-bit number may be programmed as long as the contents of R0 are greater than the contents of R1.

Horizontal Sync Position Register (R2) – This 8-bit write-only register controls the HS position. The horizontal sync position defines the horizontal sync delay (Front Porch) and the horizontal scan delay (Back Porch). When the programmed value of this register is increased, the display on the CRT screen is shifted to the left. When the programmed value is decreased the display is shifted to the right. Any 8-bit number may be programmed as long as the sum of the contents of R1, R2, and R3 are less than the contents of R0.

Sync Width Register (R3) – This 8-bit write-only register determines the width of the vertical sync (VS) pulse and the horizontal sync (HS) pulse for the MC6845☆1 CRTC. The vertical sync pulse width is fixed at 16 scan-line times for the MC6845 and the upper four bits of this register are treated as "don't cares."

The MC6845☆1 allows control of the VS pulse width for 1-to-16 scan-line times. Programming the upper four bits for 1-to-15 will select pulse widths from 1-to-15 scan-line times. Programming the upper four bits as zeros will select a VS pulse width of 16 scan-line times, allowing compatibility with the MC6845.

For both the MC6845 and the MC6845☆1, the HS pulse width may be programmed from 1-to-15 character clock periods thus allowing compatibility with the HS pulse width specifications of many different monitors. If zero is written into this register then no HS is provided.

Horizontal Timing Summary (Figure 13) – The difference between R0 and R1 is the horizontal blanking interval. This interval in the horizontal scan period allows the beam to return (retrace) to the left side of the screen. The retrace time is determined by the monitor's horizontal scan components. Retrace time is less than the horizontal blanking interval. A good rule of thumb is to make the horizontal blanking about 20% of the total horizontal scanning period for a CRT. In inexpensive TV receivers, the beam overscans the display screen so that aging of parts does not result in underscanning. Because of this, the retrace time should be about ⅓ the horizontal scanning period. The horizontal sync delay, HS pulse width, and horizontal scan delay are typically programmed with a 1:2:2 ratio.

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) – The frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7-bit write-only vertical total register (R4). The fraction of character line times is programmed in the 5-bit write-only vertical total adjust register (R5) as a number of scan-line times.

Vertical Displayed Register (R6) – This 7-bit write-only register specifies the number of displayed character rows on the CRT screen, and is programmed in character row times. Any number smaller than the contents of R4 may be programmed into R6.

Vertical Sync Position (R7) – This 7-bit write-only register controls the position of vertical sync with respect to the reference. It is programmed in character row times. The value programmed in the register is one less than the number of computed character-line times. When the programmed value of this register is increased, the display position of the CRT screen is shifted up. When the programmed value is decreased the display position is shifted down. Any number equal to or less than the vertical total (R4) may be used.

Interlace Mode and Skew Register (R8) – The MC6845 only allows control of the interlace modes as programmed by the low order two bits of this write-only register. The MC6845-1 controls the interlace modes and allows a programmable delay of zero-to-two character clock times for the DE (display enable) and cursor outputs. Table 3 describes operation of the cursor and DE skew bits. Cursor skew is controlled by bits 6 and 7 of R8 while DE skew is controlled by bits 4 and 5. Table 4 shows the interlace modes available to the user. These modes are selected using the two low order bits of this 6-bit write-only register.

In the normal sync mode (non-interlace) only one field is available as shown in Figures 7 and 15a. Each scan line is refreshed at the VS frequency (e.g., 50 or 60 Hz).

Two interlace modes are available as shown in Figures 8, 15b, and 15c. The frame time is divided between even and odd alternating fields. The horizontal and vertical timing relationship (VS delayed by ½ scan line time) results in the displacement of scan lines in the odd field with respect to the even field.

In the interlace sync mode the same information is painted in both fields as shown in Figure 15b. This is a useful mode for filling in a character to enhance readability.

In the interlace sync and video mode, shown in Figure 15c, alternating lines of the character are displayed in the even field and the odd field. This effectively doubles the given bandwidth of the CRT monitor.

Care must be taken when using either interlace mode to avoid an apparent flicker effect. This flicker effect is due to the doubling of the refresh time for all scan lines since each field is displayed alternately and may be minimized with proper monitor design (e.g., longer persistence phosphors).

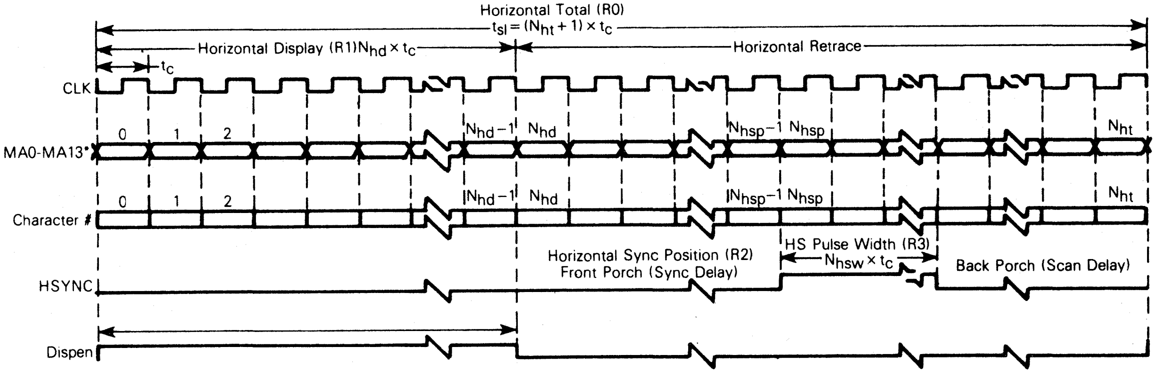
In addition, there are restrictions on the programming of the CRTC registers for interlace operation:

1. For the MC6845:
    a. The horizontal total register value, R0, must be odd (i.e., an even number of character times).
    b. For interlace sync and video mode only, the maximum scan-line address, R9, must be odd (i.e., an even number of scan lines).
    c. For interlace sync and video mode only, the vertical displayed register (R6) must be even. The programmed number Nvd, must be ½ the actual number required. The even numbered scan lines are displayed in the even field and the odd numbered scan lines are displayed in the odd field.
    d. For interlace sync and video mode only, the cursor start register (R10) and cursor end register (R11) must both be even or both odd depending on which field the cursor is to be displayed in.
2. For the MC6845☆1:
    a. The horizontal total register value, R0, must be odd (i.e., an even number of character times).
    b. For the interlace sync and video mode only, the vertical displayed register (R6) must be even. The programmed number, Nvd, must be ½ the actual number required.

TABLE 3 — CURSOR AND DE SKEW CONTROL

Value	Skew
00	No Character Skew
01	One Character Skew
10	Two Character Skew
11	Not Available

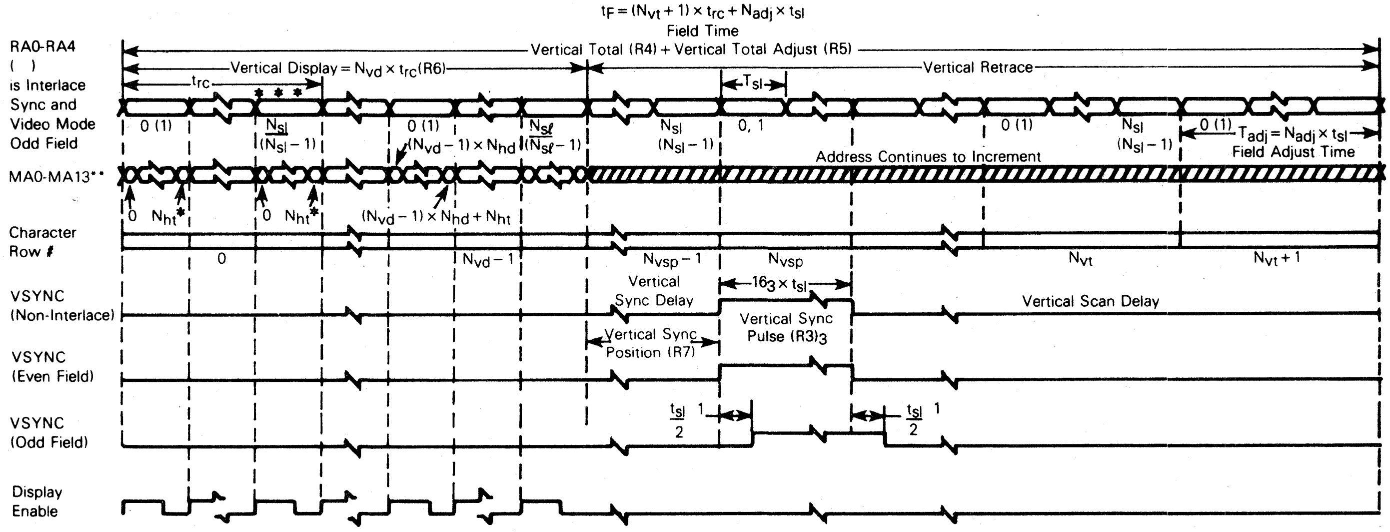
FIGURE 13 — CRTC HORIZONTAL TIMING



*Timing is shown for first displayed scan row only. See Chart in Figure 16 for other rows. The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13=0.

Note 1: Timing values are described in Table 8.

FIGURE 14 — CRTC VERTICAL TIMING



*N_{ht} must be an odd number for both interlace modes.
**Initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.
***N_{sl} must be an odd number for Interlace Sync and Video Mode.

NOTES:
1. Refer to Figure 8 – The Odd Field is offset ½ horizontal scan time.
2. Timing values are described in Table 8.
3. Vertical Sync Pulse width may be programmed from 1 to 16 scan line times for the MC6845☆1.

TABLE 4 — INTERLACE MODE REGISTER

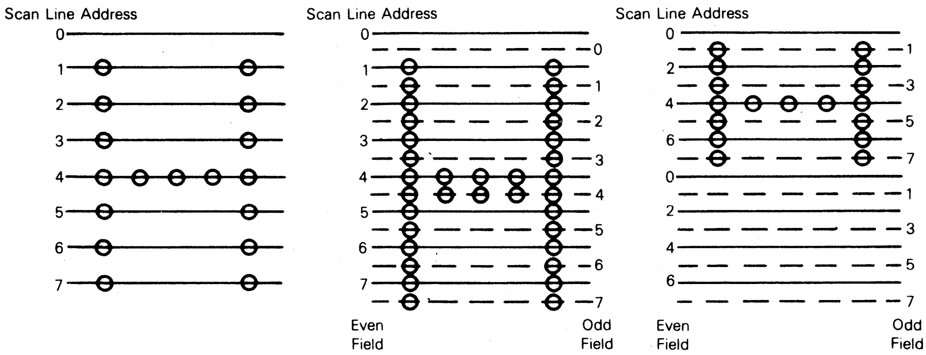
Bit 1	Bit 0	Mode
0 1	0 0	Normal Sync Mode (Non-Interlace)
0	1	Interlace Sync Mode
1	1	Interlace Sync and Video Mode

TABLE 5 — CURSOR START REGISTER

Bit 6	Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate

Example of Cursor Display Mode

FIGURE 15 — INTERLACE CONTROL



(a) Normal Sync (b) Interlace Sync (c) Interlace Sync and Video

Maximum Scan Line Address Register (R9) — This 5-bit write-only register determines the number of scan lines per character row including the spacing; thus, controlling operation of the row address counter. The programmed value is a maximum address and is one less than the number of scan lines.

CURSOR CONTROL

Cursor Start Register (R10) and Cursor End Reigster (R11) — These registers allow a cursor of up to 32 scan lines in height to be placed on any scan line of the character block as shown in Figure 16. R10 is a 7-bit write-only register used to define the start scan line and the cursor blink rate. Bits 5 and 6 of the cursor start address register control the cursor operation as shown in Table 5. Non-display, display, and two blink modes (16 times or 32 times the field period) are available. R11 is a 5-bit write-only register which defines the last scan line of the cursor.

When an external blink feature on characters is required, it may be necessary to perform cursor blink externally so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for a blinking cursor and externally inverting the video signal with an exclusive-OR gate.

Cursor Register (R14-H, R15-L) — This 14-bit read/write register pair is programmed to position the cursor anywhere in the refresh RAM area; thus, allowing hardware paging and scrolling through memory without loss of the original cursor position. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register.

OTHER REGISTERS

Start Address Register (R12-H, R13-L) — This 14-bit write-only register pair controls the first address output by the CRTC after vertical blanking. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. The start address register determines which portion of the refresh RAM is displayed on the CRT screen. Hardware scrolling by character, line, or page may be accomplished by modifying the contents of this register.

Light Pen Register (R16-H, R17-L) — This 14-bit read-only register pair captures the refresh address output by the CRTC on the positive edge of a pulse input to the LPSTB pin. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. Since the light pen pulse is asynchronous with respect to refresh address timing an internal synchronizer is designed into the CRTC. Due to delays (Figure 3) in this circuit, the value of R16 and R17 will need to be corrected in software. Figure 17 shows an interrupt driven approach although a polling routine could be used.

CRTC INITIALIZATION

Registers R0-R15 must be initialized after the system is powered up. The processor will normally load the CRTC register file from a firmware table. The worksheet of Table 6 is extremely useful in computing proper register values for the CRTC. Table 7 shows the worksheet filled out for an 80 × 24 configuration using a 7 × 9 character generator and Figure 18 shows an M6800 program which could be used to program the CRT controller. The programmed values allow use of either an MC6845 or MC6845☆1 CRTC.

The CRTC registers will have an initial value at power up. When using a direct drive monitor (sans horizontal oscillator) these initial values may result in out-of-tolerance operation. CRTC programming should be done immediately after power up especially in this type of system.

FIGURE 16 — CURSOR CONTROL

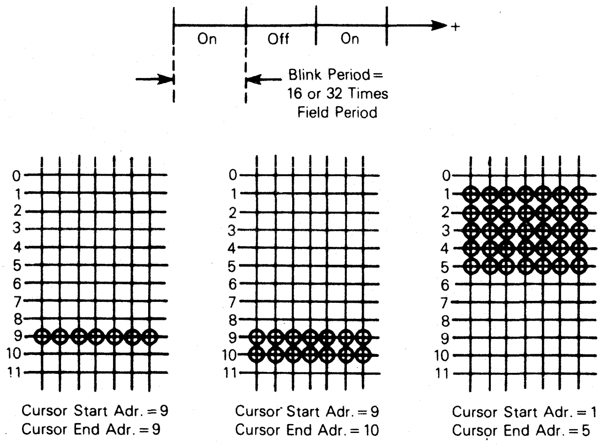


FIGURE 17 — INTERFACING OF LIGHT PEN

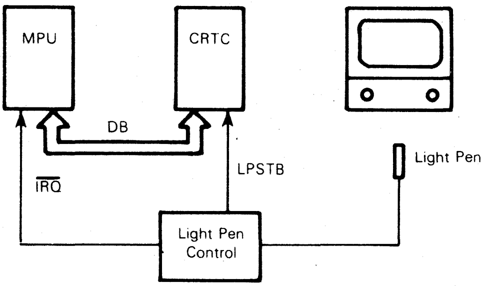


FIGURE 18 — MC6800 PROGRAM FOR CRTC INITIALIZATION

```
PAGE  001  CRTCINIT.SA:0  MC6845 / MC6845-1 CRTC initialization program

00001                            NAM   MC6845
00002                            TTL   / MC6845-1 CRTC initialization program
00003                            OPT   G,S,LLE=85 print FCB's, FDB's & XREF table
00004                     **************************************************
00005                     * Assign CRTC addresses
00006                     *
00007            9000  A CRTCAD EQU   $9000    Address Register
00008            9001  A CRTCRG EQU   CRTCAD+1 Data Register
00009                     **************************************************
00010                     * Initialization program
00011                     *
00012A 0000                      ORG   0        a place to start
00013A 0000 5F                   CLRB           clear counter
00014A 0001 CE 1020  A           LDX   #CRTTAB  table pointer
00015A 0004 F7 9000  A CRTC1     STAB  CRTCAD   load address register
00016A 0007 A6 00    A           LDAA  0,X      get register value from table
00017A 0009 B7 9001  A           STAA  CRTCRG   program register
00018A 000C 08                   INX            increment counters
00019A 000D 5C                   INCB
00020A 000E C1 10    A           CMPB  $10      finished?
00021A 0010 26 F2 0004           BNE   CRTC1    no: take branch
00022A 0012 3F                   SWI            yes: call monitor
00023                     **************************************************
00024                     * CRTC register initialization table
00025                     *
00026A 1020                      ORG   $1020    start of table
00027A 1020     65     A CRTTAB FCB   $65,$50  R0, R1  - H total & H displayed
     A 1021     50     A
00028A 1022     56     A        FCB   $56,$09  R2, R3  - HS pos. & HS width
     A 1023     09     A
00029A 1024     18     A        FCB   $18,$0A  R4, R5  - V total & V total adj.
     A 1025     0A     A
00030A 1026     18     A        FCB   $18,$18  R6, R7  - V displayed $ VS pos.
     A 1027     18     A
00031A 1028     00     A        FCB   $00,$0B  R8, R9  - Interlace & Max scan line
     A 1029     0B     A
00032A 102A     00     A        FCB   $00,$0B  R10,R11 - Cursor start & end
     A 102B     0B     A
00033A 102C     0080   A        FDB   $0080    R12,R13 - Start Address
00034A 102E     0080   A        FDB   $0080    R14,R15 - Cursor Address
00035                           END
TOTAL ERRORS 00000--00000


CRTC1   0004   CRTCAD 9000   CRTCRG 9001   CRTTAB 1020
```

TABLE 6 — CRTC FORMAT WORKSHEET

Display Format Worksheet

1.	Displayed Characters per Row		Char.
2.	Displayed Character Rows per Screen		Rows
3.	Character Matrix a. Columns		Columns
	b. Rows		Rows
4.	Character Block a. Columns		Columns
	b. Rows		Rows
5.	Frame Refresh Rate		Hz
6.	Horizontal Oscillator Frequency		Hz
7.	Active Scan Lines (Line 2 × Line 4b)		Lines
8.	Total Scan Lines (Line 6 ÷ Line 5)		Lines
9.	Total Rows Per Screen (Line 8 ÷ Line 4b)	______ Rows	and ______ Lines
10.	Vertical Sync Delay (Char. Rows)		Rows
11.	Vertical Sync Width (Scan Lines (16)) *	16	Lines
12.	Horizontal Sync Delay (Character Times)		Char. Times
13.	Horizontal Sync Width (Character Times)		Char. Times
14.	Horizontal Scan Delay (Character Times)		Char. Times
15.	Total Character Times (Line 1 + 12 + 13 + 14)		Char. Times
16.	Character Rate (Line 6 × 15)		Hz
17.	Dot Clock Rate (Line 4a × 16)		Hz

CRTC Registers

		Decimal	Hex
R0	Horizontal Total (Line 15 − 1)	______	______
R1	Horizontal Displayed (Line 1)	______	______
R2	Horizontal Sync Position (Line 1 + Line 12)	______	______
R3	Horizontal Sync Width (Line 13) *	______	______
R4	Vertical Total (Line 9 − 1)	______	______
R5	Vertical Adjust (Line 9 Lines)	______	______
R6	Vertical Displayed (Line 2)	______	______
R7	Vertical Sync Position (Line 2 + Line 10)	______	______
R8	Interlace (00 Normal, 01 Interlace, * * 03 Interlace, and Video)		______
R9	Max Scan Line Add (Line 4b − 1)	______	______
R10	Cursor Start	______	______
R11	Cursor End	______	______
R12, R13	Start Address (H and L)	______	______
R14, R15	Cursor (H and L)		______

* NOTE 1: R3 may also be programmed with Vertical Sync. Width for the MC6845★1

** NOTE 2: R8 may also be programmed with Cursor and De Skew for the MC6845★1 (See Table 3)

TABLE 7 — WORKSHEET FOR 80 × 24 FORMAT

Display Format Worksheet

1.	Displayed Characters per Row	80	Char.
2.	Displayed Character Rows per Screen	24	Rows
3.	Character Matrix a. Columns	7	Columns
	b. Rows	9	Rows
4.	Character Block a. Columns	9	Columns
	b. Rows	11	Rows
5.	Frame Refresh Rate	60	Hz
6.	Horizontal Oscillator Frequency	18,600	Hz
7.	Active Scan Lines (Line 2 × Line 4b)	264	Lines
8.	Total Scan Lines (Line 6 ÷ Line 5)	310	Lines
9.	Total Rows Per Screen (Line 8 ÷ Line 4b)	28 Rows and 2	Lines
10.	Vertical Sync Delay (Char Rows)		Rows
11.	Vertical Sync Width (Scan Lines (16)) *	16	Lines
12	Horizontal Sync Delay (Character Times)	6	Char. Times
13.	Horizontal Sync Width (Character Times)	9	Char. Times
14.	Horizontal Scan Delay (Character Times)	7	Char. Times
15.	Total Character Times (Line 1 + 12 + 13 + 14)	102	Char. Times
16.	Character Rate (Line 6 times 15)	1.8972 M	MHz
17.	Dot Clock Rate (Line 4a times 16)	17.075 M	MHz

CRTC Registers

		Decimal	Hex
R0	Horizontal Total (Line 15 minus 1)	101	65
R1	Horizontal Displayed (Line 1)	80	50
R2	Horizontal Sync Position (Line 1 + Line 12)	86	56
R3	Horizontal Sync Width (Line 13) *	9	9
R4	Vertical Total (Line 9 minus 1)	24	18
R5	Vertical Adjust (Line 9 Lines)	10	0A
R6	Vertical Displayed (Line 2)	24	18
R7	Vertical Sync Position (Line 2 + Line 10)	24	18
R8	Interlace (00 Normal, 01 Interlace, * * 03 Interlace, and Video)		0
R9	Max Scan Line Add (Line 4b minus 1)	11	B
R10	Cursor Start	0	0
R11	Cursor End	11	B
R12, R13	Start Address (H and L)	128	00 80
R14, R15	Cursor (H and L)	128	00 80

* NOTE 1: R3 may also be programmed with Vertical Sync. Width for the MC6845*1

** NOTE 2: R8 may also be programmed with Cursor and De Skew for the MC6845*1 (See Table 3)

OPERATION OF THE CRTC

TIMING CHART OF THE CRT INTERFACE SIGNALS

Timing charts of CRT interface signals are illustrated in this section with the aid of programmed example of the CRTC. When values listed in Table 8 are programmed into CRTC control registers, the device provides the outputs as shown in the timing diagrams (Figures 13, 14, 19, and 20). The screen format of this exmaple is shown in Figure 12 which illustrates the relation between refresh memory address (MA0-MA13), raster address (RA0-RA4), and the position on the screen. In this example, the start address is assumed to be "0".

ADDITIONAL CRTC APPLICATIONS

The foremost system function which may be performed by the CRTC controller is the refreshing of dynamic RAM. This is quite simple as the refresh addresses continually run.

Note that the LPSTB input may be used to support additional system functions other than a light pen. A digital-to-analog converter (DAC) and comparator could be configured to use the refresh addresses as a reference to a DAC composed of a resistive adder network connected to a comparator. The output of the comparator would generate the LPSTB input signifying a match between the refresh address analog level and the unknown voltage.

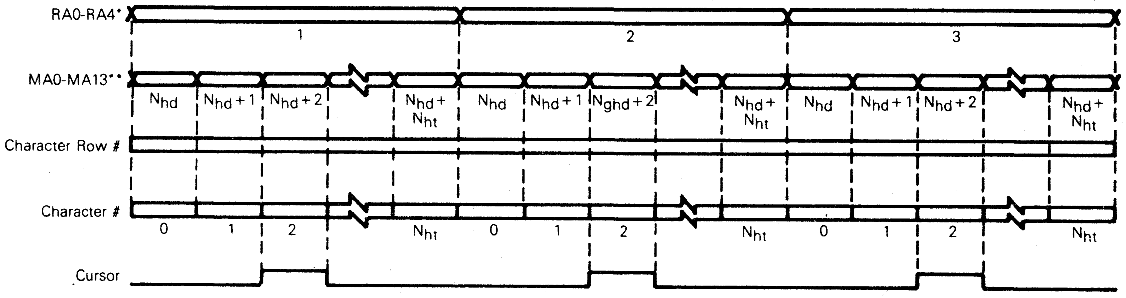
The light pen strobe input could also be used as a character strobe to allow the CRTC refresh addresses to decode a keyboard matrix. Debouncing would need to be done in software.

Both the VS and HS outputs may be used as a real-time clock. Once programmed, the CRTC will provide a stable reference frequency.

TABLE 8 — VALUES PROGRAMMED INTO CRTC REGISTERS

Reg. #	Register Name	Value	Programmed Value
R0	H. Total	$N_{ht}+1$	N_{ht}
R1	H. Displayed	N_{hd}	N_{hd}
R2	H. Sync Position	N_{hsp}	N_{hsp}
R3	H. Sync Width	N_{hsw}	N_{hsw}
R4	V. Total	$N_{vt}+1$	N_{vt}
R5	V. Scan Line Adjust	N_{adj}	N_{adj}
R6	V. Displayed	N_{vd}	N_{vd}
R7	V. Sync Position	N_{vsp}	N_{vsp}
R8	Interlace Mode		
R9	Max. Scan Line Address	N_{sl}	N_{sl}
R10	Cursor Start	1	
R11	Cursor End	3	
R12	Start Address (H)	0	
R13	Start Address (L)	0	
R14	Cursor (H)	0	
R15	Cursor (L)	2	
R16	Light Pen (H)		
R17	Light Pen (L)		

FIGURE 19 — CURSOR TIMING

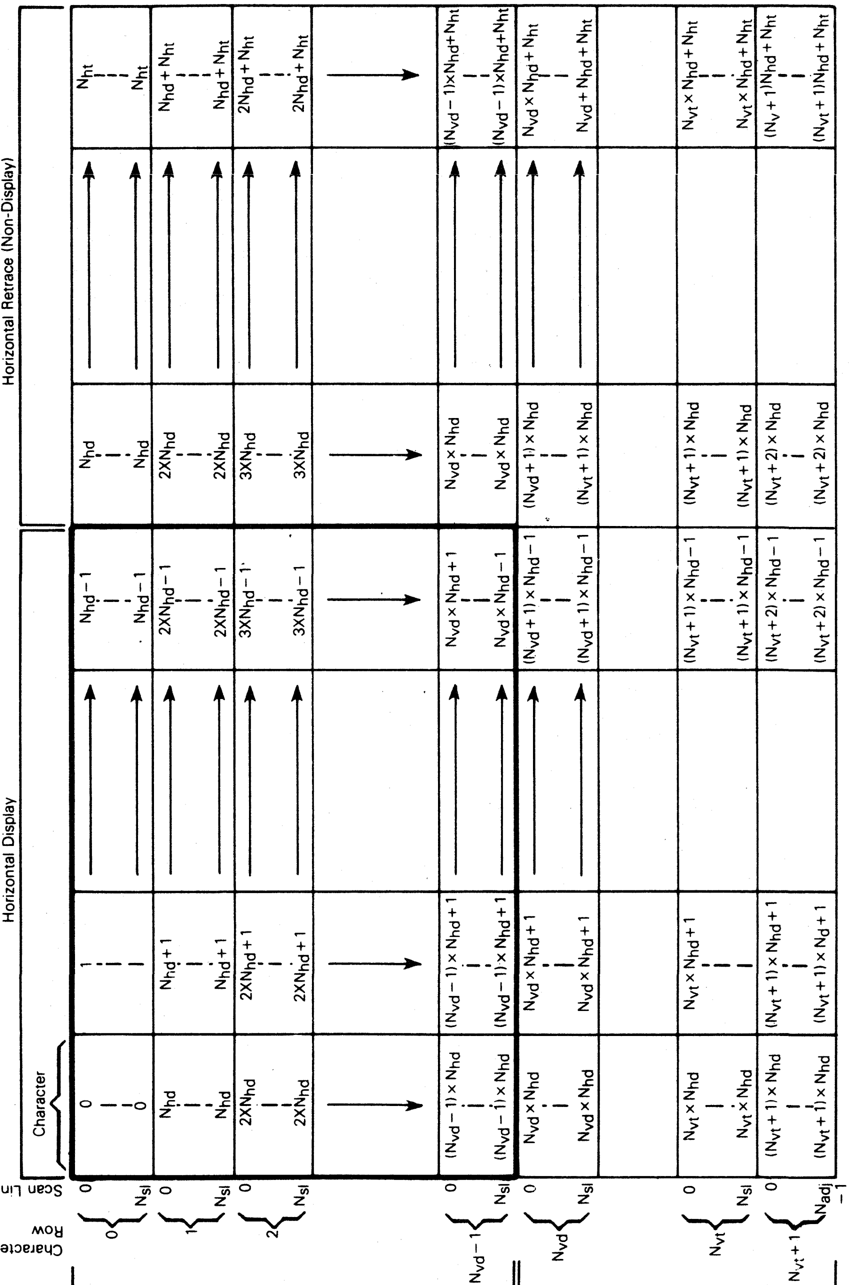


*Timing is shown for non-interlace and interlace sync modes.
Example shown has cursor programmed as:
Cursor Register = $N_{hd}+2$
Cursor Start = 1
Cursor End = 3

**The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

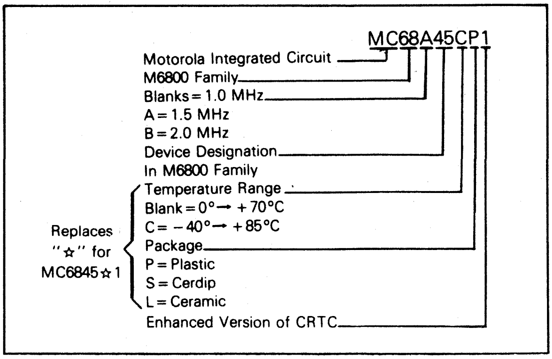
Note 1: Timing values are described in Table 8.

FIGURE 20 — REFRESH MEMORY ADDRESSING (MA0-MA13) STAGE CHART



NOTE 1: The initial MA is determined by the contents of start address register, R12/R13. Timing is shown for R12/R13=0. Only Non-Interlace and Interlace Sync Modes are shown.

ORDERING INFORMATION



Level 1 "S" = 10 Temp Cycles — (−25 to 150°C);
Hi Temp testing at T_A max.
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2.

MOTOROLA

MC6846
(1.0 MHz)
MC68A46
(1.5 MHz)

MOS

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

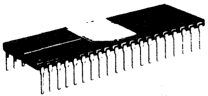
ROM—I/O—TIMER

ROM — I/O — TIMER

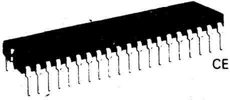
The MC6846 combination chip provides the means, in conjunction with the MC6802, to develop a basic 2-chip microcomputer system. The MC6846 consists of 2048 bytes of mask-programmable ROM, an 8-bit bidirectional data port with control lines, and a 16-bit programmable timer-counter.

This device is capable of interfacing with the MC6802 (basic MC6800, clock, and 128 bytes of RAM) as well as the entire M6800 family if desired. No external logic is required to interface with most peripheral devices.

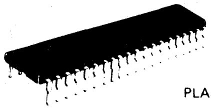
- 2048 8-Bit Bytes of Mask-Programmable ROM
- 8-Bit Bidirectional Data Port for Parallel Interface plus Two Control Lines
- Programmable Interval Timer-Counter Functions
- Programmable I/O Peripheral Data, Control, and Direction Registers
- Compatible with the Complete M6800 Microcomputer Product Family
- TTL-Compatible Data and Peripheral Lines
- Single 5-Volt Power Supply



L SUFFIX
CERAMIC PACKAGE
CASE 715

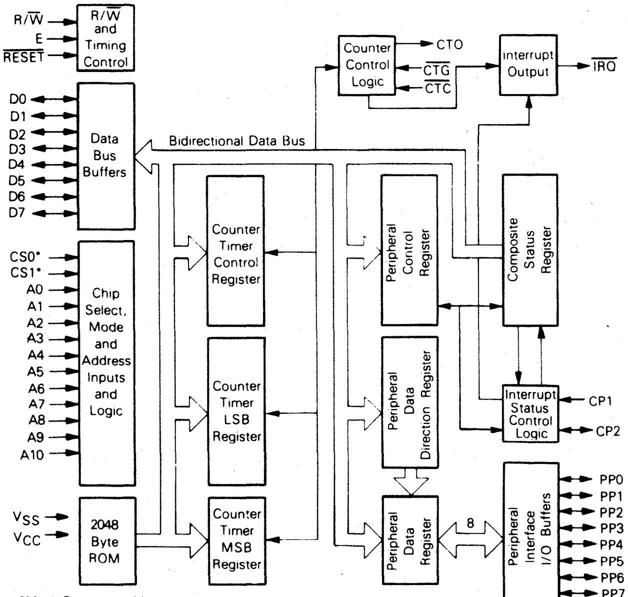


S SUFFIX
CERDIP PACKAGE
CASE 734



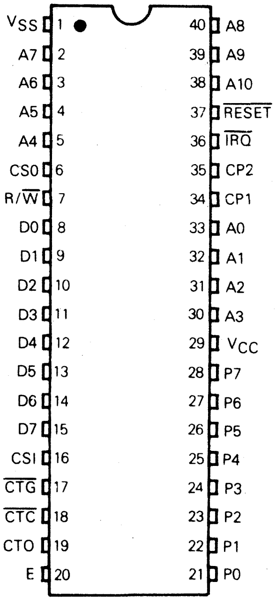
P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 — MC6846 BLOCK DIAGRAM



*Mask Programmable

FIGURE 2 — PIN ASSIGNMENTS



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 MC6846, MC68A46 MC6846C, MC68A46C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. 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	θ_{JA}	50	°C/W
Plastic		100	
Cerdip		60	

POWER CONSIDERATIONS

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

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

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:

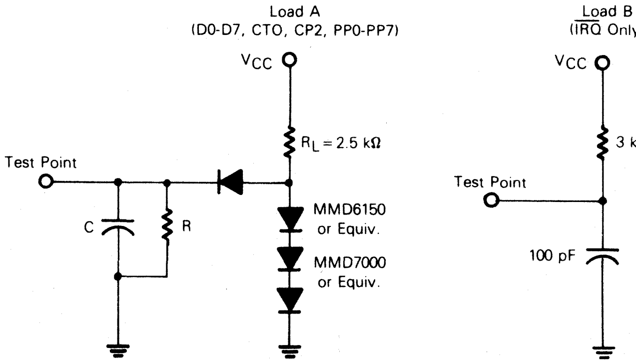
$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

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

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.

FIGURE 3 — BUS TIMING TEST LOADS



C = 130 pF for D0-D7
= 30 pF for CTO, CP2, PP0-PP7
R = 11.7 kΩ for D0-D7
= 24 kΩ for CTO, CP2, PP0-PP7

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	All Inputs	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Clock Overshoot/Undershoot	Input High Level Input Low Level	V_{OS}	$V_{CC}-0.5$ $V_{SS}-0.5$	– –	$V_{CC}+0.5$ $V_{SS}+0.5$	V
Input Leakage Current ($V_{in}=0$ to 5.25 V)	R/$\overline{\text{W}}$, $\overline{\text{RESET}}$, CS0, CS1 CP1, $\overline{\text{CTG}}$, $\overline{\text{CTC}}$, E, A0-A10	I_{in}	–	1.0	2.5	μA
Three-State (Off State) Input Current ($V_{in}=0.4$ to 2.4 V)	D0-D7 PP0-PP7, CP2	I_{TSI}	–	2.0	10	μA
Output High Voltage ($I_{Load}=-205$ μA) ($I_{Load}=-200$ μA)	 D0-D7 Other Outputs	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$	 – –	 – –	V
Output Low Voltage ($I_{Load}=1.6$ mA) ($I_{Load}=3.2$ mA)	 D0-D7 Other Outputs	V_{OL}	 – –	 – –	 $V_{SS}+0.4$ $V_{SS}+0.4$	V
Output High Current (Sourcing) ($V_{OH}=2.4$ V) ($V_O=1.5$ V, the current for driving other than TTL, e.g., Darlington Base)	 D0-D7 Other Outputs CP2, PP0-PP7	I_{OH}	 –205 –200 –1.0	 – – –	 – – –10	 μA μA mA
Output Low Current (Sinking) ($V_{OL}=0.4$ V)	 D0-D7 Other Outputs	I_{OL}	 1.6 3.2	 – –	 – –	mA
Output Leakage Current (Off State) ($V_{OH}=2.4$ V)	$\overline{\text{IRQ}}$	I_{LOH}	–	–	10	μA
Internal Power Dissipation (Measured at $T_A=0$°C)		P_{INT}	–	–	1000	mW
Capacitance ($V_{in}=0$, $T_A=25$°C, f = 1.0 MHz)	D0-D7 PP0-PP7, CP2 A0-A10, R/$\overline{\text{W}}$, $\overline{\text{RESET}}$, CS0, CS1, CP1, $\overline{\text{CTC}}$, $\overline{\text{CTG}}$ $\overline{\text{IRQ}}$	C_{in}	– – – –	– – – –	20 12.5 10 7.5	pF
	PP0-PP7, C2, CT0	C_{out}	– –	– –	5.0 10	pF
Frequency of Operation	MC6846 MC68A46	f	0.1 0.1	– –	1.0 1.5	MHz
Clock Timing Enable Cycle Time		t_{cycE}	1.0	–	–	μs
Reset Low Time		t_{RL}	2	–	–	μs
Interrupt Release		t_{IR}	–	–	1.6	μs

I/O TIMING — Peripheral I/O Lines

Characteristic	Symbol	Min	Max	Unit
Peripheral Data Setup	t_{PDSU}	200	–	ns
Rise and Fall Times CP1, CP2	t_{Pr}, t_{Pf}	–	1.0	μs
Delay Time E to CP2 Fall	t_{CP2}	–	1.0	μs
Delay Time I/O Data CP2 Fall	t_{DC}	20	–	ns
Delay Time E to CP2 Rise	t_{RS1}	–	1.0	μs
Delay Time CP1 to CP2 Rise	t_{RS2}	–	2.0	μs
Peripheral Data Delay	t_{PDW}	–	1.0	μs
Peripheral Data Setup Time for Latch	t_{PDSU}	100	–	ns
Peripheral Data Hold Time for Latch	t_{PDH}	15	–	ns

I/O TIMING — Timer-Counter Lines

Input Rise and Fall Time	$\overline{\text{CTC}}$ and $\overline{\text{CTG}}$	t_{CR}, t_{CF}	–	100	ns
Input Pulse Width High (Asynchronous Mode)		t_{PWH}	$t_{cycE}+250$	–	ns
Input Pulse Width Low (Asynchronous Mode)		t_{PWL}	$t_{cycE}+250$	–	ns
Input Setup Time (Synchronous Mode)		t_{su}	200	–	ns
Input Hold Time (Synchronous Mode)		t_{hd}	50	–	ns
Output Delay		t_{CTO}	–	1.0	μs

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident Number	Characteristic	Symbol	MC6846		MC68A46		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	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	–	60	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	ns

NOTES:

1. Voltage levels shown are $V_L \leq 0.4$ 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.

FIGURE 4 — BUS TIMING

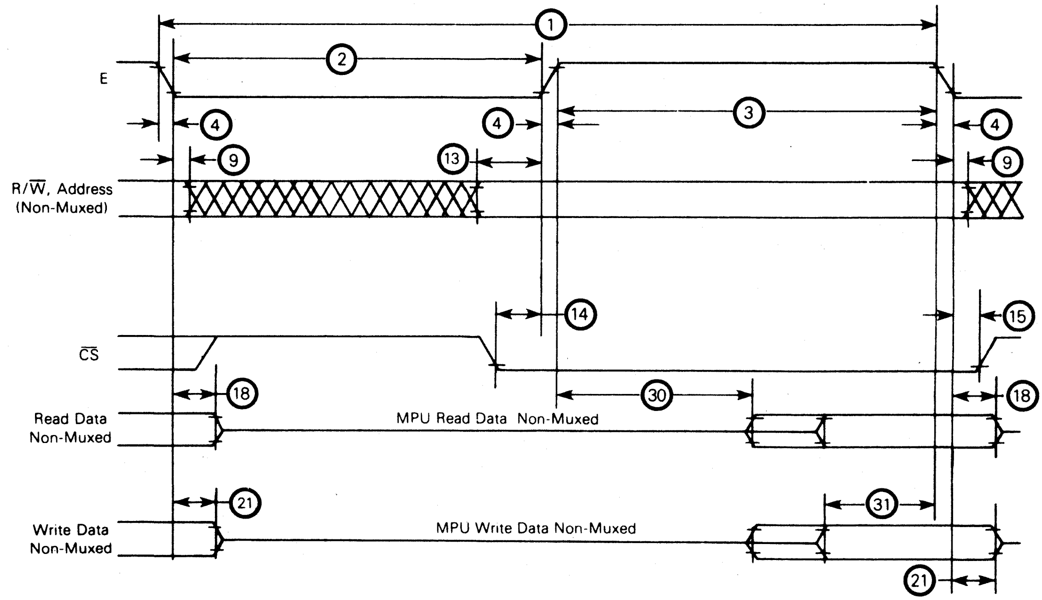


FIGURE 5 — PERIPHERAL PORT LATCH SETUP AND HOLD TIME

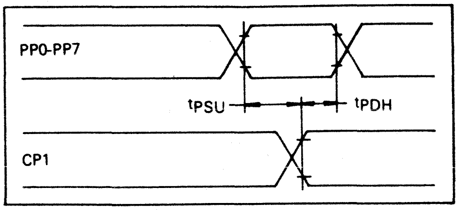
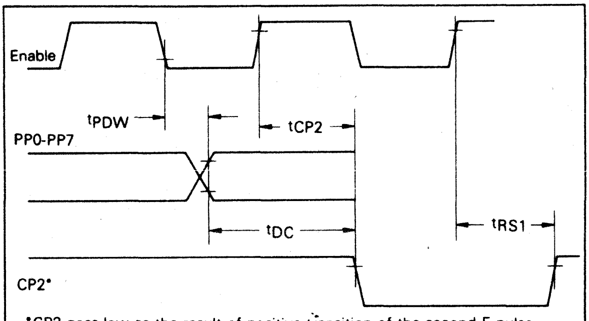


FIGURE 6 — PERIPHERAL DATA AND CP2 DELAY (Control Mode PCR5 = 1, PCR4 = 0, PCR3 = 1)



*CP2 goes low as the result of positive transition of the second E pulse.

FIGURE 7 — $\overline{\text{IRQ}}$ RELEASE TIME

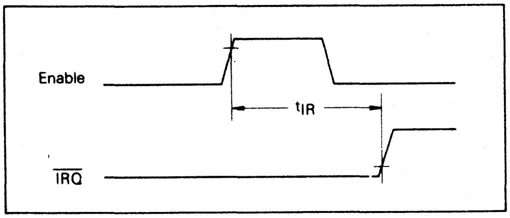


FIGURE 8 — PERIPHERAL PORT SETUP TIME

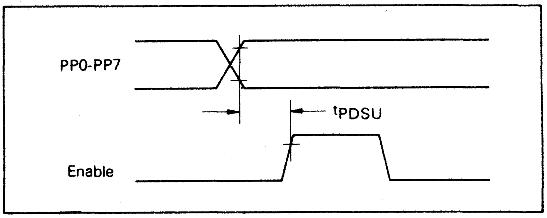


FIGURE 9 — CP2 DELAY TIME (PCR5 = 1, PCR4 = 0, PCR3 = 0)

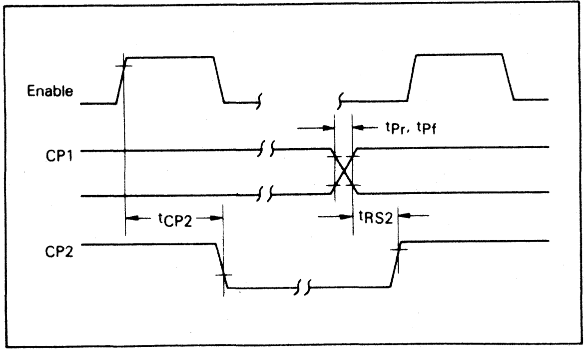


FIGURE 10 — INPUT PULSE WIDTHS

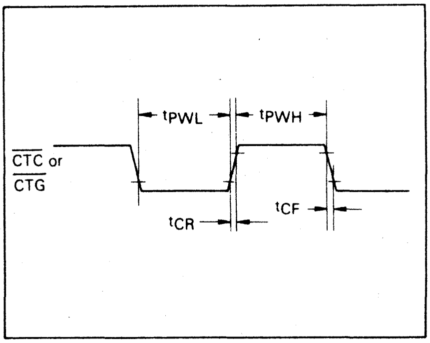
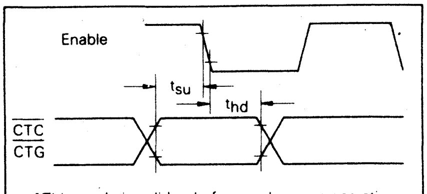
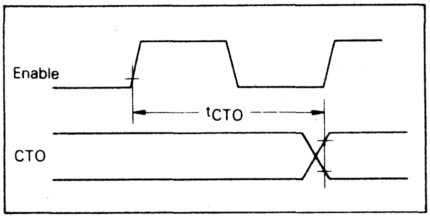


FIGURE 11 — INPUT SETUP AND HOLD TIMES*



*This mode is valid only for synchronous operation.

FIGURE 12 — OUTPUT DELAY



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 13 — TYPICAL MICROCOMPUTER

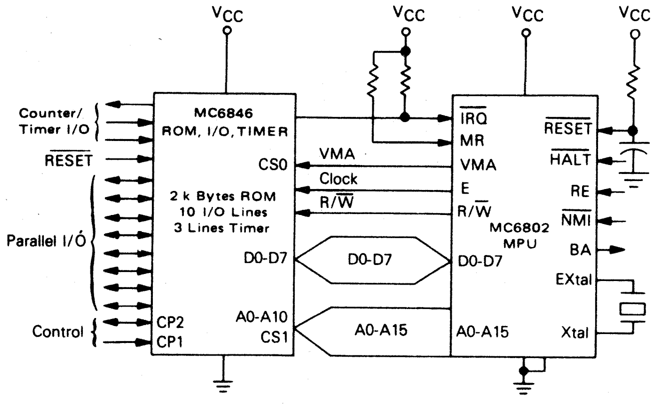


Figure 13 is a block diagram of a typical cost-effective microcomputer. The MPU is the center of the microcomputer system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts in the M6800 Microcomputer Family.

GENERAL DESCRIPTION

The MC6846 combination chip may be partitioned into three functional operating sections: read-only memory, timer-counter functions, and a parallel I/O port.

READ-ONLY MEMORY (ROM)

The mask-programmable ROM section is similar to other ROM products of the M6800 family. The ROM is organized in a 2048 by 8-bit array to provide read-only storage for a minimum microcomputer system. Two mask-programmable chip selects are available for user definition.

Address inputs A0-A10 allow any of the 2048 bytes of ROM to be uniquely addressed. Bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the MC6846.

TIMER-COUNTER FUNCTIONS

Under software control this 16-bit binary counter may be programmed to count events, measure frequencies, time intervals, or similar tasks. Internal registers associated with the I/O functions may be selected with A0, A1, and A2. It may also be used for square wave generation, single pulses of controlled duration, and gated signals. Interrupts may be generated from a number of conditions selectable by software programming.

The timer/counter control register allows control of the interrupt enable, output enable, selection of an internal or external clock source, a divide-by-8 prescaler, and operating mode. Input pin $\overline{\text{CTC}}$ (counter-timer clock) will accept an asynchronous clock pulse to decrement the internal register for the counter-timer. If the divide-by-8 prescaler is used, the maximum clock rate can be four times the master clock frequency. Gate input ($\overline{\text{CTG}}$) accepts an asynchronous TTL-compatible signal which may be used as a trigger or gating function to the counter-timer. A counter-timer output (CTO) is also available and is under software control being dependent on the timer control register, the gate input, and the clock source.

PARALLEL I/O PORT

The parallel bidirectional I/O port has functional operational characteristics similar to the B port on the MC6821 PIA. This includes eight bidirectional data lines and two handshake control signals. The control and operation of these lines are completely software programmable.

The interrupt input (CP1) will set the interrupt flag CSR1 of the composite status register. The peripheral control (CP2) may be programmed to act as an interrupt input (set CSR2) or as a peripheral control output.

SIGNAL DESCRIPTION

BUS INTERFACE

The MC6846 interfaces to the M6800 Bus via an 8-bit bidirectional data bus, two Chip Select lines, a Read/Write line, and eleven address lines. These signals, in conjunction with the M6800 VMA output, permit the MPU to control the MC6846.

BIDIRECTIONAL DATA BUS (D0-D7)

The bidirectional data lines (D0-D7) allow the transfer of data between the MPU and the MC6846. The data bus output drivers are three-state devices which remain in the high-impedance (Off) state except when the MPU performs an MC6846 register or ROM read (R/$\overline{W}$ = 1 and I/O Registers or ROM selected).

CHIP SELECT (CS0, CS1)

The CS0 and CS1 inputs are used to select the ROM or I/O timer of the MC6846. They are mask programmed to be active high or active low as chosen by the user.

ADDRESS INPUTS (A0-A10)

The Address Inputs allow any of the 2048 bytes of ROM to be uniquely selected when the circuit is operating in the ROM mode. In the I/O-Timer mode, address inputs A0, A1, and A2 select the proper I/O Register, while A3 through A10 (together with CS0 and CS1) can be used as additional qualifiers in the I/O Select circuitry. (See the section on I/O-Timer Select for additional details.)

$\overline{\text{RESET}}$

The active low state of the $\overline{\text{RESET}}$ input is used to initialize all register bits in the I/O section of the device to their proper values. (See the section on Initialization for reset conditions for timer and peripheral registers.)

ENABLE (E)

This signal synchronizes data transfer between the MPU and the MC6846. It also performs an equivalent synchronization function on the external clock, reset, and gate inputs of the MC6846 Timer section.

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

This signal is generated by the MPU and is used to control the direction of data transfer on the bidirectional data pins. A low level on the R/$\overline{W}$ input enables the MC6846 input buffers and data is transferred to the circuit during the E pulse when the part has been selected. A high level on the R/$\overline{W}$ input enables the output buffers and data is transferred to the MPU during E when the part is selected.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$)

The active low $\overline{\text{IRQ}}$ output acts to interrupt the MPU through logic included on the MC6846. This output utilizes an open-drain configuration and permits other interrupt request outputs from other circuits to be connected in a wire-OR configuration.

PERIPHERAL DATA (P0-P7)

The peripheral data lines can be individually programmed as either inputs or outputs via the Data Direction Register. When programmed as outputs, these lines will drive two standard TTL loads (3.2 mA). They are also capable of sourcing up to 1.0 mA at 1.5 V (Logic "1" output.)

When programmed as inputs, the output drivers associated with these lines enter a three-state (high impedance) mode. Since there is no internal pullup for these lines, they represent a maximum 10 μA load to the circuitry driving them — regardless of logic state.

A logic zero at the $\overline{\text{RESET}}$ input forces the peripheral data lines to the input configuration by clearing the Data Direction Register. This allows the system designer to preclude the possibility of having a peripheral data output connected to an external driver output during power-up sequence.

INTERRUPT INPUT (CP1)

Peripheral input line CP1 is an input-only that sets the Interrupt Flags of the Composite Status register. The active transition for this signal is programmed by the peripheral control register for the parallel port. CP1 may also act as a strobe for the peripheral data register when it is used as an input latch. Details for programming CP1 are in the section on the parallel peripheral port.

PERIPHERAL CONTROL (CP2)

Peripheral Control line CP2 may be programmed to act as an Interrupt input or Peripheral Control output. As an input, this line has high impedance and is compatible with standard TTL voltage levels. As an output, it is also TTL compatible and may be used as a source of 1 mA at 1.5 V to directly drive the base of a Darlington transistor switch. This line is programmed by the Peripheral Control Register.

COUNTER TIMER OUTPUT (CTO)

The Counter Timer Output is software programmable by selected bits in the timer/counter control register. The mode of operation is dependent on the Timer control register, the gate input, and the clock source. The output is TTL compatible.

EXTERNAL CLOCK INPUT ($\overline{\text{CTC}}$)

Input pin $\overline{\text{CTC}}$ will accept asynchronous TTL voltage signals to be used as a clock to decrement the Timer. The high and low levels of the external clock must be stable for at least one system clock period plus the sum of the setup and hold times for the inputs. The asynchronous clock rate can vary from dc to the limit imposed by E setup, and hold times.

The external clock input is clocked in by Enable (E) pulses. Three E periods are used to synchronize and process the external clock. The fourth E pulse decrements the internal counter. This does not affect the input frequency; it merely creates a delay between a clock input transition and internal recognition of that transition by the MC6846. All references to CTC inputs in this document relate to internal recognition of the input transition. Note that a clock transition which does not meet setup and hold time specifications may require an additional E pulse for recognition.

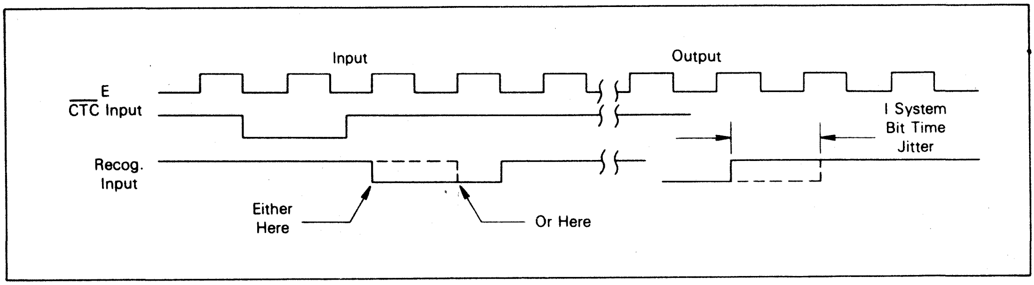
When observing recurring events, a lack of synchronization will result in either "System jitter" or "Input jitter" being observed on the output of the MC6846 when using an asynchronous clock and gate input signal. "System jitter" is the result of the input signals being out of synchronization with E permitting signals with marginal setup and hold time to be recognized by either the bit time nearest the input transition or subsequent bit time. "Input jitter" can be as great as the time between the negative going transitions of the input signal plus the system jitter if the first transition is recognized during one system cycle, and not recognized the next cycle or vice-versa. Refer to Figure 14.

GATE INPUTS ($\overline{\text{CTG}}$)

The input pin $\overline{\text{CTG}}$ accepts an asynchronous TTL-compatible signal which is used as a trigger or a clock gating function to the Timer. The gating input is clocked into the MC6846 by the E signal in the same manner as the previously discussed clock inputs. That is, $\overline{\text{CTG}}$ transition is recognized on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the $\overline{\text{CTG}}$ input must be stable for at least one system clock period plus the sum of setup and hold times. All references to $\overline{\text{CTG}}$ transition in this document relate to internal recognition of the input transition.

The $\overline{\text{CTG}}$ input of the timer directly affects the internal 16-bit counter. The operation of $\overline{\text{CTG}}$ is therefore independent of the divide-by-8 prescaler selection.

FIGURE 14 — RECOGNITION OF $\overline{\text{CTC}}$



FUNCTIONAL SELECT CIRCUITRY

I/O-TIMER SELECT CIRCUITRY

CS0 and CS1 are user programmable. Any of the four binary combinations of CS0 and CS1 can be used to select the ROM. Likewise, any other combination can be used to select the I/O-Timer. In addition, several address lines are used as qualifiers for the I/O-Timer. Specifically, A3 = A4 = A5 = logical "0". A6 can be programmed to a "1", "0", or don't care. A7 = A8 = A9 = A10 = don't care or only one line may be programmed to a logical "1". Figure 15 outlines in diagrammatic form the available chip select options.

INTERNAL ADDRESS

Seven I/O Register locations within the MC6846 are accessible to the MPU data bus. Selection of these registers is controlled by A0, A1, and A2 (as shown in Table 1) provided the I/O timer is selected. The combination status register is Read-only; all other Registers are Read and Write.

REGISTER SELECTED	A2	A1	A0
Combination Status Register	0	0	0
Peripheral Control Register	0	0	1
Data Direction Register	0	1	0
Peripheral Data Register	0	1	1
Combination Status Register	1	0	0
Timer Control Register	1	0	1
Timer MSB Register	1	1	0
Timer LSB Register	1	1	1
ROM Address	X	X	X

TABLE 1 — INTERNAL REGISTER ADDRESSES

INITIALIZATION

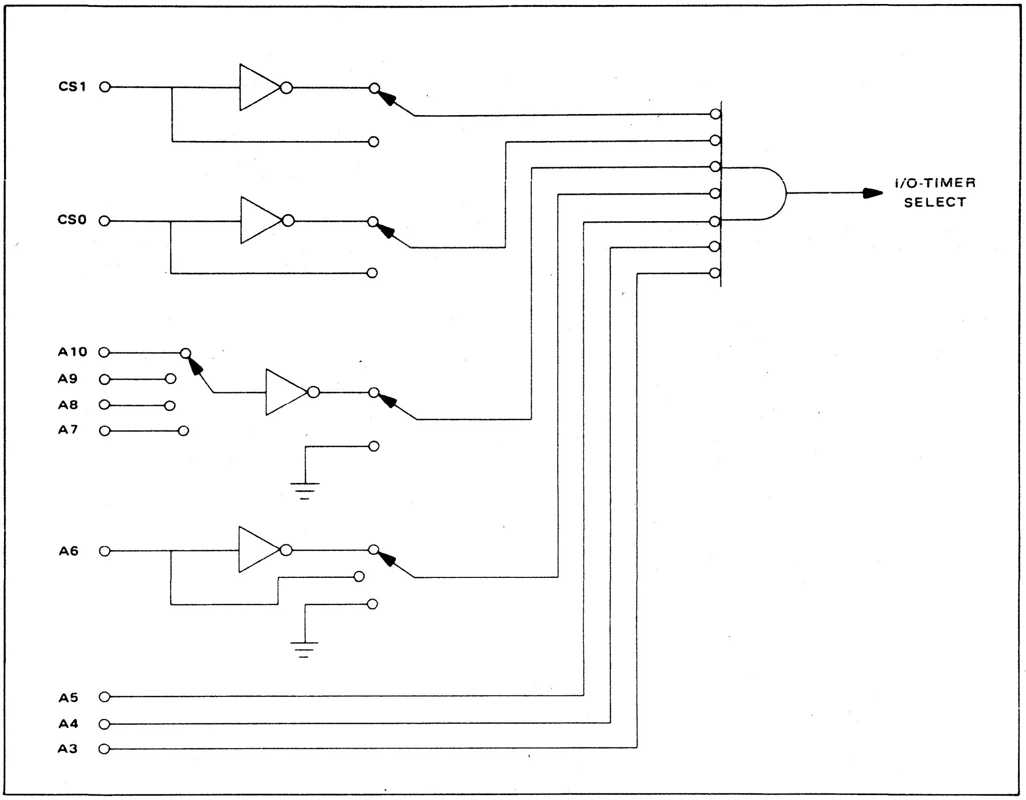
When the $\overline{\text{RESET}}$ input has accepted a low signal, all registers are initialized to the reset state. The data direction and peripheral data registers are cleared. The Peripheral Control Register is cleared except for bit 7 (the $\overline{\text{RESET}}$ bit). This forces the parallel port to the input mode with Interrupts disabled. To remove the reset condition from the parallel port, a "0" must be written into the Peripheral Control Register bit 7 (PCR7).

The counter latches are preset to their maximal count, the Timer control register bits are reset to zero except for Bit 0 (TCR0 is set), the counter output is cleared, and the counter clock disabled. This state forces the timer counter to remain in an inactive state. The combination status register is cleared of all interrupt flags. During timer initialization, the reset bit (CCR0) must be cleared.

ROM

The Mask Programmable ROM section is similar in operation to other ROM products of the M6800 Microcessor family. The ROM is organized as 2048 words of 8-bits to provide read-only storage for a minimum microcomputer system. The ROM is active when selected by the unique combination of the chip select inputs.

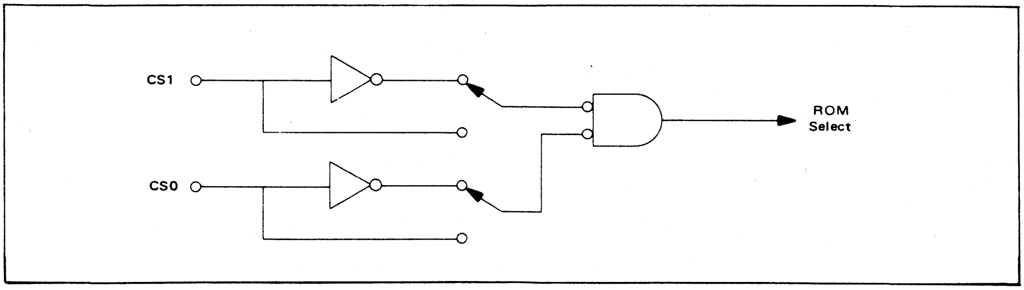
FIGURE 15 — I/O-TIMER SELECT CIRCUITRY



ROM SELECT

The active levels of CS0 and CS1 for ROM and I/O select are a user programmable option. Either CS0 or CS1 may be programmed active high or active low, but different codes must be used for ROM or I/O select. CS0 and CS1 are mask programmed simultaneously with the ROM pattern. The ROM Select Circuitry is shown in Figure 16.

FIGURE 16 — ROM SELECT CIRCUITRY



TIMER OPERATION

The Timer may be programmed to operate in modes which fit a wide variety of applications. The device is fully bus compatible with the M6800 system and is accessed by Load and Store operations from the MPU.

In a typical application, the timer will be loaded by storing two bytes of data into the counter latch. This data is then transferred into the counter during a Counter Initialization cycle. If enabled, the counter decrements on each subsequent clock cycle (which may be E or an external clock) until one of several predetermined conditions causes it to halt or recycle. Thus, the timer is programmable, cyclic in nature, controllable by external inputs or MPU program, and accessible to the MPU at any time.

COUNTER LATCH INITIALIZATION

The Timer consists of a 16-bit addressable counter and two 8-bit addressable latches. The function of the latches is to store a binary equivalent of the desired count value minus one. Counter initialization results in the transfer of the latch contents of the counter. It should be noted that data transfer to the counters is always accomplished via the latches. Thus, the counter latches may be accurately described as a 16-bit "counter initilization data" storage register.

In some modes of operation, the initialization of the latches will cause simultaneous counter initialization (i.e., immediate transfer of the new latch data into the counters). It is, therefore, necessary to insure that all 16 bits of the latches are updated simultaneously. Since the MC6846 data bus is 8 bits wide, a temporary register (MSB Buffer Register) is provided for the Most Significant Byte of the desired latch data. This is a "write-only" register selected via address lines A0, A1, and A2. Data is transferred directly from the data bus to the MSB Buffer when the chip is selected, R/$\overline{W}$ is low, and the timer MSB register is selected (A0 = "0"; A1 = A2 = "1").

The lower 8 bits of the counter latch can also be referred to as a "write-only" register. Data Bus information will be transferred directly to the LSB of a counter latch when the chip is selected, R/$\overline{W}$ is low and the Timer LSB Register is selected (A0 = A1 = A2 = "1"). Data from the MSB Buffer will automatically be transferred into the Most Significant Byte of the counter latches simultaneously with the transfer of the Data Bus information to the Least Significant Byte of the Counter Latch. For brevity, the conditions for this operation will be referred to henceforth as a "Write Timer Latches Command."

The MC6846 has been designed to allow transfer of two bytes of data into the counter latches from any source, provided the MSB is transferred first. In many applications, the source of data will be an M6800 MPU. It should therefore be noted that the 16-bit store operations of the M6800 family microprocessors (STS and STX) transfer data in the order required by the MC6846. A Store Index Register instruction, for example, results in the MSB of the X register being transferred to the selected address, then the LSB of the X register being written into the next higher location. Thus, either the index register or stack pointer may be transferred directly into a selected counter latch with a single instruction.

A logic zero at the $\overline{\text{RESET}}$ input also initializes the counter latches. All latches will assume maximum count (65,535) values. It is important to note that an internal reset (bit zero of the Timer/Control Register Set) has no effect on the counter latches.

COUNTER INITIALIZATION

Counter Initialization is defined as the transfer of data from the latches to the counter with attendant clearing of the Individual Interrupt Flag associated with the counter. Counter Initialization always occurs when a reset condition (external $\overline{\text{RESET}}$ = "0" or TCR0 = "1") is recognized. It can also occur (dependent on The Timer Mode) with a Write Timer Latches command or recognition of a negative transition of the $\overline{\text{Gate}}$ input.

Counter recycling or reinitialization occurs when a clock input is recognized after the counter has reached an all-zero state. In this case, data is transferred from the Latches to the Counter, but the Interrupt Flag is unaffected.

TIMER CONTROL REGISTER

The Timer Control Register (see Table 2) in the MC6846 is used to modify timer operation to suit a variety of applications. The Timer Control Register has a unique address space (A0 = "1", A1 = "0", A2 = "1") and therefore may be written into at any time. The least significant bit of the Control Register is used as an internal reset bit. When this bit is a logic zero, all timers are allowed to operate in the modes prescribed by the remaining bits of the timer control register.

Writing "one" into **Timer Control Register B0 (TCR0)** causes the counter to be preset with the contents of the counter latches, all counter clocks are disabled, and the timer output and interrupt flag (Status Register) are reset. The Counter Latch and Timer/Control Register are undisturbed by an Internal Reset and may be written into regardless of the state of TCR0.

Timer Control Register Bit 1 (TCR1) is used to select the clock source. When TCR1 = "0", the external clock input $\overline{\text{CTC}}$ is selected, and when TCR1 = "1", the timer uses E.

Timer Control Register Bit 2 (TCR2) enables the divide-by-8 prescaler (TCR2 = "1"). In this mode, the clock frequency is divided by eight before being applied to the counter. When TCR2 = "0" the system clock is applied directly to the counter.

TCR3, 4, 5 select the Timer Operating Mode, and are discussed in the next section.

Timer Control Register Bit 6 (TCR6) is used to mask or enable the Timer Interrupt Request. When TCR6 = "0", the Interrupt Flag is masked from the timer. When TCR6 = "1", the Interrupt Flag is enabled into Bit 7 of the Composite Status Register (Composite IRQ Bit), which appears on the $\overline{\text{IRQ}}$ output pin.

Timer Control Register Bit 7 (TCR7) has a special function when the timer is in the Cascaded Single Shot mode. (This function is explained in detail in the section describing the mode.) In all other modes, TCR7 merely acts as an output enable bit. If TCR7 = "0", the Counter Timer Output (CTO) is forced low. Writing a logic one into TCR7 enables CTO. For more information on its operation, see the specific mode description.

TABLE 2 — FORMAT FOR TIMER/COUNTER CONTROL REGISTER (E)

CONTROL REGISTER BIT	STATE	BIT DEFINITION	STATE DEFINITION
TCR0	0	Internal Reset	Timer Enabled
	1		Timer in Preset State
TCR1	0	Clock Source	Timer uses External Clock ($\overline{\text{CTC}}$)
	1		Timer uses System Clock (E)
TCR2	0	÷ 8 Prescaler Enabler	Clock is not Prescaled
	1		Clock is prescaled by ÷ 8 Counter
TCR3	X	Operating Mode Selection	See Table 3
TCR4	X		
TCR5	X		
TCR6	0	Timer Interrupt Enable	$\overline{\text{IRQ}}$ Masked from Timer
	1		$\overline{\text{IRQ}}$ Enabled from Timer
TCR7	0	Timer Output Enable	Counter Output (CTO) Set LOW
	1		Counter Output Enabled

TIMER OPERATING MODES

The MC6846 has been designed to operate effectively in a wide variety of applications. This is accomplished by using three bits of the control register (TCR3, TCR4, and TCR5) to define different operating modes of the Timer, outlined in Table 3.

CONTINUOUS OPERATING MODE (TCR3=0, TCR5=0)

The timer may be programmed to operate in a continuous counting mode by writing zeros into bits 3 and 5 of the timer control register. Assuming that the timer output is enabled (TCR7="1"), a square wave will be generated at the Timer Output CTO (see Table 4).

Either a Timer Reset (TCR0="1" or External $\overline{\text{RESET}}$="0") condition or internal recognition of a negative transition of the $\overline{\text{CTG}}$ input results in Counter Initialization. A Write Timer Latches command can be selected as a Counter Initialization signal by clearing TCR4.

The discussion of the Continuous Mode has assumed the application requires an output signal. It should be noted the Timer operates in the same manner with the output disabled (TCR7="0"). A Read Timer Counter command is valid regardless of the state of TCR7.

TABLE 3 — OPERATING MODES

TCR3	TCR4	TCR5	Timer Operating Mode	Counter Initialization	Interrupt Flag Set
0	0	0	Continuous	$\overline{\text{CTG}}\downarrow + \text{W} + \text{R}$	T.O.
0	0	1	Cascaded Single Shot	$\overline{\text{CTG}}\downarrow + \text{R}$	T.O.
0	1	0	Continuous	$\overline{\text{CTG}}\downarrow + \text{R}$	T.O.
0	1	1	Normal Single Shot	$\overline{\text{CTG}}\downarrow + \text{R}$	T.O.
1	0	0	Frequency Comparison	$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}} \cdot (\text{W} + \text{T.O.}) + \text{R}$	$\overline{\text{CTG}}\downarrow$ Before T.O.
1	0	1		$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}} + \text{R}$	T.O. Before $\overline{\text{CTG}}\downarrow$
1	1	0	Pulse Width Comparison	$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}} + \text{R}$	$\overline{\text{CTG}}\uparrow$ Before T.O.
1	1	1			T.O. Before $\overline{\text{CTG}}\uparrow$

R = Reset Condition

W = Write Timer Latches

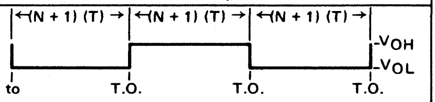
T.O. = Counter Time Out

$\overline{\text{CTG}}\downarrow$ = Negative Transition of Pin 17

$\overline{\text{CTG}}\uparrow$ = Positive Transition of Pin 17

$\overline{\text{I}}$ = Interrupt Flag (CSR0) = 0

TABLE 4 — CONTINUOUS OPERATING MODES

CONTINUOUS MODE (TCR3 = 0, TCR7 = 1, TCR5 = 0)			
CONTROL REGISTER		INITIALIZATION/OUTPUT WAVEFORMS	
TCR2	TCR4	Counter	Timer Output (2X)
0	0	Initialization $\overline{CTG}\downarrow + \overline{W} + R$	
0	1	$\overline{CTG}\downarrow + R$	

$\overline{CTG}$ = Negative Transition $\overline{GATE}$ Input.
$\overline{W}$ = Write Timer Latches Command.
R = Timer Reset (TCR0 = 1 or External $\overline{RESET}$ = 0)
N = 16 Bit Number in Counter Latch.

T = Period of Clock Input to Counter.
to = Counter Initialization Cycle.
T.O. = Counter Time Out (All Zero Condition).

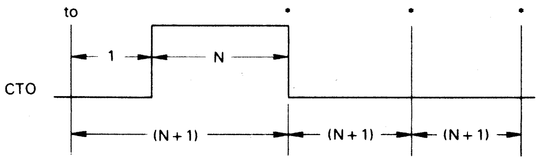
NORMAL SINGLE-SHOT TIMER MODE (TCR3 = 0, TCR4 = 1, TCR5 = 1)

This mode is identical to the Continuous Mode with two exceptions. The first of these is obvious from the name — the output returns to a low-level after the initial Time Out and remains low until another Counter Initialization cycle occurs. The output waveform (CTO) is shown in Figure 17.

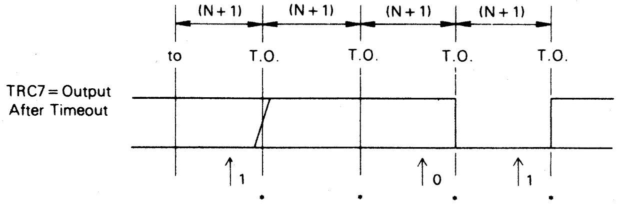
The internal counting mechanism remains cyclical in the Single-Shot Mode. Each Time Out of the counter results in the setting of an Individual Flag and re-initialization of the counter.

The second major difference between the Single-Shot and Continuous modes is that the internal counter enable is not dependent on the $\overline{CTG}$ input level remaining in the low state for the Single-Shot mode. Aside from these differences, the two modes are identical.

FIGURE 17 — SINGLE-SHOT MODES



(A) Normal Single-Shot Mode Output Waveform



(B) Cascaded Single-Shot Mode Output Waveform

1 = Write a "1" into TCR-7
0 = Write a "0" into TCR-7

*Point at which an interrupt may occur.

NOTE: All time intervals shown above assume the Gate (CTG) and Clock (CTC) signals are synchronized to E with the specified setup and hold time requirements.

TABLE 5 — TIME INTERVAL MODES

TCR3 = 1			
TCR4	**TCR5**	**APPLICATION**	**CONDITION FOR SETTING INDIVIDUAL INTERRUPT FLAG**
0	0	Frequency Comparison	Interrupt Generated if $\overline{CTG}$ Input Period (1/F) is Less Than Counter Time Out (T.O.)
0	1	Frequency Comparison	Interrupt Generated if $\overline{CTG}$ Input Period (1/F) is Greater Than Counter Time Out (T.O.)
1	0	Pulse Width Comparison	Interrupt Generated if $\overline{CTG}$ Input "Down Time" is Less Than Counter Time Out (T.O.)
1	1	Pulse Width Comparison	Interrupt Generated if $\overline{CTG}$ Input "Down Time" is Greater Than Counter Time Out (T.O.)

TIME INTERVAL MODES (TCR3 = 1)

The Time Interval Modes are provided for applications requiring more flexibility of interrupt generation and Counter Initialization. The Interrupt Flag is set in these modes as a function of both Counter Time Out and transistions of the CTG input. Counter Initialization is also affected by Interrupt Flag status. The output signal is not defined in any of these modes. Other features of the Time Interval Modes are outlined in Table 5.

CASCADED SINGLE-SHOT MODE (TCR3 = 0, TCR4 = 0, TCR5 = 1)

This mode is identical to the single-shot mode with two exceptions. First, the output waveform does not return to a low level and remain low after timeout. Instead, the output levels remains at its initialized level until it is re-programmed and changed by timeout. The output level may be changed at any timeout or may have any number of timeouts between changes.

The second difference is the method used to change the output level. Timer Control Register Bit 7 (TCR7) has a special function in this mode. The timer output (CTO) is equal to TCR7 clocked by timeout. At every timeout, the contents of TCR7 is clocked to and held at the CTO output. Thus, output pulses of length greater than one timer cycle can be generated by cascading timer cycles and counting timeouts with a software program. (See Figure 17.)

An interrupt is generated at each timeout. To cascade timer cycles, the MPU would need an interrupt routine to: 1) count each timeout and determine when to change TCR7; 2) write into TCR7 the state corresponding to the next desired state of the output waveform (only necessary during the last timer cycle before the output is to change state); and 3) clear the interrupt flag by reading the combination status register followed by Read Timer MSB. It is also possible, if desired, to change the length of the timer cycle by reinitializing the timer latches. This allows more flexibility for obtaining desired times.

FREQUENCY COMPARISON MODE (TCR3 = 1, TCR4 = 0)

The timer within the MC6846 may be programmed to compare the period of a pulse (giving the frequency after calculations) at the CTG input with the time period required for Counter Time Out. A negative transistion of the CTG input enables the counter and starts a Counter Initialization cycle — provided that other conditions, as noted in Table 6, are satisfied. The counter decrements on each clock signal recognized during or after Couter Initialization until an Interrupt is generated, a Write Timer Latches command is issued, or a Timer Reset condition occurs. It can be seen from Table 6 that an interrupt condition will be generated if TCR5 = "0" and the period of the pulse (single pulse or measured separately repetative pulses) at the CTG input is less than the Counter Time Out period. If TCR5 = "1", an interrupt is generated if the reverse is true.

Assume now with TCR5 = "1" that a Counter Initialization has occurred and that the CTG input has returned low prior to Counter Time Out. Since there is no Individual Interrupt Flag generated, this automatically starts a new Counter Initialization Cycle. The process will continue with frequency comparison being performed on each CTG input cycle until the mode is changed, or a cycle is determined to be above the predetermined limit.

TABLE 6 — FREQUENCY COMPARISON MODE

CRX3 = 1, CRX4 = 0				
Control Reg Bit 5 (CRX5)	**Counter Initialization**	**Counter Enable Flip-Flop Set (CE)**	**Counter Enable Flip-Flop Reset (CE)**	**Interrupt Flag Set (I)**
0	$\overline{G}\downarrow\cdot\overline{I}\cdot(\overline{CE}+TO\cdot CE)+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	$W+R+I$	G↓ Before TO
1	$\overline{G}\downarrow\cdot\overline{I}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	$W+R+I$	TO Before G↓

I represents the interrupt for the timer.

TABLE 7 — PULSE WIDTH COMPARISON MODE

CRX3 = 1, CRX4 = 1				
Control Reg Bit 5 (CRX5)	**Counter Initialization**	**Counter Enable Flip-Flop Set (CE)**	**Counter Enable Flip-Flop Reset (CE)**	**Interrupt Flag Set (I)**
0	$\overline{G}\downarrow\cdot\overline{I}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	$W+R+I+G$	$\overline{G}\uparrow$ Before TO
1	$\overline{G}\downarrow\cdot\overline{I}+R$	$\overline{G}\downarrow\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	$W+R+I+G$	TO Before $\overline{G}\uparrow$

PULSE WIDTH COMPARISON MODE (TCR3 = 1, TCR4 = 1)

This mode is similar to the Frequency Comparison Mode except for the limiting factor being a positive, rather than negative, transition of the CTG input. With TCR5 = "0", an Individual Interrupt Flag will be generated if the zero level pulse applied to the CTG input is less than the time period required for Counter Time Out. With TCR5 = "1", the interrupt is generated when the reverse condition is true.

As can be seen in Table 7, a positive transition of the CTG input disables the counter. With TCR5 = "0", it is therefore possible to directly obtain the width of any pulse causing an interrupt.

DIFFERENCES BETWEEN THE MC6840 AND THE MC6846 TIMERS

1) Control registers 1 and 3 are buried (access through control register 2 only) in the MC6840 timer. In the MC6846, all registers are directly accessable.

2) The MC6840 has a dual 8-bit continuous mode for generating non-symmetrical waveforms. The MC6846, instead, has a cascaded one shot mode which can accomplish the same function, but also allows the user to generate waveforms longer than one timeout.

3) Because of the different modes, there is a difference in the control registers between the MC6840 and the MC6846.

COMPOSITE STATUS REGISTER

The Composite Status Register (CSR) is a read-only register which is shared by the Timer and the Peripheral Data Port of the MC6846. Three individual interrupt flags in the register are set directly via the appropriate conditions in the timer or peripheral port. The composite interrupt flag — and the IRQ Output — respond to these individual interrupts only if corresponding enable bits are set in the appropriate Control Registers. (See Figure 18.) The sequence of assertion is not detected. Setting TCR6 while CSR0 is high will cause CSR7 to be set, for example.

The Composite Interrupt Flag (CSR7) is clear only if all enabled Individual Interrupt Flags are clear. The conditions for clearing CSR1 and CSR2 are detailed in a later section. The Timer Interrupt Flag (CSR0) is cleared under the following conditions:

1) Timer Reset — Internal Reset Bit (TCR0) = "1" or External $\overline{\text{RESET}}$ = "0"

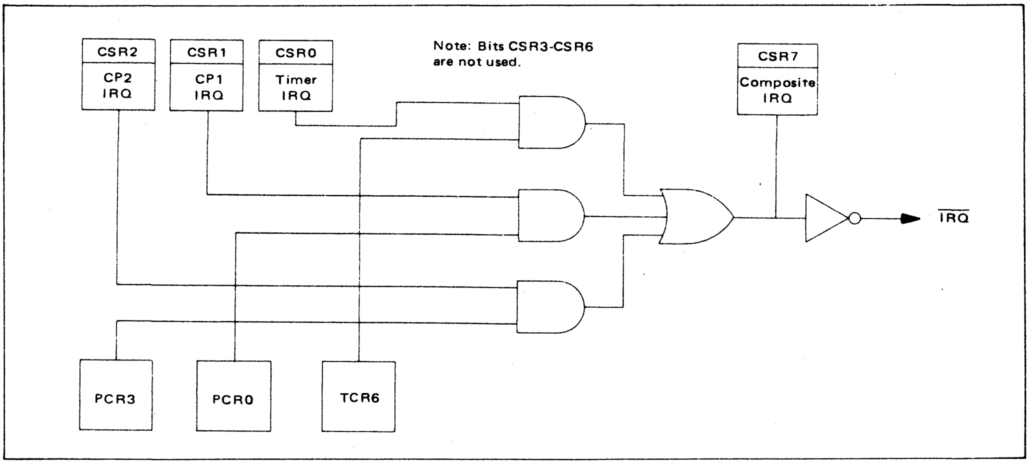
2) Any Counter Initialization condition.

3) A Write Timer Latches command if Time Interval modes (TCR3 = "1") are being used.

4) A Read Timer Counter command, provided this is preceded by a Read Composite Status Register while CSR0 is set. This latter condition prevents missing an Interrupt Request generated after reading the Status Register and prior to reading the counter.

The remaining bits of the Composite Status Register (CSR3-CSR6) are unused. They return a logic zero when read.

FIGURE 18 — COMPOSITE STATUS REGISTER AND ASSOCIATED LOGIC



I/O OPERATION

PARALLEL PERIPHERAL PORT

The peripheral port of the MC6846 contains eight Peripheral Data lines (P0-P7), two Peripheral Control lines (CP1 and CP2), a Data Direction Register, a Peripheral Data Register, and a Peripheral Control Register. The port also directly affects two bits (CSR1 and CSR2) of the Composite Status Register.

The Peripheral Port is similar to the "B" side of a PIA (MC6820 or MC6821) with the following exceptions:

1) All registers are directly accessible in the MC6846. Data Direction and Peripheral Data in the MC6820/6821 are located at the same address, with Bit Two of the Control Register used for register selection.

2) Peripheral Control Register Bit Two (PCR2) of the MC6846 is used to select an optional input latch function. This option is not available with MC6820/6821 PIA's.

3) Interrupt Flags are located in the MC6846 composite status register rather than Bits 6 and 7 of the Control Register as used in the MC6820/MC6821.

4) Interrupt Flags are cleared in the MC6820/6821 by reading data from the Peripheral Data Register. MC6846 Interrupt Flags are cleared by either reading or writing to the Peripheral Data Register — provided that this sequence is followed a) Flag Set, b) Read Composite Status Register, c) Read/Write Peripheral Data Register is followed.

5) Bit 6 of the MC6846 Peripheral Control Register is not used. Bit 7 (PCR7) is an Internal Reset Bit not available on the MC6820/6821.

6) The Peripheral Data lines (and CP2) of the MC6846 feature internal current limiting which allows them to directly drive the base of Darlington NPN transistors.

DATA DIRECTION REGISTER

The MPU can write directly to this 8-bit register to configure the Peripheral Data lines as either inputs or outputs. A particular bit within the register (DDRN) is used to control the corresponding Peripheral Data line (PN). With DDRN = "0", PN becomes an input; if DDRN = "1", PN is an output. As an example, writing Hex $0F into the Data direction Register results in P0 through P3 becoming outputs and P4 through P7 being inputs. Hex $55 in the Data direction Register results in alternate outputs and inputs at the parallel port.

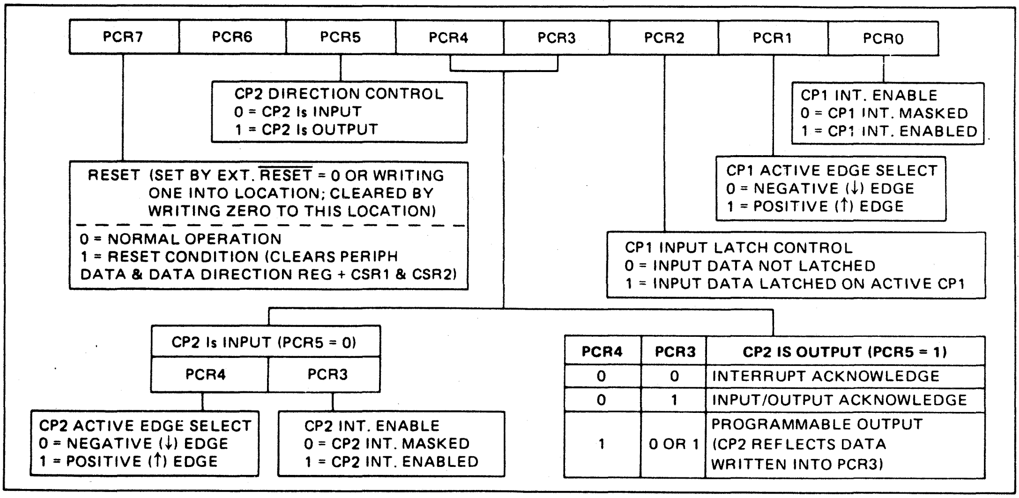
PERIPHERAL DATA REGISTER

This 8-bit register is used for transferring data between the peripheral data port and the MPU. Any bit corresponding to an output line will be used to drive the output buffer associated with that line. Data in these output bits is normally provided by an MPU Write function. (Input bits — those associated with input lines — are unchanged by a Write Command.) Any input bit will reflect the state of the associated input line if the input latch function is deselected. If the Control Register is programmed to provide input latching, the input bit will retain the state at the time CP1 was activated until the Peripheral Data Register is read by the MPU.

PERIPHERAL CONTROL REGISTER

This 8-bit register is used to control the reset function as well as for selection of optional functions of the two peripheral control lines (CP1 and CP2). The Peripheral Control Register functions are outlined in Table 8.

TABLE 8 — PERIPHERAL CONTROL REGISTER FORMAT (EXPANDED)



PCR4	PCR3	CP2 IS OUTPUT (PCR5 = 1)
0	0	INTERRUPT ACKNOWLEDGE
0	1	INPUT/OUTPUT ACKNOWLEDGE
1	0 OR 1	PROGRAMMABLE OUTPUT (CP2 REFLECTS DATA WRITTEN INTO PCR3)

PERIPHERAL PORT RESET (PCR7)

Bit 7 of the Peripheral Control Register (PCR7) may be used to initialize the peripheral section of the MC6846. When this bit is set high, the peripheral data register, the peripheral data direction register, and the interrupt flags associated with the peripheral port (CSR1 and CSR2) are all cleared. Other bits in the peripheral control register are not affected by PCR7.

PCR7 is set by either a logic zero at the External $\overline{RESET}$ input or under program control by writing a "one" into the location. In any case, PCR7 may be cleared only by writing a "zero" into the location while $\overline{RESET}$ is high. The bit must be cleared to activate the port.

CONTROL OF CP1 PERIPHERAL CONTROL LINE

CP1 may be used as an interrupt request to the MC6846, as a strobe to allow latching of input data, or both. In any case, the input can be programmed to be activated by either a positive or negative transition of the signal. These options are selected via Control Register Bits PCR0, PCR1, and PCR2.

Control Register Bit 0 (PCR0) is used to enable the interrupt transfer circuitry of the MC6846. Regardless of the state of PCR0, an active transition of CP1 causes the Composite Status Register Bit One (CSR1) to be set. If PCR0 = "1", this interrupt will be reflected in the Composite Interrupt Flag (CSR7), and thus at the $\overline{IRQ}$ output. CSR1 is cleared by a Peripheral Port Reset condition or by either reading or writing to the peripheral data register after the Composite Status Register was last read. This precludes inadvertent clearing of interrupt flags generated between the time the Status Register is read and the manipulation of peripheral data.

Control Register Bit One (PCR1) is used to select the edge which activates CP1. When PCR1 = "0", CP1 is active on negative transitions (high-to-low). Low-to-high transitions are sensed by CP1 when PCR1 = "1".

In addition to its use as an interrupt input, CP1 can be used as a strobe to capture input data in an internal latch. This option is selected by writing a "one" into Peripheral Control Register Bit Two (PCR2). In operation, the data at the pins designated by the Data Direction Register as inputs will be captured by an active transition of CP1. An MPU Read of the Peripheral Data Register will result in the captured data being transferred to the MPU — and it also releases the latch to allow capture of new data. Note that successive active transistions with no Read Peripheral Data Command between does not update the input latch. Also, it should be noted that use of the input latch function (which can be deselected by writing a zero into PCR2) has no effect on output data. It also does not affect Interrupt function of CP1.

CONTROL OF CP2 PERIPHERAL CONTROL LINE

CP2 may be used as an input by writing a zero into PCR5. In this configuration, CP2 becomes a dual of CP1 in regard to generation of interrupts. An active transition (as selected by PCR4) causes Bit Two of the Composite Status Register to be set. PCR3 is then used to select whether the CP2 transition is to cause CSR7 to be set — and thereby cause $\overline{IRQ}$ to go low. CP2 has no effect on the input latch function of the MC6846.

Writing a one into PCR5 causes CP2 to function as an output. PCR4 then determines whether CP2 is to be used in a handshake or programmable output mode. With PCR4 = "1", CP2 will merely reflect the data written into PCR3. Since this can readily be changed under program control, this mode allows CP2 to be a programmable output line in much the same manner as those lines selected as outputs by the Data Direction Register.

The handshaking mode (PCR5 = "1", PCR4 = "0") allows CP2 to perform one of two functions as selected by PCR3. With PCR3 = "1", CP2 will go low on the first positive E transition. This Input/Output Acknowledge signal is released (returns high) on the next positive transition of E.

In the Interrupt Acknowledge mode (PCR5 = "1", PCR4 = PCR3 = "0"), CP2 is set when CSR1 is set by an active transition of CP1. It is released (goes low) on the first positive transition of E after CSR1 has been cleared via an MPU Read or Write to the Peripheral Data Register. (Note that the previously described conditions for clearing CSR1 still apply.)

RESET SEQUENCE

A typical reset sequence for the MC6846 will include initialization of both the Peripheral Control and Data Direction Registers of the parallel port. It is necessary to set up the Peripheral Control Register first, since PCR7 = "0" is a condition for writing data into the Data Direction Register. (A logic zero at the external $\overline{RESET}$ input automatically sets PCR7.)

SUMMARY

The MC6846 has several optional modes of operation which allow it to be used in a variety of applications. The following tables are provided for reference in selecting these modes.

TABLE 9 — MC6846 INTERNAL REGISTER ADDRESSES

A2	A1	A0	REGISTER SELECTED
0	0	0	Combination Status Register
0	0	1	Peripheral Control Register
0	1	0	Data Direction Register
0	1	1	Peripheral Data Register
1	0	0	Combination Status Register
1	0	1	Timer Control Register
1	1	0	Timer MSB Register
1	1	1	Timer LSB Register
X	X	X	ROM Address

TABLE 10 — COMPOSITE STATUS REGISTER

CSR7	CSR3-CSR6 NOT USED. DEFAULT TO ZERO WHEN READ	CSR2	CSR1	CSR0
COMPOSITE INTERRUPT FLAG 0 = NO ENABLED INTERRUPT FLAG SET 1 = ONE OR MORE ENABLED INTERRUPT FLAGS SET.* INVERSE OF THIS BIT APPEARS AT $\overline{\text{IRQ}}$ OUTPUT *STATUS OF THIS BIT CAN BE EXPRESSED AS: CSR7 = CSR0 · TCR6 + CSR1 · PCR0 + CSR2 · PCR3		CP2 INTERRUPT FLAG 0 = NO INT REQ 1 = INT REQUESTED	CP1 INTERRUPT 0 = NO INT REQ. 1 = INT REQUESTED	TIMER INTERRUPT FLAG 0 = NO INT REQ. 1 = INT REQUESTED

TABLE 11 — TIMER CONTROL REGISTER

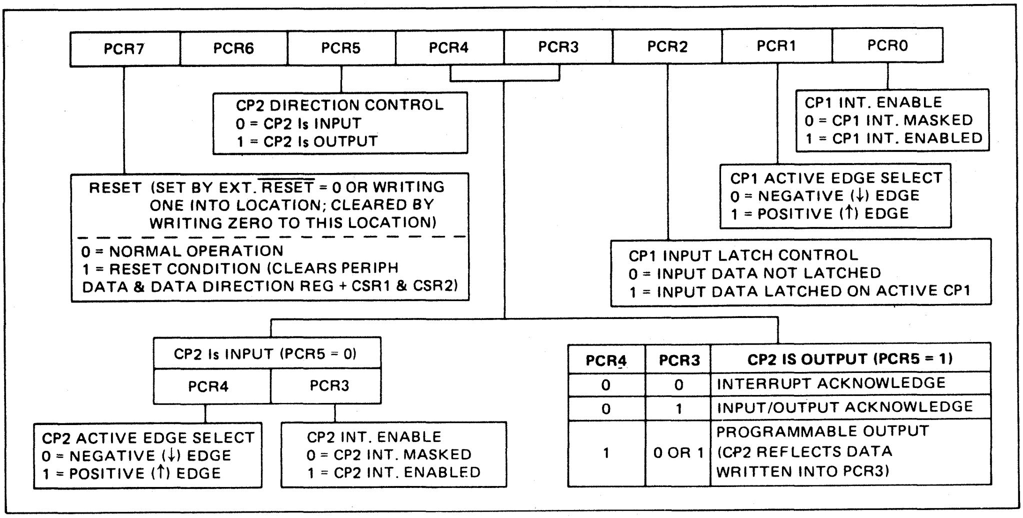
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
TIMER OUTPUT ENABLE 0 = OUTPUT DISABLED (LOW) 1 = OUTPUT ENABLED FOR CASCADED SINGLE-SHOT 0 = OUTPUT GOES LOW AT TIME OUT 1 = OUTPUT GOES HIGH AT TIME OUT	INTERRUPT ENABLE 0 = $\overline{\text{IRQ}}$ MASKED 1 = $\overline{\text{IRQ}}$ ENABLED				÷ 8 PRESCALE ENABLE 0 = CLOCK NOT PRESCALED 1 = CLOCK PRESCALED (÷ 8)	CLOCK SOURCE 0 = EXTERNAL CLOCK (CTC) 1 = INTERNAL CLOCK (ϕ2)	INTERNAL RESET 0 = TIMER ENABLED 1 = RESET STATE

TCR3	TCR4	TCR5	TIMER OPERATING MODE	COUNTER INITIALIZATION	INTERRUPT FLAG SET
0	0	0	CONTINUOUS	$\overline{\text{CTG}}\downarrow$ + W + R	T.O.
0	0	1	CASCADED SINGLE SHOT	$\overline{\text{CTG}}\downarrow$ + R	T.O
0	1	0	CONTINUOUS	$\overline{\text{CTG}}\downarrow$ + R	T.O.
0	1	1	NORMAL SINGLE SHOT	$\overline{\text{CTG}}\downarrow$ + R	T.O.
1	0	0	FREQUENCY COMPARISON	$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}} \cdot$ (W + T.O.) + R	$\overline{\text{CTG}}\downarrow$ BEFORE T.O.
1	0	1		$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}}$ + R	T.O. BEFORE $\overline{\text{CTG}}\downarrow$
1	1	0	PULSE WIDTH COMPARISON	$\overline{\text{CTG}}\downarrow \cdot \overline{\text{I}}$ + R	$\overline{\text{CTG}}\uparrow$ BEFORE T.O.
1	1	1			T.O. BEFORE $\overline{\text{CTG}}\uparrow$

R = RESET CONDITION
W = WRITE TIMER LATCHES
T.O. = COUNTER TIME OUT

$\overline{\text{CTG}}\downarrow$ = NEG TRANSITION OF PIN 17
$\overline{\text{CTG}}\uparrow$ = POS TRANSITION OF PIN 17
$\overline{\text{I}}$ = INTERRUPT FLAG (CSR0) = 0

TABLE 12 — PERIPHERAL CONTROL REGISTER



PCR4	PCR3	CP2 IS OUTPUT (PCR5 = 1)
0	0	INTERRUPT ACKNOWLEDGE
0	1	INPUT/OUTPUT ACKNOWLEDGE
1	0 OR 1	PROGRAMMABLE OUTPUT (CP2 REFLECTS DATA WRITTEN INTO PCR3)

CUSTOM PROGRAMMING*

By the programming of a single photomask for the MC6846, the customer may specify the content of the memory and the method of enabling the outputs.

Information on the general options of the MC6846 should be submitted on an Organizational Data form such as that shown in Figure 19.

Information for custom memory content may be sent to Motorola in one of two forms (shown in order of preference):

1. EPROMs
2. 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 to:

Motorola Inc.
MPU Marketing L2787
3501 Ed Bluestein Blvd.
Austin, Texas 78721

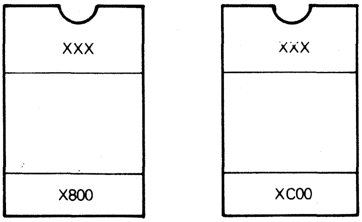
A copy of the cover letter should also be mailed separately.

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 ($X800-$XFFF). See Figure A-1 for recommended marking procedure.

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

FIGURE A-1



XX = Customer ID

MDOS DISKETTE

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

---

*Motorola provides two ROM patterns in the MC6846:

1. MIKBUG 2.0 — MC6846L1,P1
2. TVBUG 1.2 — MC6846L3,P3

FIGURE 19 — FORMAT FOR PROGRAMMING GENERAL OPTIONS

ORGANIZATIONAL DATA
MC6846 COMBINATION ROM-I/O-TIMER

Customer:

Company ______________________________

Part No. ______________________________

Originator ______________________________

Phone No. ______________________________

Motorola Use Only:

Quote: ______________________________

Part No.: ______________________________

Specif. No.: ______________________________

Enable Options: (ROM ENABLE MUST DIFFER FROM I/O-TIMER)

	1	0	1	0
CS0	☐	☐	☐	☐
CS1	☐	☐	☐	☐
	ROM SECTION		I/O-TIMER SECTION	

		CHECK ONE COLUMN ONLY				
I/O-TIMER SELECT		☐	☐	☐	☐	☐
A6	A10	X	1	X	X	X
1 0 X	A9	X	X	1	X	X
☐ ☐ ☐	A8	X	X	X	1	X
	A7	X	X	X	X	1

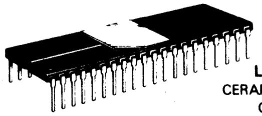
1 ≥ 2.0V.
0 ≤ 0.8V.
X = NOT USED

MOTOROLA

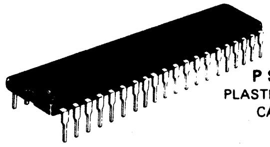
MC6847
Non-Interlace
MC6847Y
Interlace

MOS
(N-CHANNEL, SILICON-GATE)

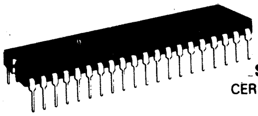
VIDEO DISPLAY GENERATOR



L SUFFIX
CERAMIC PACKAGE
CASE 715

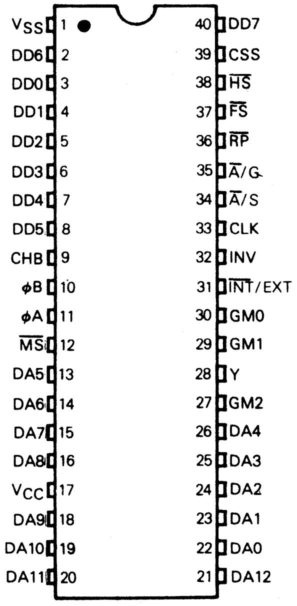


P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT



Pin	Name	Pin	Name
1	V_{SS}	40	DD7
2	DD6	39	CSS
3	DD0	38	$\overline{HS}$
4	DD1	37	$\overline{FS}$
5	DD2	36	$\overline{RP}$
6	DD3	35	$\overline{A}$/G
7	DD4	34	$\overline{A}$/S
8	DD5	33	CLK
9	CHB	32	INV
10	ϕB	31	$\overline{INT}$/EXT
11	ϕA	30	GM0
12	$\overline{MS}$	29	GM1
13	DA5	28	Y
14	DA6	27	GM2
15	DA7	26	DA4
16	DA8	25	DA3
17	V_{CC}	24	DA2
18	DA9	23	DA1
19	DA10	22	DA0
20	DA11	21	DA12

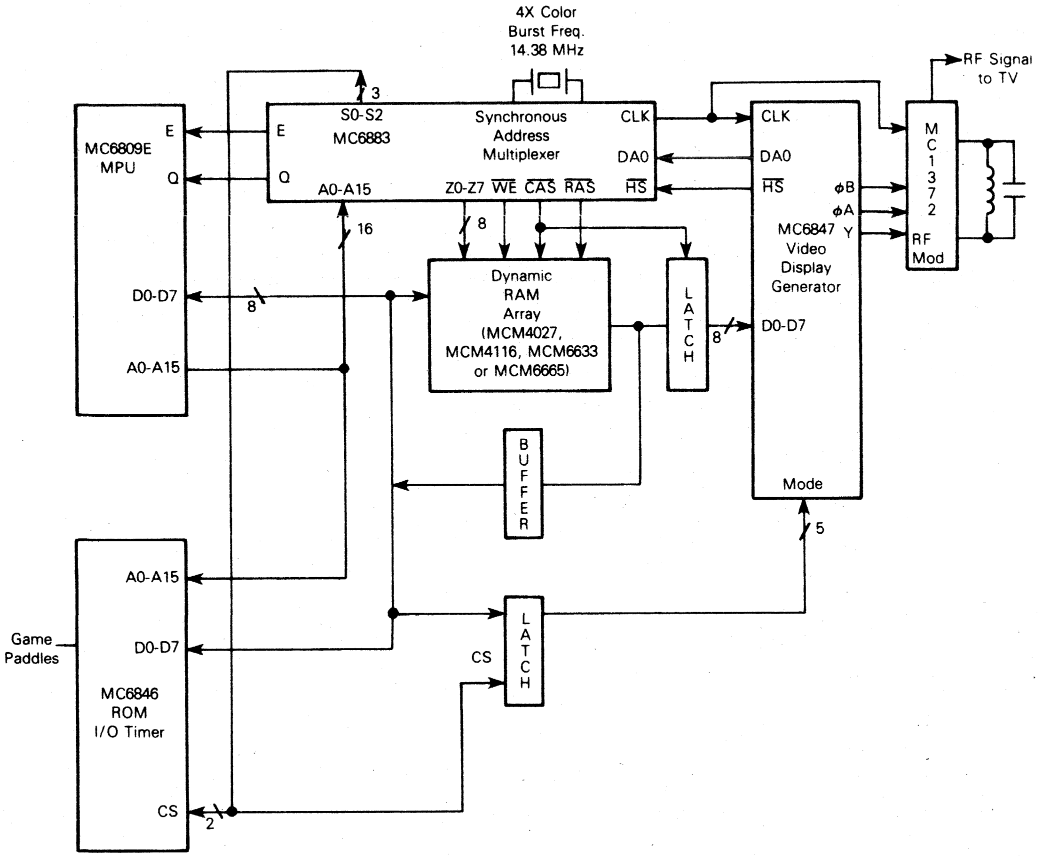
MC6847/MC6847Y VIDEO DISPLAY GENERATOR (VDG)

The video display generator (VDG) provides a means of interfacing the M6800 microprocessor family (or similar products) to a standard color or black and white NTSC television receiver. Applications of the VDG include video games, process control displays, home computers, education, communications, and graphics applications.

The VDG reads data from memory and produces a video signal which will allow the generation of alphanumeric or graphic displays. The generated video signal may be modulated to either channel 3 or 4 by using the compatible MC1372 (TV chroma and video modulator). This modulated signal is suitable for reception by a standard unmodified television receiver. A typical TV game is shown in Figure 1.

- Compatible with the M6800 Family, the M68000 Family, and Other Microprocessor Families
- Generates Four Different Alphanumeric Display Modes, Two Semigraphic Modes, and Eight Graphic Display Modes
- The Alphanumeric Modes Display 32 Characters Per Line by 16 Lines Using Either the Internal ROM or an External Character Generator
- Alphanumeric and Semigraphic Modes May Be Mixed on a Character-by-Character Basis
- Alphanumeric Modes Support Selectable Inverse on a Character-by-Character Basis
- Internal ROM May Be Mask Programmed with a Custom Pattern
- Full Graphic Modes Offer 64 × 64, 128 × 64, 128 × 96, 128 × 192, or 256 × 192 Densities
- Full Graphic Modes Use One of Two 4-Color Sets or One of Two 2-Color Sets
- Compatible with the MC1372 and MC1373 Modulators Via Y, R-Y (ϕA), and B-Y (ϕB) Interface
- Compatible with the MC6883 (74LS783) Synchronous-Address Multiplexer
- Available in Either an Interlace (NTSC Standard) or Non-interlace Version

FIGURE 1 — BLOCK DIAGRAM OF A TV GAME USING THE VDG AND THE MC6809E MPU



ELECTRICAL SPECIFICATIONS
ABSOLUTE MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage Any Pin	V_{in}	−0.3 to +7.0	V
Operating Temperature	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−65 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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Plastic		100	
Cerdip		60	

DC (STATIC) CHARACTERISTICS (V_{CC}=5.0 V ±5%, V_{SS}=0.0 V, T_A=0°C to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage CLK Other Inputs	V_{IH}	 $V_{SS}+2.4$ $V_{SS}+2.0$	 – –	 V_{CC} V_{CC}	V
Input Low Voltage CLK Other Inputs	V_{IL}	 $V_{SS}-0.3$ $V_{SS}-0.3$	 – –	 $V_{SS}+0.4$ $V_{SS}+0.8$	V
Input Leakage Current, Force 5.25 V on Pin Under Test, V_{CC}=5.5 V CLK, GM0-GM2, INV, $\overline{\text{INT}}$/EXT, $\overline{\text{MS}}$, V_{SS}, DD0-DD7, $\overline{\text{A}}$/S, $\overline{\text{A}}$/G	I_{in}	–	–	2.5	µA
Three-State (Off State) Input Current DA0-DA12 Force 2.4 V and 0.4 V on Pin Under Test	I_{OL}	–	–	±10	µA
Output High Voltage (C_{Load}=30 pF, I_{Load}=−100 µA $\overline{\text{RP}}$, $\overline{\text{HS}}$, $\overline{\text{FS}}$	V_{OH}	2.4	–	–	V
Output High Voltage (C_{Load}=55 pF, I_{Load}=−100 µA) DA0-DA12	V_{OH}	2.4	–	–	V
Output Low Voltage (C_{Load}=30 pF, I_{Load}=1.6 mA) RP, HS, FS	V_{OL}	–	–	$V_{SS}+0.4$	V
Output Low Voltage (C_{Load}=55 pF, I_{Load}=1.6 mA) DA0-DA12	V_{OL}	–	–	$V_{SS}+0.4$	V
Output High Current (Sourcing) (V_{OH}=2.4 V) All Outputs (Except φA, φB, Y, and CHB)	I_{OH}	−100	–	–	µA
Output Low Current (Sinking) (V_{OL}=0.4 V) All Outputs (Except φA, φB, Y, and CHB)	I_{OL}	1.6	–	–	mA
Input Capacitance (V_{in}=0, T_A=25°C, f=1.0 MHz) All Inputs	C_{in}	–	–	7.5	pF
Internal Power Dissipation (Measured at T_A=0 to 70°C)	P_{INT}	–	–	600	mW
Chroma φA Voltage (Figure 3) (C_{Load}=20 pF, R_{Load}=100 kΩ) (Note 1)	V_{IH} V_R V_{OL}	1.8 1.34 0.8	2.0 1.5 1.0	2.2 1.66 1.2	V
Chroma φB Voltage (Figure 3) (C_{Load}=20 pF, R_{Load}=100 kΩ) (Note 1)	V_{IH} V_R V_{OL} V_{Burst}	1.8 1.34 0.80 1.07	2.0 1.5 1.0 1.25	2.2 1.66 1.2 1.43	V
Luminance Y Voltage (Figure 3) (C_{Load}=20 pF, R_{Load}=100 kΩ) (Voltage Synchronization) (Voltage Blank) (Voltage Black) (Voltage White Low) (Voltage White Medium) (Voltage White High) (Note 1)	 V_S V_{Blank} V_{Black} V_{WL} V_{WM} V_{WH}	 0.9 0.63 0.58 0.51 0.40 0.27	 1.0 0.77 0.72 0.65 0.54 0.42	 1.1 0.9 0.83 0.75 0.65 0.53	V
Chroma Bias Voltage (C_{Load}=20 pF, R_{Load}=100 kΩ)	V_R	0.27 V_{CC}	0.3 V_{CC}	0.33 V_{CC}	V
Resistor % of V_{SS} Tracking (Analog Outputs Linearity Error)	R_T	–	1.0	3.0	%

NOTE 1: The specified minimum and maximum number reflect performance of the VDG of the specified temperature range. Overlapping voltage levels will not occur. Refer to Figure 2.

POWER CONSIDERATIONS

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

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

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:

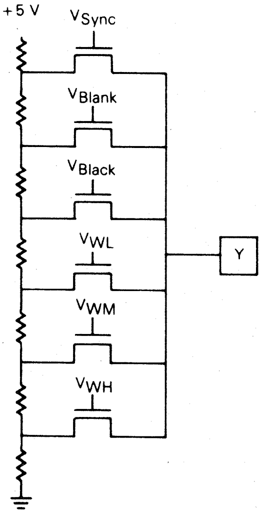
$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

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

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.

FIGURE 2 – PSEUDO ANALOG LUMINANCE RESISTOR CHAIN



NOTE: The chrominance output chain is similar in design to the luminance chain.

AC (DYNAMIC) CHARACTERISTICS (V_{CC} = 5.0 V ±5%, T_A = 0°C to 70°C) (Load Circuit of Figure 3)

Characteristic		Symbol	Min	Max	Unit	Figure
CLK (Frequency (3.579545 Color Burst Frequency)		f	3.579535	3.597555	MHz	4
CLK Duty Cycle		CLK_{dc}	45	55	%	4
Clock Rise Time		t_{CLKr}	–	50	ns	4
Clock Fall Time		t_{CLKf}	–	50	ns	4
Clock Pulse Width		PW_{CLK}	120	160	ns	4
Horizontal Display Address Delay from Counter	DA0-DA3	t_{HDAD}	–	490	ns	4, 5, 6
	DA4	t_{HDA4D}	–	550	ns	5, 6
Horizontal Display Address Hold Time		t_{HDAH}	0	–	ns	4, 5, 6
		t_{HDA4H}	0	–		
Display Data Setup Time	CSS, INV, $\overline{A}$/S, $\overline{INT}$/EXT, DD0-DD7	t_{DDS}	70	–	ns	4, 5, 6
Display Data Hold Time	CSS, INV, $\overline{A}$/S, $\overline{INT}$/EXT, DD0-DD7	t_{DDH}	140	–	ns	4, 5, 6
Horizontal Sync ($\overline{HS}$) Delay	Fall	t_{DHSf}	–	550	ns	7
	Rise	t_{DHSr}	–	740		
Row Preset ($\overline{RP}$) Delay	Fall	t_{DRPf}	–	660	ns	7
	Rise	t_{DRDr}	–	540		
Vertical Display Address Delay from Counter	DA5-DA12	t_{VDAD}	–	6.0	µs	7
Vertical Display Address Hold Time		t_{VDAH}	–	220	ns	7
Field Sync ($\overline{FS}$) Delay	Fall	t_{DFSf}	–	520	ns	8
	Rise	t_{DFSr}	–	600		
Memory Select Low to Display Address High-Impedance		t_{DMST}	–	80	ns	9
Memory Select High to Display Address Valid		t_{DMSV}	–	400	ns	9
Chroma Rise and Fall Times						
(φA Rise Time)		$t_{rC\phi A}$	–	100	ns	12
		$t_{r\phi A}$	–	100		
(φA Fall Time)		$t_{fC\phi A}$	–	100		
		$t_{f\phi A}$	–	100		
(φB Rise Time)		$t_{rC\phi B}$	–	100		
		$t_{r\phi B}$	–	100		
(φB Fall Time)		$t_{fC\phi B}$	–	100		
		$t_{f\phi B}$	–	100		
Color Burst Rise Time on φB Output		t_{CBr}	–	100	ns	12
Color Burst Fall Time on φB Output		t_{CBf}	–	100	ns	12
Chroma Phase Delay (Measured with Respect to "Y" Output)						
φA		t_{YA}	−50	140	ns	11
φB		t_{YB}	−50	140		
Luminance Rise Time		t_{ry}	–	100	ns	12
Luminance Fall Time		t_{fy}	–	100	ns	12
Horizontal Sync Rise Time on Y Output		t_{Hr}	–	100	ns	12
Horizontal Sync Fall Time on Y Output		t_{Hf}	–	100	ns	12
Horizontal Blanking Rise Time on Y Output		t_{HBr}	–	100	ns	12
Horizontal Blanking Fall Time on Y Output		t_{HBf}	–	100	ns	12
Front Porch Duaration Time (7 × 1/f)		t_{FP}	1.8	2.4	µs	12
Back Porch Duration Time (17.5 × 1/f)		t_{BP}	4.5	5.1	µs	12
Left Border Duration Time (29.5 × 1/f)		t_{LB}	7.5	8.3	µs	12
Right Border Duration Time (28 × 1/f)		t_{RB}	7.5	8.3	µs	12
Color Burst Duration Time (10.5 × 1/f)		t_{CB}	2.7	3.2	µs	12

FIGURE 3 — TEST LOADS

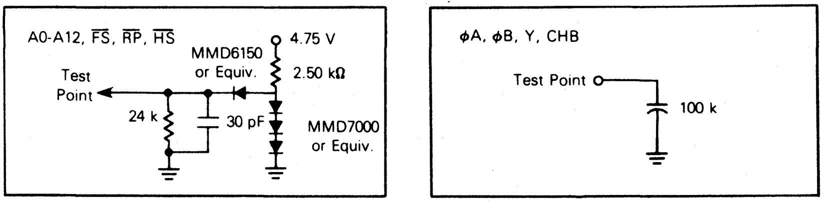
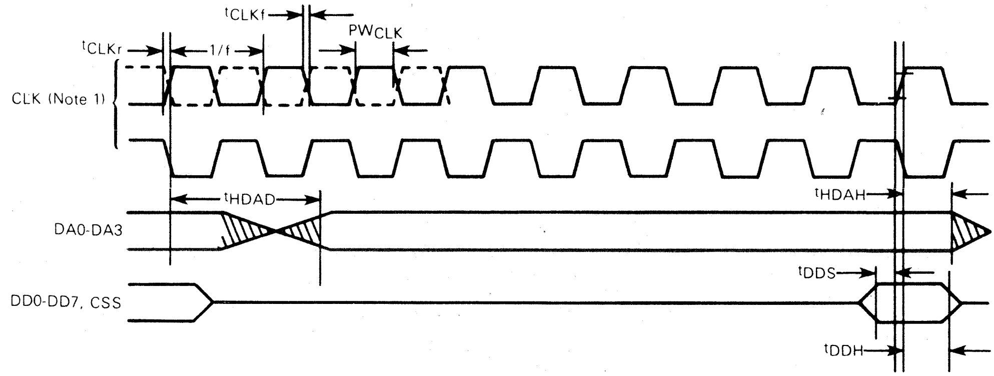


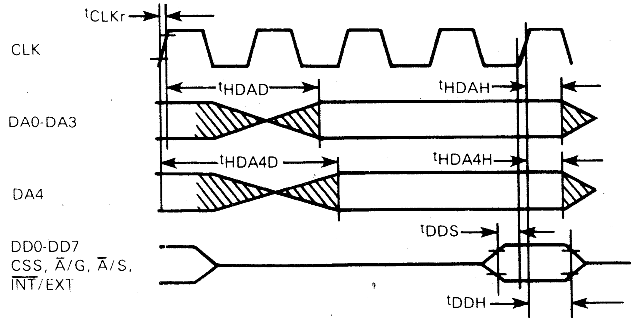
FIGURE 4 — CLOCK AND LONG CYCLE HORIZONTAL ACCESS TIMING



NOTES:

1. The VDG may power-up using either the rising or falling edge of the clock (dotted line).
2. Transitions of DA4-DA12 occur outside the display area. DA0-DA3 access the 16 bytes of data displayed during each scan line in the display area.
3. Long cycle timing applies to CG1, RG1, RG2, and RG3 modes (see Table 3). $\overline{A}$/G is high; AS, $\overline{INT}$/EXT, and INV input levels do not affect the VDG in long cycle modes.
4. Usable RAM access time for the long cycle may be calculated using the following equation:
   $t_{RACL} = 8 \cdot 1/f_{max} - t_{HDAD_{max}} - t_{DDS_{max}} - t_{CLK}$.
   If address and data buffers are used, the access time must be adjusted accordingly.
5. All timing is measured to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts unless otherwise specified.

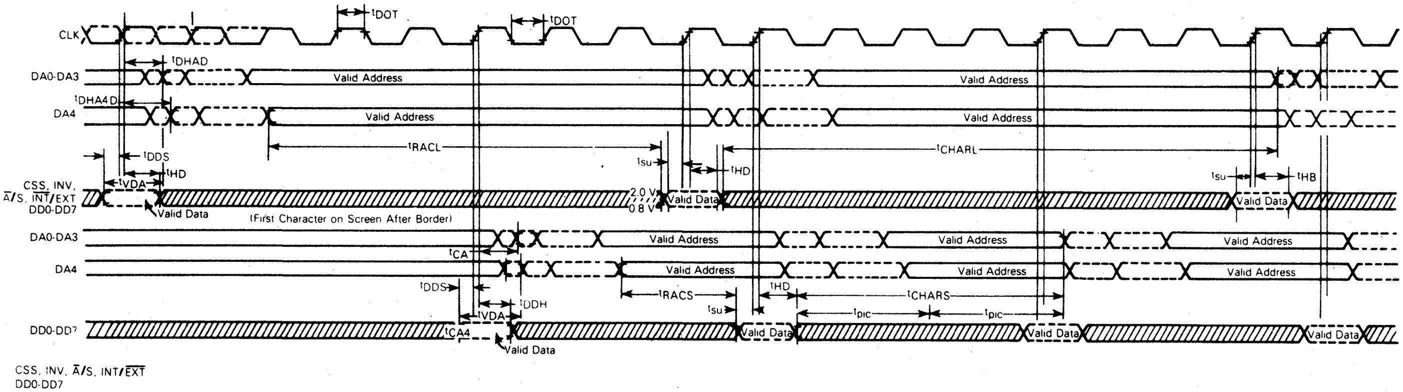
FIGURE 5 — SHORT CYCLE HORIZONTAL ACCESS TIMING



NOTES:

1. The VDG may power-up using either the rising or falling edge of the clock as shown in Figure 4.
2. Transitions of DA5-DA12 occur outside the display area. DA0-DA4 access the 32 bytes of data displayed during each scan line in the display area.
3. Short cycle timing applies to the four alphanumeric modes, two semigraphic modes, and to the CG2, CG3, CG6, RG6 modes (see Table 3). For the four graphic modes, $\overline{A}$/G is high and the $\overline{A}$/S, $\overline{INT}$/EXT, and INV input levels do not affect the VDG.
4. Usable RAM access time for the short cycle may be calculated using the following equation:
   $t_{RACS} = 4 \cdot 1/f_{max} - t_{HDA4D_{max}} - t_{DDS_{max}} - t_{CLKr}$
   If address and data buffers are used, the access time must be adjusted accordingly.
5. All timing is measured to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts unless otherwise specified.

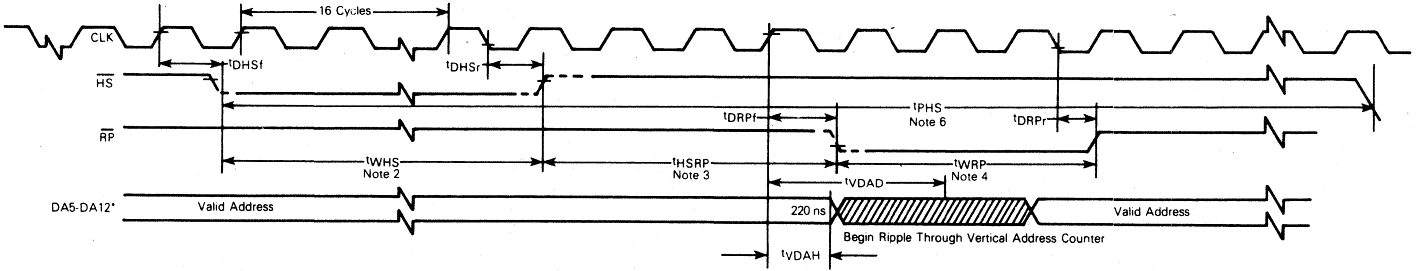
FIGURE 6 — HORIZONTAL ADDRESS AND VALID DATA SETUP AND HOLD TIMING
(Timing Relationships Shown From Beginning of Line)



*Long element/access modes: CG1, RG1, RG2, RG3

**Short element/access mode: CG2, CG3, CG6, RG6, Alphanumerics, Semigraphics

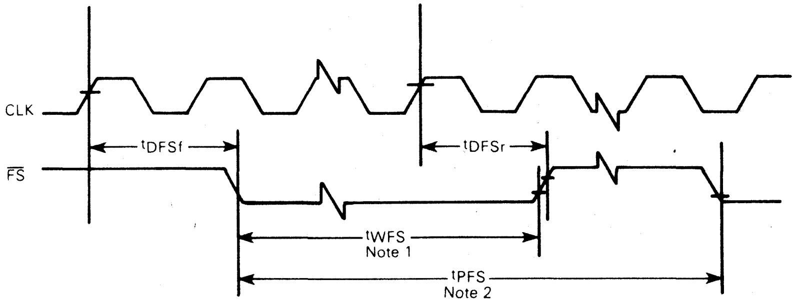
FIGURE 7 — VERTICAL ADDRESS, ROW PRESET AND HORIZONTAL SYNCHRONIZATION TIMING



NOTES:

1. All timing is measured to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts unless otherwise specified.
2. HS pulse width may be determined by $t_{WHS} = 16.5 \cdot 1/f - t_{DHSF} + t_{DHSr}$.
3. HS to RP may be determined by $t_{HSRP} = 3.5 \cdot 1/f - t_{DHSr} + t_{DRPf}$.
4. RP pulse width may be determined by $t_{WRP} = 3.5 \cdot 1/f - t_{DRPf} + t_{DRPr}$.
5. DA5-DA12 will change during the inactive portion of the display.
6. $t_{PHS} = 227.5 \cdot 1/f$.
7. $t_{DCT} = ½ f$.

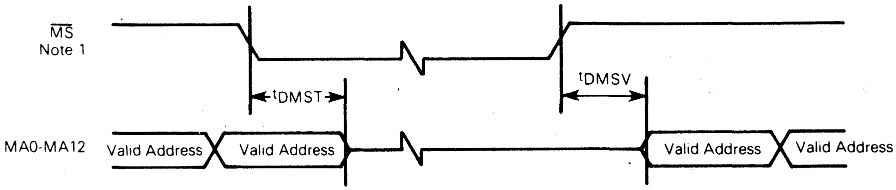
FIGURE 8 — FIELD SYNC ($\overline{FS}$) TIMING



NOTES:

1. $t_{WFS} = 32 \cdot t_{PHST} = 32 \cdot (277.5 \cdot 1/f)$
2. $t_{PFS} = 262 \cdot t_{PHST} = 262 \cdot (227.5 \cdot 1/f)$ for MC6847
   $t_{PFS} = 262.5 \cdot t_{PHST} = 262.5 \cdot (227.5 \cdot 1/f)$ for MC6847Y

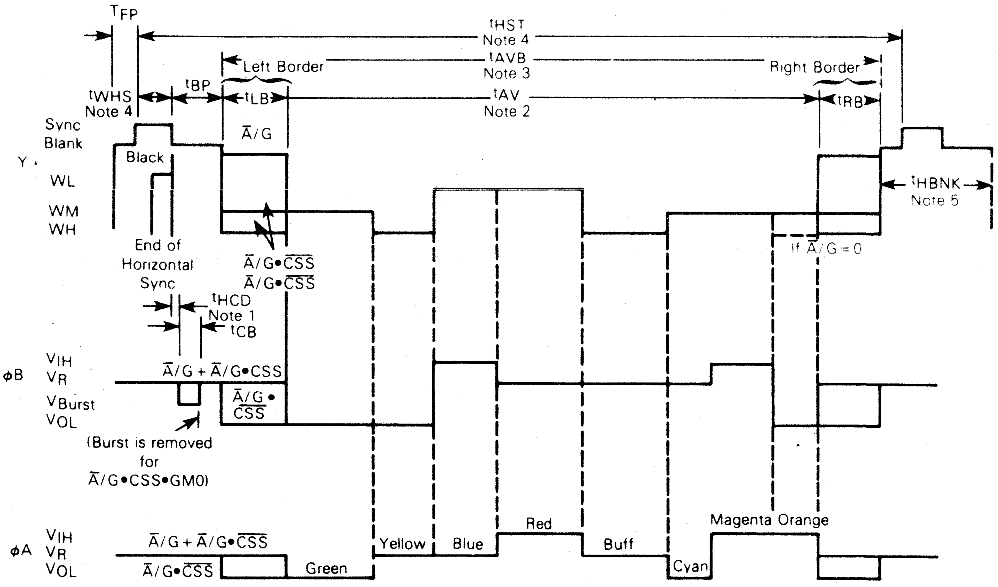
FIGURE 9 — MEMORY SELECT ($\overline{MS}$) TIMING



NOTES:

1. MS is asserted asynchronously with respect to CLK.

FIGURE 10 — VIDEO AND CHROMINANCE OUTPUT WAVEFORM RELATIONSHIPS



NOTES:

1. $t_{HCD} = 3.5 \cdot 1/f$
2. $t_{AV} = 128 \cdot 1/f$
3. $t_{AVB} = 185.5 \cdot 1/f$
4. Refer to Figure 7
5. $t_{HBNK} = 42 \cdot 1/f$

FIGURE 11 — CHROMA PHASE DELAY

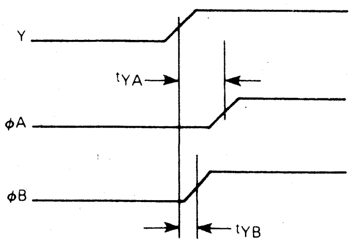
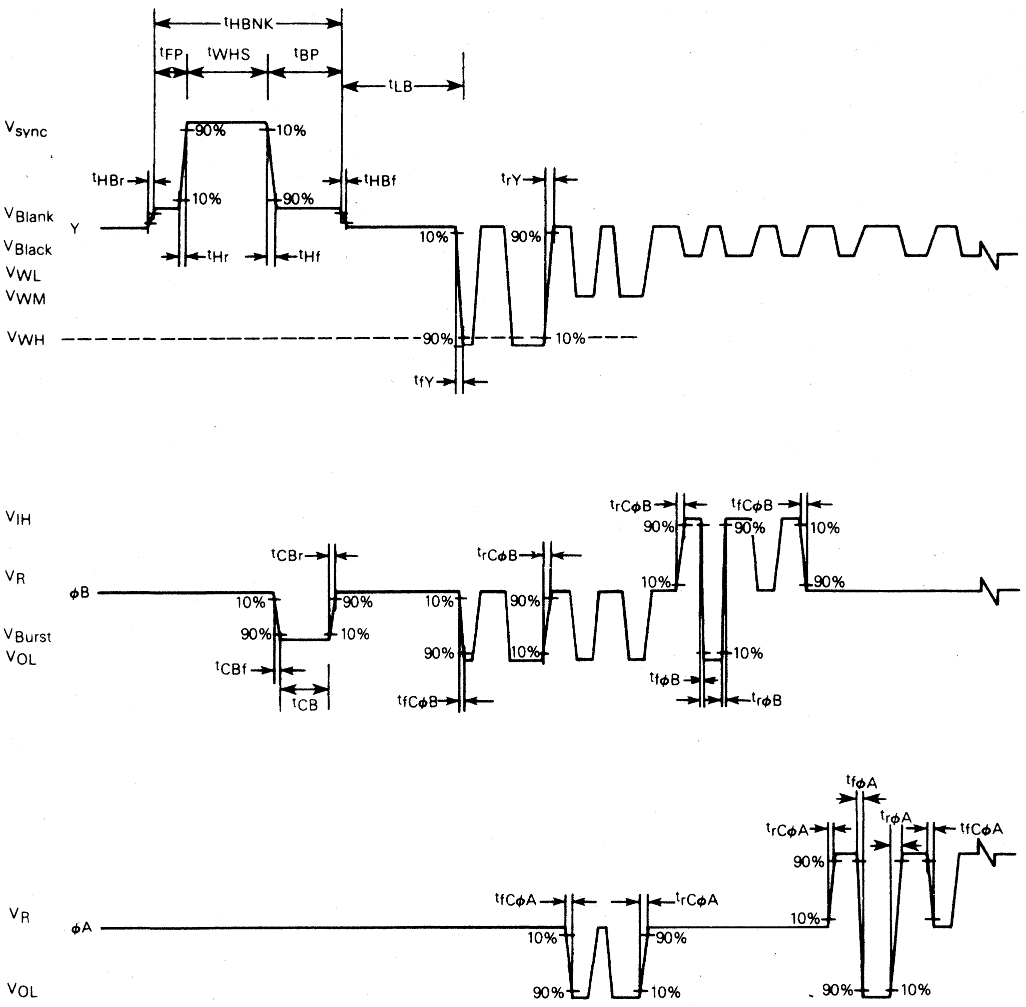


FIGURE 12 — TIMING DIAGRAMS
VIDEO RISE AND FALL TIMES (Illustrates Beginning of One Horizontal Line)



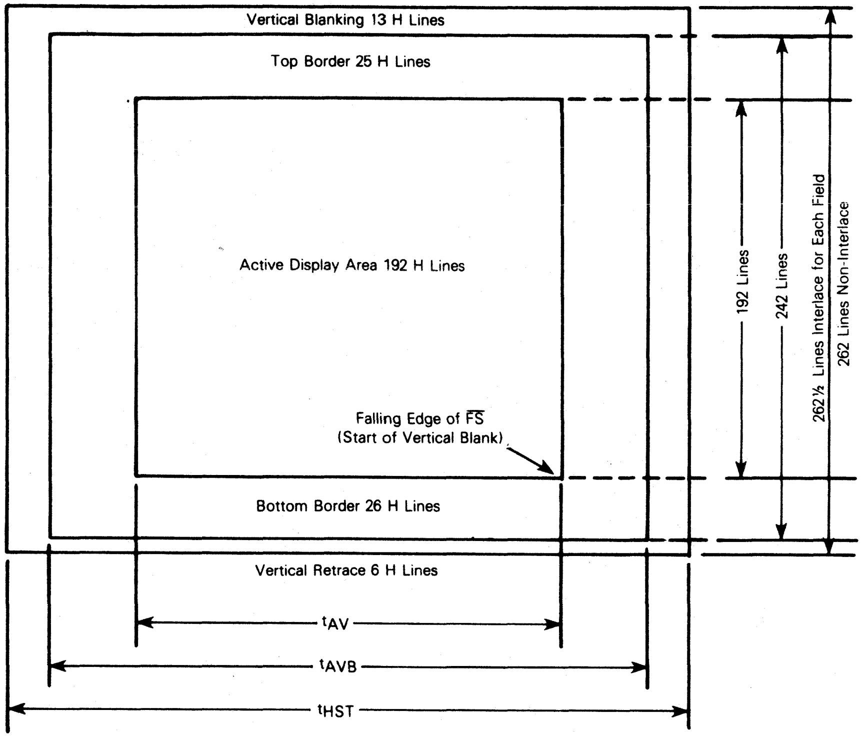


FIGURE 13 — DISPLAY AREA TIMING

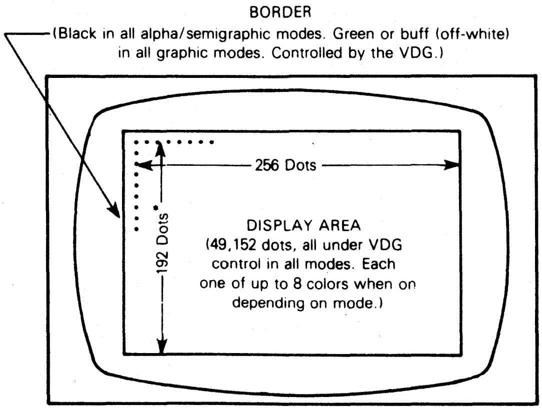


FIGURE 14 — TYPICAL FORMAT OF THE TELEVISION SCREEN

*One on each non-interlaced line; for interlace, the lines of the odd field are copied into the even field thus doubling the number of displayed dots.

VIDEO DISPLAY GENERATOR DESCRIPTION

The MC6847/MC6847Y video display generators provide a simple interface for display of digital information on a color monitor or standard color/black and white television receiver.

Television transmissions in North and South America and Japan conform to the National Television System Committee (NTSC) standards. This system is based on a field repetition rate of 60 fields per second. There are 525 interlaced lines per frame or one-half this number per field.

The MC6847 scans one field of 262 lines 60 times per second. The MC6847 non-interlace VDG is recommended for use in systems (i.e., TV games and personal computers) where absolute NTSC compatibility is not required. If NTSC compatibility is required, perhaps for caption overlays on broad-case signals, then the MC6847Y interlace VDG is recommended.

NOTE

A system with the MC6847 VDG and the MC1372 video modulator forms a transmitter, transmitting at 61.2 MHz (channel 3) or 67.25 MHz (channel 4) depending on component values chosen. This being a Class I TV device, care must be taken to meet FCC requirements Part 15, Subpart H. However, if the composite video output from the MC1372 were to drive the television directly, Section 15.7 of the FCC specification must be adhered to.

SIGNAL DESCRIPTION

DISPLAY ADDRESS OUTPUT LINES (DA0-DA12)

Thirteen address lines are used by the VDG to scan the display memory as shown in Figures 4-7. The starting address of the display memory is located at the upper left corner of the display screen. As the television sweeps from the left to right and top to bottom, the VDG increments the RAM display address. The timing for two accesses starting at the beginning of the line is shown in Figure 6. These lines are TTL compatible and may be forced into a high-impedance state whenever MS (pin 12) goes low. A0-A3 change during the active display area. A4 changes during the active display area in the alphanumerics, semigraphics, CG2, CG3, CG6, and RG6 modes. A5-A12 do not toggle within the active display area but instead, ripple through the address during border and blanking time.

DATA INPUTS (DD0-DD7)

Eight TTL compatible data lines are used to input data from RAM to be processed by the VDG. The data is then interpreted and transformed into luminance (Y) and chroma outputs (ϕA and ϕB).

POWER INPUTS — V_{CC} requires +5 volts ±5%. V_{SS} requires zero volts and is normally ground. The tolerance and current requirements of the VDG are specified in the Electrical Characteristics.

VIDEO OUTPUTS (ϕA, ϕB, Y, CHB) — These four analog outputs are used to transfer luminance and color information to a standard NTSC color television receiver, either via the MC1372 RF modulator or via drivers directly into Y, ϕA, ϕB television video inputs (see Figures 10, 11, and 12).

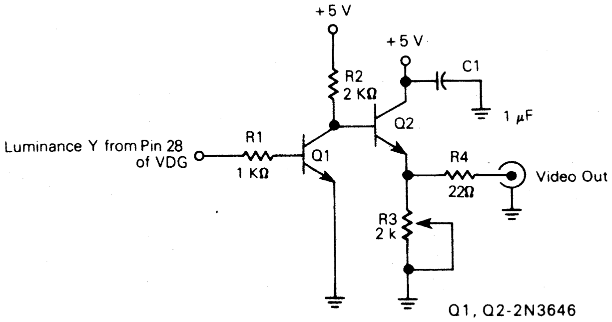
Luminance (Y) — This six level analog output contains composite sync, blanking and four levels of video luminance.

ϕA — This three level analog output is used in combination with ϕB and Y outputs to specify one of eight colors.

ϕB — This four level output is used in combination with ϕA and Y outputs to specify one of eight colors. Additionally, one analog level is used to specify the time of the color burst reference signal.

Chroma Bias (CHB) — This pin is an analog output and provides a DC reference corresponding to the quiescent value of ϕA and ϕB. CHB is used to guarantee good thermal tracking and minimize the variation between the MC1372 and MC6847. This pin, when pulled low, resets certain registers within the chip. In a user's system, this pin should not normally be used as an input. It is used mainly to enhance test capabilities within the factory.

FIGURE 15 — VIDEO TO B&W MONITOR



Contributed by Ted Hanwick, Honeywell, Inc.

SYNCHRONIZING INPUTS ($\overline{MS}$, CLK)

THREE-STATE CONTROL — ($\overline{MS}$) is a TTL compatible input which, when low, forces the VDG address lines into a high-impedance state, as shown in Figure 9. This may be done to allow other devices (such as an MPU) to address the display memory (RAM).

CLOCK (CLK) — The VDG clock input (CLK) requires a 3.579545 MHz (standard color burst) TV crystal frequency square wave. The duty cycle of this clock must be between 45 and 55% since it controls the width of alternate dots on the television screen. The MC1372 RF modulator may be used to supply the 3.579545 MHz clock and has provisions for a duty cycle adjustment. The VDG will power-up using either the rising or falling edge of the clock. The dotted line on the CLK signal in Figure 4 indicates this characteristic of latching in data on either clock edge.

SYNCHRONIZING OUTPUTS ($\overline{FS}$, $\overline{HS}$, $\overline{RP}$)

Three TTL compatible outputs provide circuits, exterior to the VDG, with timing references to the following internal VDG states:

FIELD SYNC ($\overline{FS}$) — The high-to-low transition of the $\overline{FS}$ output coincides with the end of active display area (see Figure 8). During this time interval, an MPU may have total access to the display RAM without causing undesired flicker on the screen. The low-to-high transition of $\overline{FS}$ coincides with the trailing edge of the vertical synchronization pulse.

HORIZONTAL SYNC ($\overline{HS}$) — The $\overline{HS}$ pulse coincides with the horizontal synchronization pulse furnished to the television receiver by the VDG (see Figure 7). The high-to-low transition of the $\overline{HS}$ output coincides with the leading edge of the horizontal synchronization pulse and the low-to-high transition coincides with the trailing edge.

ROW PRESET ($\overline{RP}$) — If desired, an external character generator ROM may be used with the VDG. However, an external four bit counter must be added to supply row addresses. The counter is clocked by the $\overline{HS}$ signal and is cleared by the $\overline{RP}$ signal. $\overline{RP}$ pulses occur in all alphanumeric and semigraphics modes; no pulses are output in the full graphic modes. $\overline{RP}$ occurs after the first valid 12 lines. Therefore, use an $\overline{FS}$ clocked preloadable counter such as a 74LS161 as shown in Figures 7, 14, and 23.

MODE CONTROL LINES INPUT ($\overline{A}$/G, $\overline{A}$/S, $\overline{INT}$/EXT, GM0, GM1, GM2, CSS, INV)

Eight TTL compatible inputs are used to control the operating mode of the VDG. $\overline{A}$/S $\overline{INT}$/EXT, CSS, and INV may be changed on a character-by-character basis. The CSS pin is used to select between two possible alphanumeric colors when the VDG is in the alphanumeric mode and between two color sets when the VDG is in the Semigraphics 6 or full graphic modes. Table 1 illustrates the various modes that can be obtained using the mode control lines. There are two different types of memory access concerning these modes, they are a short and a long access cycle, which differ by a

FIGURE 16 — EXTERNAL CHARACTER GENERATOR ROW COUNTER FOR MC6847

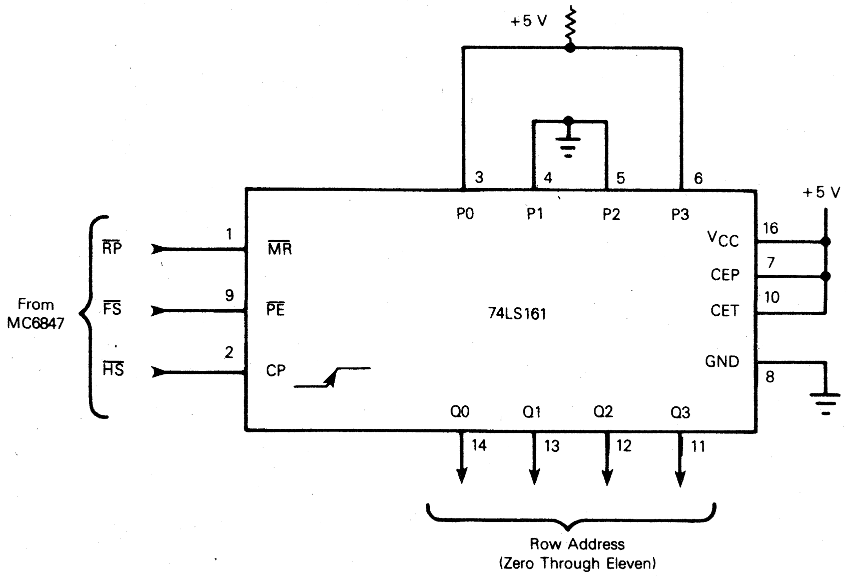


TABLE 1 — MODE CONTROL LINES (INPUTS)

$\overline{\text{A}}$/G	$\overline{\text{A}}$/S	$\overline{\text{INT}}$/EXT	INV	GM2	GM1	GM0	Alpha/Graphic Mode Select	# of Colors
0	0	0	0	X	X	X	Internal Alphanumerics	2
0	0	0	1	X	X	X	Internal Alphanumerics Inverted	
0	0	1	0	X	X	X	External Alphanumerics	
0	0	1	1	X	X	X	External Alphanumerics Inverted	
0	1	0	X	X	X	X	Semigraphics 4 (SG4)	8
0	1	1	X	X	X	X	Semigraphics 6 (SG6)	8
1	X	X	X	0	0	0	64 × 64 Color Graphics One (CG1)	4
1	X	X	X	0	0	1	128 × 64 Resolution Graphics One (RG1)	2
1	X	X	X	0	1	0	128 × 64 Color Graphics Two (CG2)	4
1	X	X	X	0	1	1	128 × 96 Resolution Graphics Two (RG2)	2
1	X	X	X	1	0	0	128 × 96 Color Graphics Three (CG3)	4
1	X	X	X	1	0	1	128 × 192 Resolution Graphics Three (RG3)	2
1	X	X	X	1	1	0	128 × 192 Color Graphics Six (CG6)	4
1	X	X	X	1	1	1	256 × 192 Resolution Graphics Six (RG6)	2

shift of one full 3.58 MHz cycle. One of the differences between these access times, in the short access time frame, is a shift of one full 3.58 MHz cycle from the corresponding normal long access time frame, as shown in Figure 6. The modes using short access times read memory twice as often as the long access modes.

OPERATION OF THE VDG

A simplified block diagram of the VDG is shown in Figure 17a and a detailed block diagram is shown in Figure 17b.

The externally generated 3.58 MHz color burst clock drives the VDG. Referring to Figures 11 and 12, note that the horizontal screen span from blanking to blanking is 193.1 clock periods (≈ 53.95 μs). The display window is offset from the left-hand edge by 283 periods and lasts for 128 periods (35.75 μs). Of the 242 lines on the vertical screen from blanking to blanking, 192 lines are used for the display. The display window is offset from the top by 25 lines. Under the constraint of the master clock, the smallest display element possible for the VDG is half period of the 3.58 MHz clock wide by one scan line high. All other display elements are multiples of this basic size.

DISPLAY MEMORY ADDRESS DRIVERS

The address drivers normally drive the video refresh address into the display memory so characters may be displayed on the CRT. When the memory select pin (MS) is pulled low by an external decoder, the driver outputs go to a high-impedance state so external three-state drivers may switch the MPU produced address onto the display memory address bus; the MPU may directly manipulate data in the display memory.

VIDEO TIMING AND CONTROL

This subsystem of the VDG includes the mode decoding, timing generation, and associated row counter logic, and uses the 3.58 MHz color frequency to generate horizontal and vertical timing information (via linear shift register counters), which the video and chroma encoder uses to generate color video information. The horizontal timing for the VDG is summarized in Figure 7. Ten and one-half cycles of the 3.58 MHz subcarrier are transmitted on the back porch of every horizontal blanking period. This color burst is suppressed during vertical sync and equalizing intervals. Color burst is also suppressed in the most dense two color graphic modes. This leads to some interesting rainbow effects on the display which is frequency and pattern dependent. The vertical timing for the VDG is given in Figure 18. Vertical retrace is initiated by the luminance signal being brought to the blanking level. The vertical blanking period begins with three lines of equalizing pulses followed by three lines of serrated vertical sync pulses followed by three more lines of equalizing pulses. The remaining vertical blanking period contains the normal horizontal sync pulses. The equalizing and serration pulses are at half line frequency. Notice the difference in spacing between the last horizontal sync pulse and the first equalizing pulse in even and odd fields. It is the half line difference between fields that produces the interlaced picture in a frame. Vertical timing between fields for the non-interlaced VDG, on the other hand, is identical. The equalizing and serration pulses are, however, at the horizontal frequency.

The 3.58 MHz color frequency is also used to clock the video shift register load counter. This counter and the video shift clock inhibit circuitry derive the dot-clock for the output of the video shift registers and the load signals for the video shift registers' input latches. The vertical and horizontal address counters generate the addresses for the external display memory.

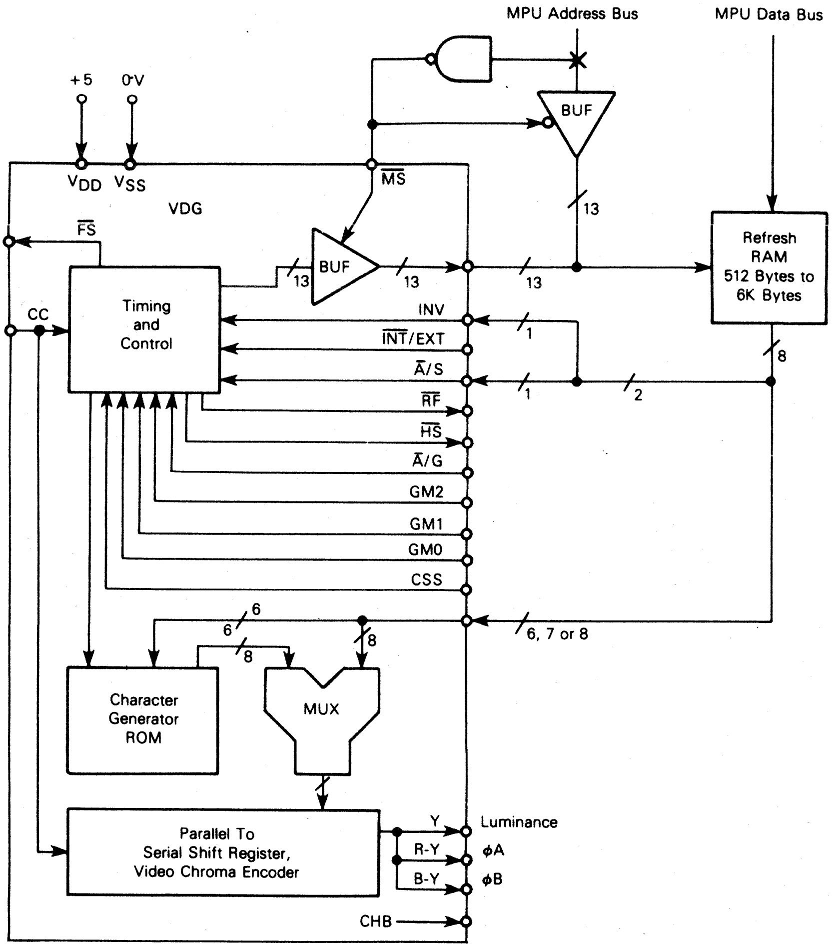
INTERNAL CHARACTER GENERATOR ROM

Since many uses of the VDG will involve the display of alphanumeric data, a character-generator ROM is included on the chip. This ROM will generate 64 standard 5 × 7 dot matrix characters from standard 6-bit ASCII input. A standard character set is included in the MC6847 although the ROM is custom programmable.

INTERNAL/EXTERNAL CHARACTER GENERATOR MULTIPLEXER

The internal/external multiplexer allows the use of either the internal ROM or an external character generator. This multiplexer may be switched on a character-by-character basis to allow mixed internal and external characters on the CRT. The external character may be any desired dot-pattern in the standard 8 × 12 one-character display matrix, thus allowing the maximum 256 × 192 screen density.

FIGURE 17a — SIMPLIFIED VDG BLOCK DIAGRAM



VIDEO AND COLOR SUBSYSTEM

The 8-bit output of the internal/external multiplexer is serialized in an 8-bit shift register clocked at the dot-clock frequency.

The luminance information from the shift register is summed with the horizontal and vertical sync signals to produce a composite video signal less the chrominance information, called Y. The luminance signal, Y, and the two chrominance outputs, ϕA (R-Y) and ϕB (B-Y), can be combined (modulated) by an MC1372 into a composite video signal with color. Figures 8, 9, 10, and 16 show the relationship between the luminance and chrominance signals and the resultant color.

FIGURE 17b — DETAILED VDG BLOCK DIAGRAM

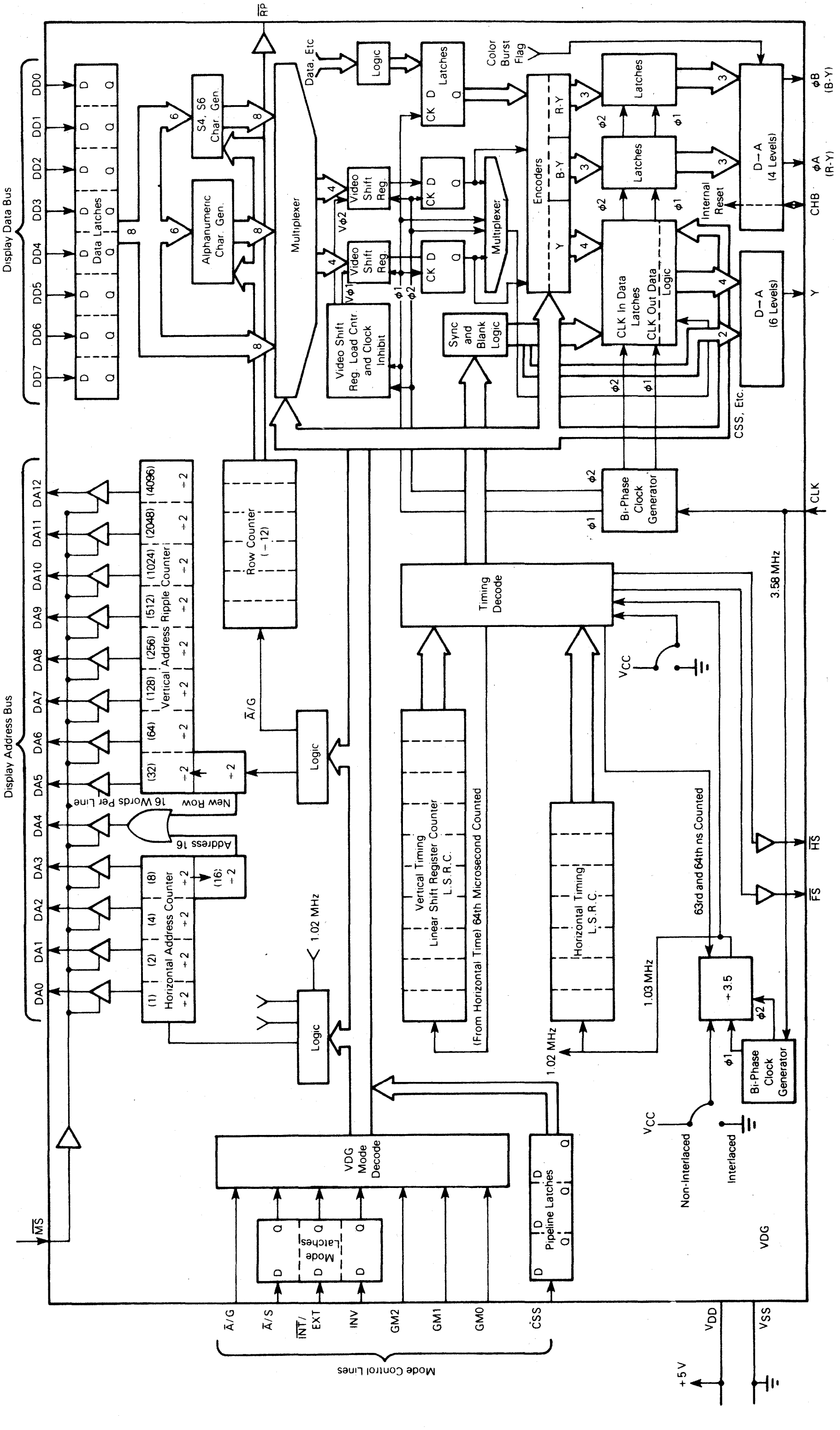
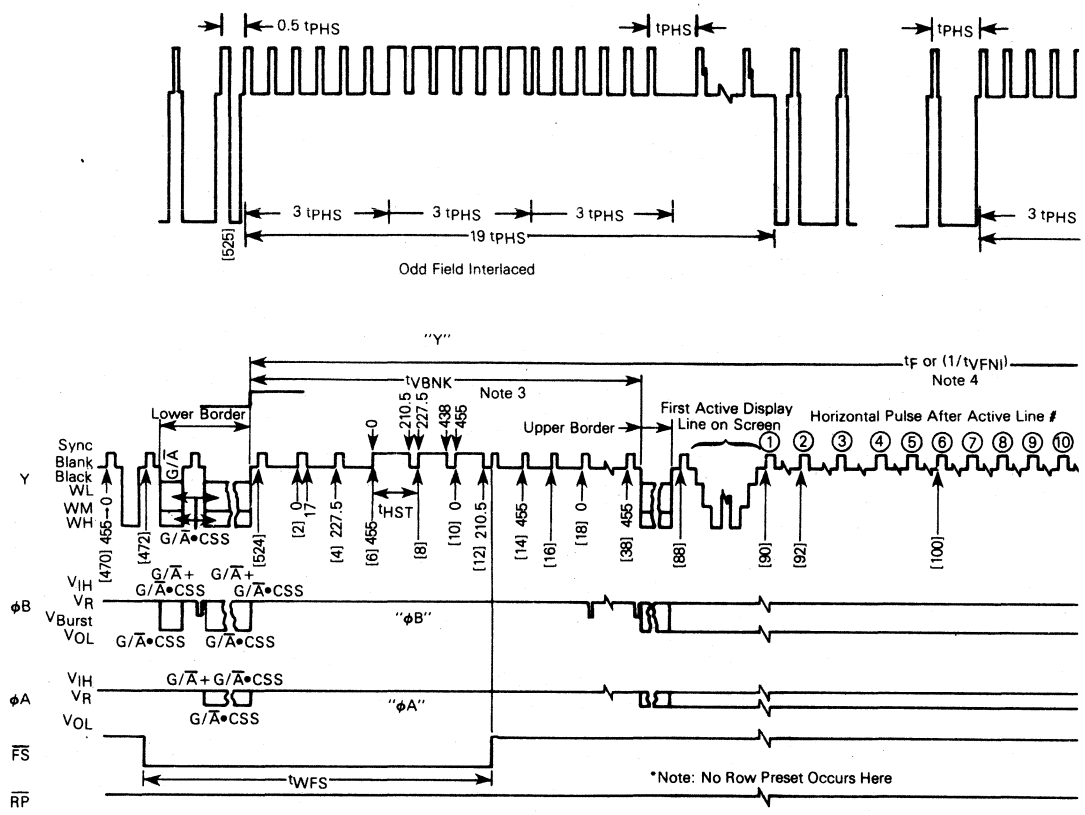


FIGURE 18 — NON-INTERLACE VERTICAL TIMING
(For Interlace Vertical Timing Use Inserts)



NOTES:

1. [ ] # of 31.468 kHz clocks
   (1/31.468 kHz) = 31.7783 μs
   Non-Parentheses = # of 3.58 MHz Clocks
   1/3.58 MHz = 279.366 ns
   Time marks 455 and 0 are the same points in time.

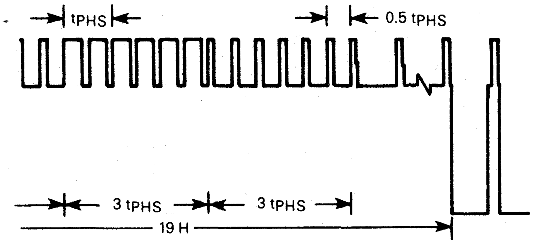
   Example Timing Calculations:
   t_{HST} = (227.5-0) × 279.366 ns = 63.5μs
   = (455-227.5) × 279.366 ns = 63.5 μs
   t_{WHS} = (455-438) × 279.366 ns = 4.75 μs
   Lower Border = (524-472) × 31.7783 μs-t_{HBNK}
   = 1.6525 ms-11.6 ns = 1.64 ms

   Upper Border = (88 − 38) × 31.7783 μs − t_{HBNK} =
   = 1.5889 μs-11.6 μs = 1.58 ms

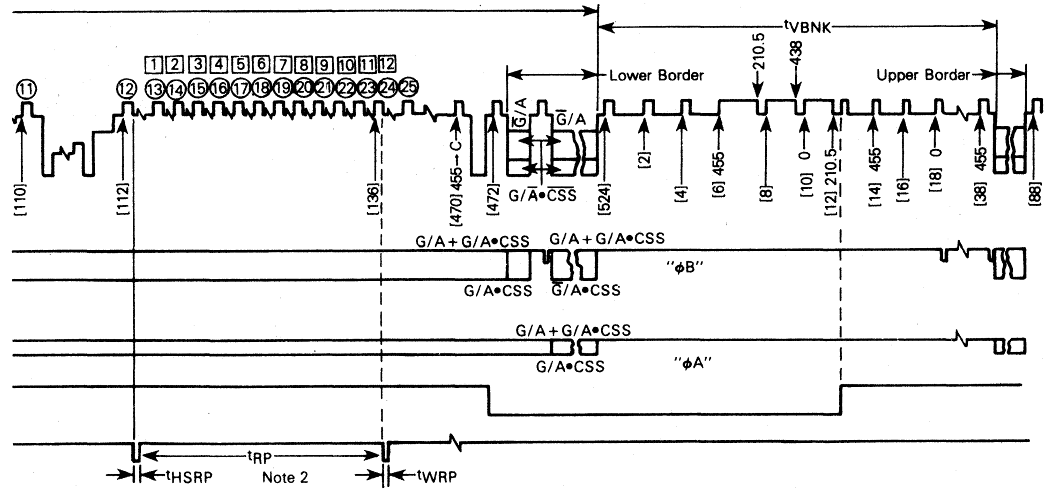
2. t_{RP} = 12 horizontal scan lines.
3. t_{VBNK} = 20•t_{PHS} = 20•(227.5•1/f)
4. t_F = 262•t_{PHS} = 262•(227.5•1/f) for Non-Interlace.
   t_F = 262.5•T_{PHS} = 262.5•(227.5•1/f) for Interlace.

*t_{HBNK} could actually be considered as part of the border — especially for purposes of writing to the screen. The same holds true for the upper border.

FIGURE 18 — NON-INTERLACE VERTICAL TIMING
(For Interlace Vertical Timing Use Inserts)
(Continued)



Even Field Interlaced



First Row Preset to Occur
Comes After the First Active
Row of Characters.

DISPLAY MODES

There are two major display modes in the VDG. Major mode 1 contains four alphanumeric and two limited graphic modes. Major mode 2 contains eight graphic modes. Of these, four are full color graphic and four restricted color graphic modes. The mode selection for the VDG is summarized in Table 2. The mnemonics of these fourteen modes are explained in the following sections.

In major mode 1 the display window is divided into 32 columns by 16 character element rows thus requiring 512 bytes of memory. Each character element is 8 half periods by 12 scan lines in size as shown in Figure 19. The area outside the display window is black.

The VDG has a built-in character generator ROM containing the 64 ASCII characters in a 5 × 7 format (see Figure 20). The 5 × 7 character font is positioned two columns to the right and three rows down within the 8 × 12 character element. Six bits of the 8-bit data word are typically used for the internal ASCII character generator. The remaining two bits may be used to implement inverse video, color switching, or external character generator ROM selection on a character-by-character basis. For those who wish to display lower case letters, special characters, or even limited-graphics, an external ROM may be used. If such external ROM is used, all of the 8 × 12 picture elements, or pixels, in the character element can be utilized. Characters may be either green on a dark green background or orange on a dark orange background, depending on the state of the CSS pin. The invert pin can be used to display dark characters on a bright background.

TABLE 2 — SUMMARY OF MODES
Major Mode 1 — Alpha Modes

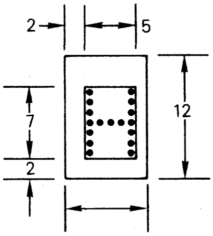
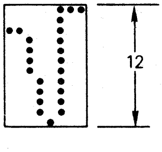
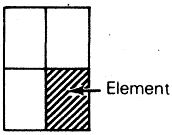
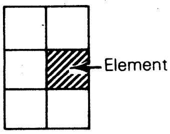
Title	Memory	Display Elements	Colors
Alphanumerics (Internal)	512 × 8		2
Alphanumerics (External)	512 × 8		2

TABLE 2 — SUMMARY OF MODES
Major Mode 1 — Alpha Modes
(Continued)

Title	Memory	Display Elements	Colors
Semigraphic 4	512 × 8		8
Semigraphic 6	512 × 8		4

Major Mode 2 — Graphics Modes

Title	Memory	Colors	Comments
64 × 64 Color Graphic	1 k × 8	4	Matrix 64 × 64 Elements
128 × 64 Graphics*	1 k × 8	2	Matrix 128 Elements Wide by
128 × 64 Color Graphic	2 k × 8	4	64 Elements High
128 × 96 Graphics*	1.5 k × 8	2	Matrix 128 Elements Wide by
128 × 96 Color Graphic	3 k × 8	4	96 Elements High
128 × 192 Graphics*	3 k × 8	2	Matrix 128 Elements Wide by
128 × 192 Color Graphic	6 k × 8	4	192 Elements High
256 × 192 Graphics	6 k × 8	2	Matrix 256 Elements Wide by 192 Elements High

*Graphics mode turns on or off each element. The color may be one of two.

TABLE 3 — DETAILED DESCRIPTION OF VDG MODES

VDG Pins									Color			
$\overline{\text{MS}}$	G/$\overline{\text{A}}$	S/$\overline{\text{A}}$	EXT/$\overline{\text{INT}}$	GM2	GM1	GM0	CSS	INV	Character Color	Background	Border	Display Mode
1	0	0	0	X	X	X	0 1	0 1 0 1	Green Black Orange Black	Black Green Black Orange	Black Black	32 Characters per row 16 Characters in rows
1	0	0	1	X	X	X	0 1	0 1 0 1	Green Black Orange Black	Black Green Black Orange	Black Black	32 Characters per row 16 Characters in rows
1	0	1	0	X	X	X	X	X	Lx C2 C1 C0 0 X X X 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1	Color Black Green Yellow Blue Red Buff Cyan Magenta Orange	Black	64 Display elements per row 32 Display elements in rows in rows
1	0	1	1	X	X	X	0 1	X	Lx C1 C0 0 X X 1 0 0 1 0 1 1 1 0 1 1 1 0 X X 1 0 0 1 0 1 1 1 0 1 1 1	Color Black Green Yellow Blue Red Black Buff Cyan Magenta Orange	Black	64 Display elements per row 48 Display elements in rows
1	1	X	X	0	0	0	0 1	X	C1 C0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1	Color Green Yellow Blue Red Buff Cyan Magenta Orange	Green Buff	64 Display elements per row 64 Display elements
1	1	X	X	0	0	1	0 1	X	Lx 0 1 0 1	Color Black Green Black Buff	Green Buff	128 Display elements per row 64 Display elements in rows
1	1	X	X	0	1	0	0 1	X	Same color as Color Graphics One		Green Buff	128 Display elements per row 64 Display elements in rows
1	1	X	X	0	1	1	0 1	X	Same color as Resolution Graphics One		Green Buff	128 Display elements per row 96 Display elements in rows
1	1	X	X	1	0	0	0 1	X	Same color as Color Graphics One		Green Buff	128 Display elements per row 96 Display elements in rows
1	1	X	X	1	0	1	0 1	X	Same color as Resolution Graphics One		Green Buff	128 Display elements per row 192 Display elements in rows
1	1	X	X	1	1	0	0 1	X	Same color as Color Graphics One		Green Buff	128 Display elements per row 192 Display Elements in rows
1	1	X	X	1	1	1	0 1	X	Same color as Resolution Graphics One		Green Buff	256 Display elements per row 192 Display elements in rows

TABLE 3 — DETAILED DESCRIPTION OF VDG MODES
(Continued)

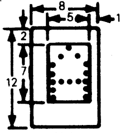
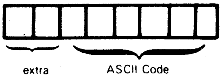
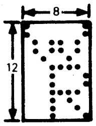
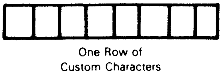
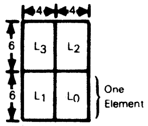
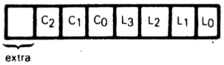
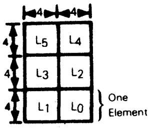
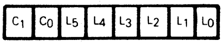
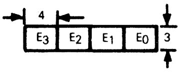
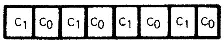
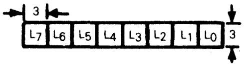
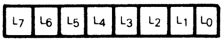
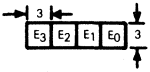
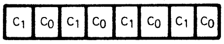
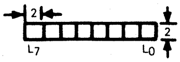
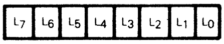
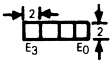
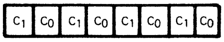
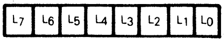
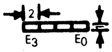
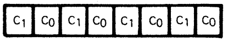
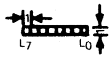
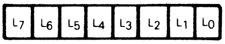
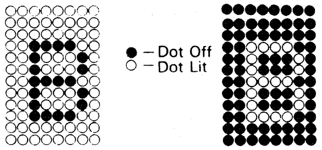
TV Screen	VDG Data Bus	Comments
Detail		
 Internal Alphanumerics		The ALPHANUMERIC INTERNAL mode uses an internal character generator (which contains the following five dot by seven dot characters: ,@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]∫∫→SP !"#$%&'( )* + , - ,0123456789:;< = >?. The six bit ASCII code leaves two bits free and these may be externally connected to the mode pins (G/$\overline{A}$, S/$\overline{A}$, EXT/$\overline{INT}$, GM2, GM1, GM0, CSS or INV).
		The ALPHANUMERIC EXTERNAL mode uses an external character generator as well as a row counter. Thus, custom character fonts or graphic symbol sets with up to 256 different 8 × 12 dot "characters" may be displayed.
		The SEMIGRAPHICS FOUR mode uses an internal "course graphics" generator in which a rectangle (eight dots by twelve dots) is divided into four equal parts. The luminance of each part is determined by a corresponding bit on the VDG data bus. The color of illuminated parts is determined by three bits.
		The SEMIGRAPHIC SIX mode is similar to the SEMIGRAPHIC FOUR mode with the following differences. The eight dot by twelve dot rectangle is divided into six equal parts. Color is determined by the two remaining bits.
		The COLOR GRAPHICS ONE mode uses a maximum of 1024 bytes of display RAM in which one pair of bits specifies one picture element.
		The RESOLUTION GRAPHICS ONE mode uses a maximum of 1024 bytes of display RAM in which one bit specifies one picture element.
		The COLOR GRAPHICS TWO mode uses a maximum of 2048 bytes of display RAM in which one pair of bits specifies one picture element.
		The RESOLUTION GRAPHICS TWO mode uses a maximum of 1536 bytes of display RAM in which one bit specifies one picture element.
		The COLOR GRAPHICS THREE mode uses a maximum of 3072 bytes of display RAM in which one pair of bytes specifies one picture element.
		The RESOLUTION GRAPHICS THREE mode uses a maximum of 3072 bytes of display RAM in which one bit specifies one picture element.
		The COLOR GRAPHICS SIX mode uses a maximum of 6144 bytes of display RAM in which one pair of bits specifies one picture element.
		The RESOLUTION GRAPHICS SIX mode uses a maximum of 6144 bytes of display RAM in which one bit specifies one picture element.

FIGURE 19 — ALPHANUMERIC MODE (INTERNAL)

512 Characters (32 × 16)
Typical Character



Inverted	Normal
Black Character	Black Background
Orange or Green	Orange or Green
Background (Selectable)	Character (Selectable)

Character Source:
Internal — 6 Bit ASCII Generator ROM On Chip or User Definable
External — Users ROM

The two limited graphic modes are Semigraphics 4 and Semigraphics 6. In Semigraphics 4, the 8 × 12 dot character block is divided into four pixels (each pixel is four half-clocks by six scan lines). The four low-order bits (DD0-DD3) of each incoming byte of data select one of sixteen possible illumination patterns while the next three bits (DD4-DD6) determine the color of the illuminated elements. The most significant bit is unused. Figure 21 shows the color and pattern selections. In Semigraphics 6 the 8 × 12 dot character block is divided into six pixels, each four half-clocks by four scan lines. The six low-order bits of each byte of incoming data select one of 64 possible illumination patterns while the CSS input and the high-order data bits (DD6-DD7) determine the color of the illuminated elements.

The display window in major mode 2 (full graphics) has a less rigorous format than in major mode 1. The display elements vary from one scan line to three scan lines in height. The length of the display element is either eight or sixteen half-periods wide. Each display element is divided into four or eight pixels. The former corresponds to a full color mode while the latter a restricted color mode, like the semigraphics modes, represents illumination data. When it is high the pixel is illuminated with the color chosen by the color set select (CSS) pin. When it is low the pixel is black. In the full color modes, pairs of data bits choose one of four colors in one of two color sets defined by the CSS pin. Depending on the state of the CSS pin, the area outside the display window is either green or buff. The display formats and color selection for this major mode are summarized in Figure 19.

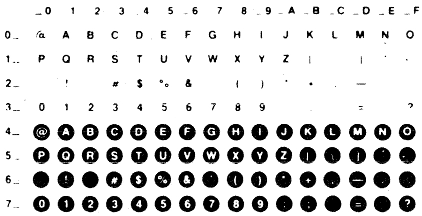
THE 64 × 64 COLOR GRAPHICS ONE (CG1) MODE — The 64 × 64 color graphics mode generates a display matrix of 64 elements wide by 64 elements high. Each element may be one of four colors. A 1k × 8 display memory is required. The display RAM is accessed 16 times per horizontal line. Each pixel equals four half-clocks by three scan lines.

THE 128 × 64 RESOLUTION GRAPHICS ONE (RG1) MODE — The 128 × 64 graphics mode generates a matrix 128 elements wide by 64 elements high. Each element may be either ON or OFF. However, the entire display may be one of two colors, selected by using the color set select pin. A 1k × 8 display memory is required. The display RAM is accessed 16 times per horizontal line. Each pixel equals two half-clocks by three scan lines.

FIGURE 20 — AVAILABLE ALPHANUMERICS

MD4 = INV = D4
MD7 = AS = D7

○ = Inverted Character — Illuminated Background, Dark Character



THE 128 × 64 COLOR GRAPHICS TWO (CG2) MODE — The 128 × 64 color graphics mode generates a display matrix 128 elements wide by 64 elements high. Each element may be one of four colors. A 2k × 8 display memory is required. The display RAM is accessed 32 times per horizontal line. Each pixel equals two half-clocks by three scan lines.

THE 128 × 96 RESOLUTION GRAPHICS TWO (RG2) MODE — The 128 × 96 graphics mode generates a display matrix 128 elements wide by 96 elements high. Each element may be either ON or OFF. However, the entire display may be one of two colors selected by using the color set select pin. A 1.5k × 8 display memory is required. The display RAM is accessed 16 times per horizontal line. Each pixel equals two half-clocks by two scan lines.

THE 128 × 96 COLOR GRAPHICS THREE (CG3) MODE — The 128 × 96 color graphics mode generates a display 128 elements wide by 96 elements high. Each element may be one of four colors. A 3k × 8 display memory is required. The display RAM is accessed 32 times per horizontal line. Each pixel equals two half-clocks by two scan lines.

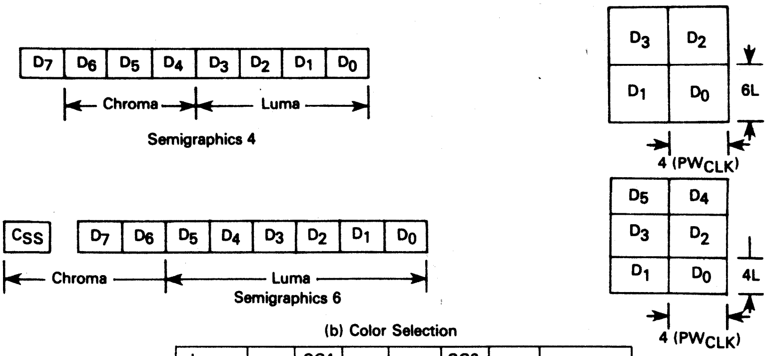
THE 128 × 192 RESOLUTION GRAPHICS THREE (RG3) MODE — The 128 × 192 graphics mode generates a display matrix 128 elements wide by 192 elements high. Each element may be either ON or OFF, but the ON element may be one of two colors selected with the color set select pin. A 3k × 8 display memory is required. The display RAM is accessed 16 times per horizontal line. Each pixel equals two half-clocks by one scan line.

THE 128 × 192 COLOR GRAPHICS SIX (CG6) MODE — The 128 × 192 color graphics mode generates a display 128 elements wide by 192 elements high. Each element may be one of four colors. A 6k × 8 display memory is required. The display RAM is accessed 32 times per horizontal line. Each pixel equals two half-clocks by one scan line.

THE 256 × 192 RESOLUTION GRAPHICS SIX (RG6) MODE — The 256 × 193 graphics mode generates a display 256 elements wide by 192 elements high. Each element may be either ON or OFF, but the ON element may be one of two colors selected with the color set select pin. A 6k × 8 display memory is required. The display RAM is accessed 32 times per horizontal line. Each pixel equals one half-clock by one scan line.

FIGURE 21 — SEMIGRAPHIC MODE ENCODING

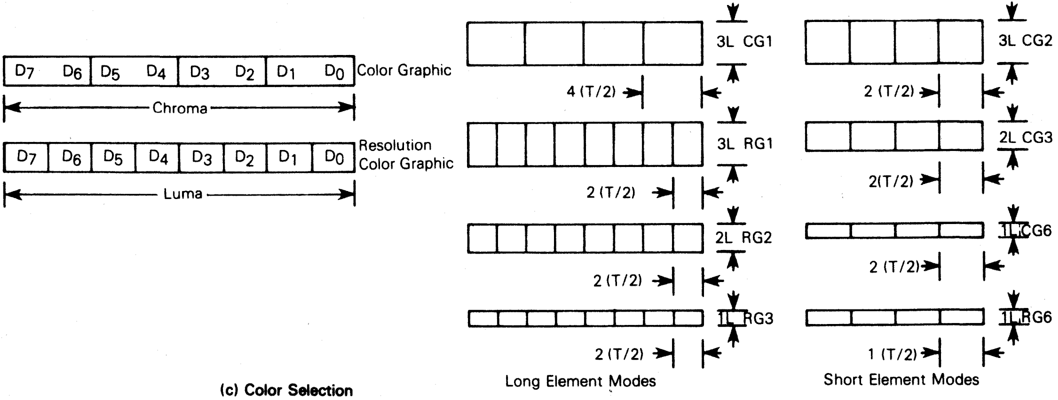
(a) Data and Display Formats



(b) Color Selection

Luma DN		SG4			SG6		Color
	D_6	D_5	D_4	CSS	D_7	D_6	
0	X	X	X	X	X	X	Black
1	0	0	0	0	0	0	Green
1	0	0	1	0	0	1	Yellow
1	0	1	0	0	1	0	Blue
1	0	1	1	0	1	1	Red
1	1	0	0	1	0	0	Buff
1	1	0	1	1	0	1	Cyan
1	1	1	0	1	1	0	Magenta
1	1	1	1	1	1	1	Orange

FIGURE 22 — GRAPHIC MODE ENCODING


(a) Data Format

(b) Display Format

(c) Color Selection

CSS	Border	Resolution Color Mode		Color Mode		
		D_N	Color	D_{N+1}	D_N	Color
0	Green	0	Black	0	0	Green
0	Green	1	Green	0	1	Yellow
0	Green	1	Green	1	0	Blue
0	Green	1	Green	1	1	Red
1	Buff	0	Black	0	0	Buff
1	Buff	1	Buff	0	1	Cyan
1	Buff	1	Buff	1	0	Magenta
1	Buff	1	Buff	1	1	Orange

TYPICAL SYSTEM IMPLEMENTATION

The block diagram in Figure 23 shows how the VDG is related to other functional blocks in a typical system (non-6883). A negative row preset signal ($\overline{\text{RP}}$) generated by the VDG initializes the row scan counter for the external character generator once every twelve scan lines, while the negative horizontal sync ($\overline{\text{HS}}$) acts as clock to this counter. The negative field sync ($\overline{\text{FS}}$) generates an interrupt to the MPU, signifying that the display memory can be updated without interference with the VDG display function. This signal must not be confused with the system vertical sync signal. Field sync is activated by the end of the vertical display window and deactivated by the trailing edge of vertical sync. This gives the MPU a total of thirty-two scan lines or 2.03 ms to update the display memory. The MPU acknowledges the interrupt request from the VDG by bringing the negative memory select input (MS) to the VDG low. This puts the address bus output from the VDG into high-impedance state, thus relinquishing bus control to the MPU. The timing relationship of horizontal sync, row preset, and field sync are shown in Figures 7, 8, and 13.

The display memory is an element-by-element map of the display window on the screen. The VDG addresses the display memory storage locations in succession and translates their contents into luminance and chrominance levels. The frequency of address update is dependent on the length of the display element. Recall that display elements in major mode 1 are four periods and major mode 2 are either four or eight periods of the master clock. Data from the display memory is latched on every address transition. Hence, the data for the first display element must be stable four or eight periods before the horizontal display window depending on the display mode selected. This timing requirement is illustrated in Figure 6.

Examination of Figures 21 and 22 reveal that all display elements within major mode 1 are similar while those within major mode 2 are largely dissimilar. Therefore, mode switching between alphanumeric modes and semigraphic modes can be carried out freely. Care must be taken, however, when performing mode switching in major mode 2. The only compatible modes are between CG1 and RG1, and between CG6 and RG6. Minor mode switching within the same major mode in a given element row can be achieved as long as it is between compatible modes. It should be quite apparent that major mode switching on an element-by-element basis is impractical. It can be achieved, however, at the expense of added component count. The element formats in the VDG lend themselves to major mode switching between element rows. The presence of row preset in major mode 1 serves as a flag for the beginning of a new element row. Detection of this signal can initiate a major mode switch from 1 to 2.

Display memory size is a function of the display density. Quite often a graphic display contains shapes that are several times larger than that of the display elements in the VDG. This is particularly true of certain video games. Much of the display consists of a fixed background. The vertical size of a display element can be doubled or quadrupled by simply ignoring the lowest order or the first two low order vertical addresses, respectively, from the VDG. Reduction of address lines naturally leads to reduction in memory size. Another method of memory reduction is to store objects or object fragments in ROM and store their display addresses in the RAM portion of display memory. Here, the larger the object fragment, the greater the memory saving.

ASSOCIATED DEVICES

MC6883 — SYNCHRONOUS ADDRESS MULTIPLEXER (SAM)

This device, a linear bipolar companion to the MC6800 or MC6809E (external clock inputs), is primarily a VDG transparent-access controller. It allows the microprocessor to load and store to VDG display memory ("screen RAM") without waiting for a blank screen interval. Figure 1 shows a typical system using the SAM and the MC6809E. The inherent interleaved direct memory accesses (IDMA) which occur, continuously keep the VDG updated with the proper data (independently of mode), as well as keeping the dynamic memory (used as system memory with the MC6833) refreshed. This is done through a IDMA process as well, during the time the VDG does not need display data (horizontal and vertical sync times).

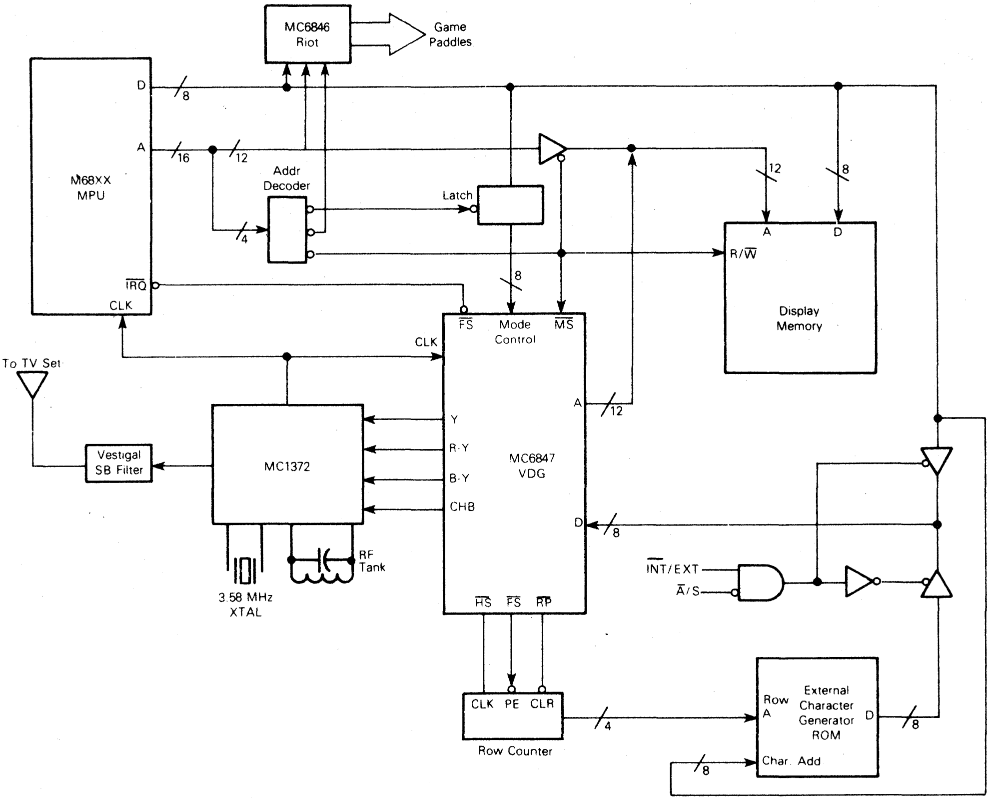
In addition to being a transparent memory access and dynamic memory controller, the SAM also functions as an external clock generator for the MC6800/6809E (slight additional circuitry is required for the MC6800).

MC1372/1373 CHROMA/RF MODULATOR

The MC1372 is a chrominance phase-shift modulator with built in RF up-converter. The part may be used without the RF modulator for chroma only, or the RF oscillator may be defeated and composite chrominance and luminance can be obtained.

The MC1373 is an RF modulator only (similar to the second half of the MC1372) and can be used to up-modulate separate luma and chroma signals at the receiver for high quality video reception.

FIGURE 23 — TYPICAL VDG SYSTEM



APPENDIX A
CUSTOM MC6847 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

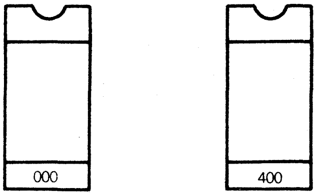
MDOS disk file

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 24 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 24 — 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 MC6847s 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). 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.

FIGURE A-2

Customer Name ______________________________

Address ______________________________

City ______________________________

Phone (__________)____________________State ______________________________Zip ____________________

Contact Ms/Mr ______________________________

Customer Part Number ______________________________

Pattern Media
2708 PROM
2716 PROM

MDOS Disk
(Note 2) ______________________________

Other (NOTE: Other media requires prior factory approval)

Signature ______________________________

Title ______________________________

MOTOROLA

MC6850
(1.0 MHz)
MC68A50
(1.5 MHz)
MC68B50
(2.0 MHz)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

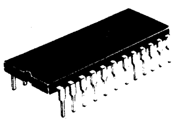
The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

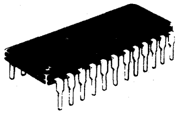
- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS
(N-CHANNEL, SILICON-GATE)

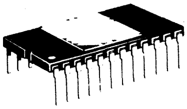
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



S SUFFIX
CERDIP PACKAGE
CASE 623

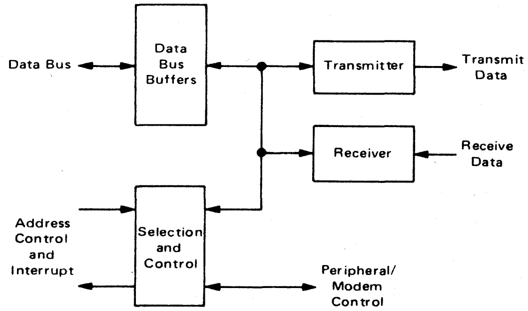


P SUFFIX
PLASTIC PACKAGE
CASE 709

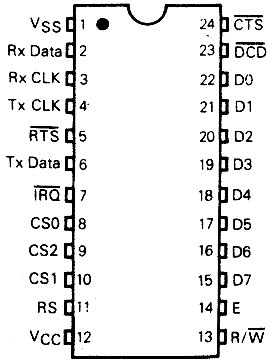


L SUFFIX
CERAMIC PACKAGE
CASE 716

MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



MAXIMUM RATINGS

Characteristics	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 MC6850, MC68A50, MC68B50 MC6850C, MC68A50C, MC68B50C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. 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 Ceramic Cerdip	θ_{JA}	 120 60 65	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, CS0, CS1, $\overline{CS2}$, Enable RS, Rx D, Rx C, $\overline{CTS}$, $\overline{DCD}$	I_{in}	–	1.0	2.5	µA
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	µA
Output High Voltage (I_{Load} = −205 µA, Enable Pulse Width <25 µs) (I_{Load} = −100 µA, Enable Pulse Width <25 µs)	D0-D7 Tx Data, $\overline{RTS}$	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$	 – –	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width <25 µs)		V_{OL}	–	–	$V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{LOH}	–	1.0	10	µA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	300	525	mW
Internal Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 E, Tx CLK, Rx CLK, R/W, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$	C_{in}	 – –	 10 7.0	 12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	RTS, Tx Data $\overline{IRQ}$	C_{out}	– –	– –	10 5.0	pF

SERIAL DATA TIMING CHARACTERISTICS

Characteristic		Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low (See Figure 1)	÷16, ÷64 Modes ÷1 Mode	PW_{CL}	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Pulse Width, High (See Figure 2)	÷16, ÷64 Modes ÷1 Mode	PW_{CH}	600 900	– –	450 650	– –	280 500	– –	ns
Data Clock Frequency	÷16, ÷64 Modes ÷1 Mode	f_C	– –	0.8 500	– –	1.0 750	– –	1.5 1000	MHz kHz
Data Clock-to-Data Delay for Transmitter (See Figure 3)		t_{TDD}	–	600	–	540	–	460	ns
Receive Data Setup Time (See Figure 4)	÷1 Mode	t_{RDS}	250	–	100	–	30	–	ns
Receive Data Hold Time (See Figure 5)	÷1 Mode	t_{RDH}	250	–	100	–	30	–	ns
Interrupt Request Release Time (See Figure 6)		t_{IR}	–	1.2	–	0.9	–	0.7	µs
Request-to-Send Delay Time (See Figure 6)		t_{RTS}	–	560	–	480	–	400	ns
Input Rise and Fall Times (or 10% of the pulse width if smaller)		t_r, t_f	–	1.0	–	0.5	–	0.25	µs

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

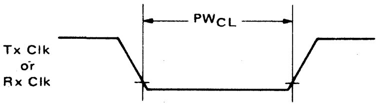


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

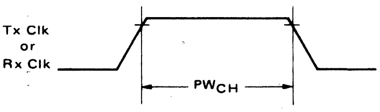


FIGURE 3 — TRANSMIT DATA OUTPUT DELAY

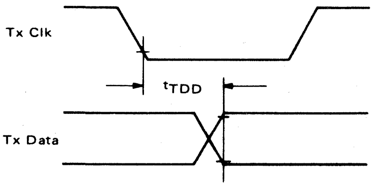


FIGURE 4 — RECEIVE DATA SETUP TIME
(÷1 Mode)

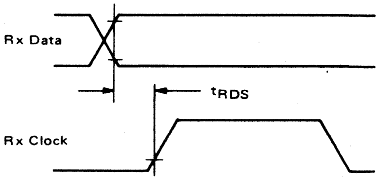


FIGURE 5 — RECEIVE DATA HOLD TIME
(÷1 Mode)

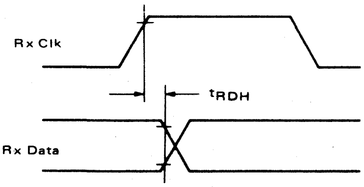
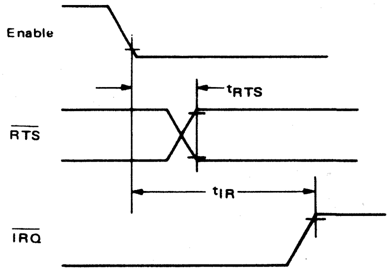


FIGURE 6 — REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES

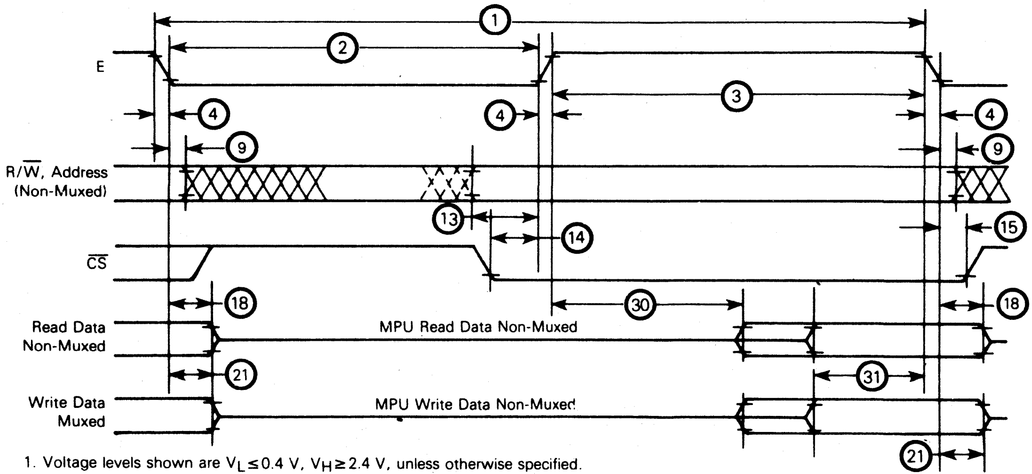


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

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

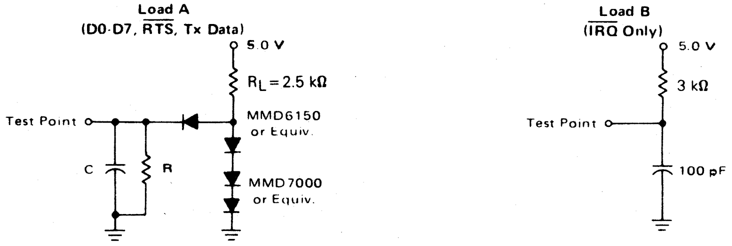
Ident. Number	Characteristic	Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 7 — BUS TIMING CHARACTERISTICS



1. Voltage levels shown are $V_L \leq 0.4$ 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.

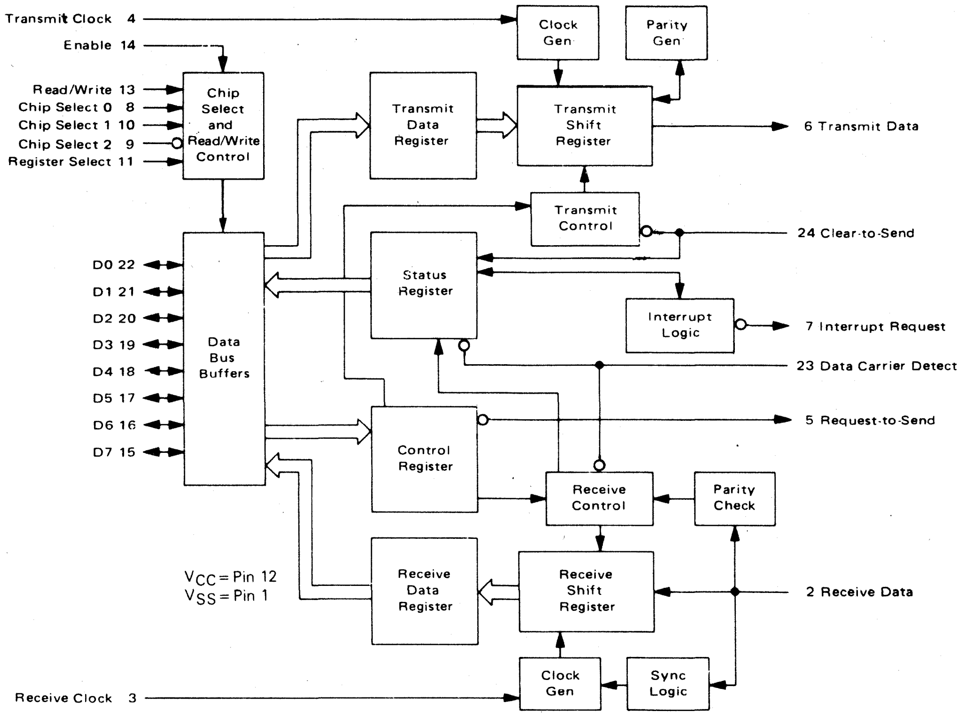
FIGURE 8 — BUS TIMING TEST LOADS



C = 130 pF for D0-D7
= 30 pF for RTS and Tx Data

R = 11.7 kΩ for D0-D7
= 24 kΩ for RTS and Tx Data

FIGURE 9 — EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the ACIA appears as two addressable memory locations. Internally, there are four registers: two read-only and two write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and three peripheral/modem control lines.

POWER ON/MASTER RESET

The master reset (CR0, CR1) should be set during system initialization to insure the reset condition and prepare for programming the ACIA functional configuration when the communications channel is required. During the first master reset, the $\overline{\text{IRQ}}$ and $\overline{\text{RTS}}$ outputs are held at level 1. On all other master resets, the $\overline{\text{RTS}}$ output can be programmed high or low with the $\overline{\text{IRQ}}$ output held high. Control bits CR5 and CR6 should also be programmed to define the state of $\overline{\text{RTS}}$ whenever master reset is utilized. The ACIA also contains internal power-on reset logic to detect the power line turn-on transition and hold the chip in a reset state to prevent erroneous output transitions prior to initialization. This circuitry depends on clean power turn-on transitions. The power-on reset is released by means of the bus-programmed master reset which must be applied prior to operating the ACIA. After master resetting the ACIA, the programmable Control Register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits, parity (even, odd, or none), etc.

TRANSMIT

A typical transmitting sequence consists of reading the ACIA Status Register either as a result of an interrupt or in the ACIA's turn in a polling sequence. A character may be written into the Transmit Data Register if the status read operation has indicated that the Transmit Data Register is empty. This character is transferred to a Shift Register where it is serialized and transmitted from the Transmit Data output preceded by a start bit and followed by one or two stop bits. Internal parity (odd or even) can be optionally added to the character and will occur between the last data bit and the first stop bit. After the first character is written in the Data Register, the Status Register can be read again to check for a Transmit Data Register Empty condition and current peripheral status. If the register is empty, another character can be loaded for transmission even though the first character is in the process of being transmitted (because of double buffering). The second character will be automatically transferred into the Shift Register when the first character transmission is completed. This sequence continues until all the characters have been transmitted.

RECEIVE

Data is received from a peripheral by means of the Receive Data input. A divide-by-one clock ratio is provided for an externally synchronized clock (to its data) while the divide-by-16 and 64 ratios are provided for internal synchronization. Bit synchronization in the divide-by-16 and 64 modes is initiated by the detection of 8 or 32 low samples on the receive line in the divide-by-16 and 64 modes respectively. False start bit deletion capability insures that a full half bit of a start bit has been received before the internal clock is synchronized to the bit time. As a character is being received, parity (odd or even) will be checked and the error indication will be available in the Status Register along with framing error, overrun error, and Receive Data Register full. In a typical receiving sequence, the Status Register is read to determine if a character has been received from a peripheral. If the Receiver Data Register is full, the character is placed on the 8-bit ACIA bus when a Read Data command is received from the MPU. When parity has been selected for a 7-bit word (7 bits plus parity), the receiver strips the parity bit (D7=0) so that data alone is transferred to the MPU. This feature reduces MPU programming. The Status Register can continue to be read to determine when another character is available in the Receive Data Register. The receiver is also double buffered so that a character can be read from the data register as another character is being received in the shift register. The above sequence continues until all characters have been received.

INPUT/OUTPUT FUNCTIONS

ACIA INTERFACE SIGNALS FOR MPU

The ACIA interfaces to the M6800 MPU with an 8-bit bidirectional data bus, three chip select lines, a register select line, an interrupt request line, read/write line, and enable line. These signals permit the MPU to have complete control over the ACIA.

ACIA Bidirectional Data (D0-D7) — The bidirectional data lines (D0-D7) allow for data transfer between the ACIA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ACIA read operation.

ACIA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers and clocks data to and from the ACIA. This signal will normally be a derivative of the MC6800 $\phi 2$ Clock or MC6809 E clock.

Read/Write (R/$\overline{\text{W}}$) — The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the ACIA's input/output data bus interface. When Read/Write is high (MPU Read cycle), ACIA output drivers are turned on and a selected register is read. When it is low, the ACIA output drivers are turned off and the MPU writes into a selected register. Therefore, the Read/Write signal is used to select read-only or write-only registers within the ACIA.

Chip Select (CS0, CS1, $\overline{\text{CS2}}$) — These three high-impedance TTL-compatible input lines are used to address the ACIA. The ACIA is selected when CS0 and CS1 are high and $\overline{\text{CS2}}$ is low. Transfers of data to and from the ACIA are then performed under the control of the Enable Signal, Read/Write, and Register Select.

Register Select (RS) — The Register Select line is a high-impedance input that is TTL compatible. A high level is used to select the Transmit/Receive Data Registers and a low level the Control/Status Registers. The Read/Write signal line is used in conjunction with Register Select to select the read-only or write-only register in each register pair.

Interrupt Request ($\overline{\text{IRQ}}$) — Interrupt Request is a TTL-compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The $\overline{\text{IRQ}}$ output remains low as long as the cause of the interrupt is present and the appropriate interrupt enable within the ACIA is set. The $\overline{\text{IRQ}}$ status bit, when high, indicates the $\overline{\text{IRQ}}$ output is in the active state.

Interrupts result from conditions in both the transmitter and receiver sections of the ACIA. The transmitter section causes an interrupt when the Transmitter Interrupt Enabled condition is selected (CR5•$\overline{\text{CR6}}$), and the Transmit Data Register Empty (TDRE) status bit is high. The TDRE status bit indicates the current status of the Transmitter Data Register except when inhibited by Clear-to-Send ($\overline{\text{CTS}}$) being high or the ACIA being maintained in the Reset condition. The interrupt is cleared by writing data into the Transmit Data Register. The interrupt is masked by disabling the Transmitter Interrupt via CR5 or CR6 or by the loss of $\overline{\text{CTS}}$ which inhibits the TDRE status bit. The Receiver section causes an interrupt when the Receiver Interrupt Enable is set and the Receive Data Register Full (RDRF) status bit is high, an Overrun has occurred, or Data Carrier Detect ($\overline{\text{DCD}}$) has gone high. An interrupt resulting from the RDRF status bit can be cleared by reading data or resetting the ACIA. Interrupts caused by Overrun or loss of $\overline{\text{DCD}}$ are cleared by reading the status register after the error condition has occurred and then reading the Receive Data Register or resetting the ACIA. The receiver interrupt is masked by resetting the Receiver Interrupt Enable.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data. Clock frequencies of 1, 16, or 64 times the data rate may be selected.

Transmit Clock (Tx CLK) — The Transmit Clock input is used for the clocking of transmitted data. The transmitter initiates data on the negative transition of the clock.

Receive Clock (Rx CLK) — The Receive Clock input is used for synchronization of received data. (In the ÷ 1 mode, the clock and data must be synchronized externally.) The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) — The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format. Synchronization with a clock for detection of data is accomplished internally when clock rates of 16 or 64 times the bit rate are used.

Transmit Data (Tx Data) — The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The ACIA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Request-to-Send and Data Carrier Detect.

Clear-to-Send ($\overline{\mathrm{CTS}}$) — This high-impedance TTL-compatible input provides automatic control of the transmitting end of a communications link via the modem Clear-to-Send active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

Request-to-Send ($\overline{\mathrm{RTS}}$) — The Request-to-Send output enables the MPU to control a peripheral or modem via the data bus. The RTS output corresponds to the state of the Control Register bits CR5 and CR6. When CR6=0 or both CR5 and CR6=1, the $\overline{\mathrm{RTS}}$ output is low (the active state). This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect ($\overline{\mathrm{DCD}}$) — This high-impedance TTL-compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem Data Carrier Detect output. The $\overline{\mathrm{DCD}}$ input inhibits and initializes the receiver section of the ACIA when high. A low-to-high transition of the Data Carrier Detect initiates an interrupt to the MPU to indicate the occurrence of a loss of carrier when the Receive Interrupt Enable bit is set. The Rx CLK must be running for proper $\overline{\mathrm{DCD}}$ operation.

ACIA REGISTERS

The expanded block diagram for the ACIA indicates the internal registers on the chip that are used for the status, control, receiving, and transmitting of data. The content of each of the registers is summarized in Table 1.

TRANSMIT DATA REGISTER (TDR)

Data is written in the Transmit Data Register during the negative transition of the enable (E) when the ACIA has been addressed with RS high and R/$\overline{\mathrm{W}}$ low. Writing data into the register causes the Transmit Data Register Empty bit in the Status Register to go low. Data can then be transmitted. If the transmitter is idling and no character is being transmitted, then the transfer will take place within 1-bit time of the trailing edge of the Write command. If a character is being transmitted, the new data character will commence as soon as the previous character is complete. The transfer of data causes the Transmit Data Register Empty (TDRE) bit to indicate empty.

RECEIVE DATA REGISTER (RDR)

Data is automatically transferred to the empty Receive Data Register (RDR) from the receiver deserializer (a shift register) upon receiving a complete character. This event causes the Receive Data Register Full bit (RDRF) in the status buffer to go high (full). Data may then be read through the bus by addressing the ACIA and selecting the Receive Data Register with RS and R/$\overline{\mathrm{W}}$ high when the ACIA is enabled. The non-destructive read cycle causes the RDRF bit to be cleared to empty although the data is retained in the RDR. The status is maintained by RDRF as to whether or not the data is current. When the Receive Data Register is full, the automatic transfer of data from the Receiver Shift Register to the Data Register is inhibited and the RDR contents remain valid with its current status stored in the Status Register.

TABLE 1 — DEFINITION OF ACIA REGISTER CONTENTS

Data Bus Line Number	Buffer Address			
	$\mathrm{RS} \bullet \overline{\mathrm{R}/\overline{\mathrm{W}}}$ Transmit Data Register	$\mathrm{RS} \bullet \mathrm{R}/\overline{\mathrm{W}}$ Receive Data Register	$\overline{\mathrm{RS}} \bullet \overline{\mathrm{R}/\overline{\mathrm{W}}}$ Control Register	$\overline{\mathrm{RS}} \bullet \mathrm{R}/\overline{\mathrm{W}}$ Status Register
	(Write Only)	(Read Only)	(Write Only)	(Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect ($\overline{\mathrm{DCD}}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear to Send ($\overline{\mathrm{CTS}}$)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request ($\overline{\mathrm{IRQ}}$)

* Leading bit = LSB = Bit 0

** Data bit will be zero in 7 bit plus parity modes.

*** Data bit is "don't care" in 7 bit plus parity modes.

CONTROL REGISTER

The ACIA Control Register consists of eight bits of write-only buffer that are selected when RS and $R/\overline{W}$ are low. This register controls the function of the receiver, transmitter, interrupt enables, and the Request-to-Send peripheral/modem control output.

Counter Divide Select Bits (CR0 and CR1) – The Counter Divide Select Bits (CR0 and CR1) determine the divide ratios utilized in both the transmitter and receiver sections of the ACIA. Additionally, these bits are used to provide a master reset for the ACIA which clears the Status Register (except for external conditions on $\overline{CTS}$ and $\overline{DCD}$) and initializes both the receiver and transmitter. Master reset does not affect other Control Register bits. Note that after power-on or a power fail/restart, these bits must be set high to reset the ACIA. After resetting, the clock divide ratio may be selected. These counter select bits provide for the following clock divide ratios:

CR1	CR0	Function
0	0	÷1
0	1	÷16
1	0	÷64
1	1	Master Reset

Word Select Bits (CR2, CR3, and CR4) – The Word Select bits are used to select word length, parity, and the number of stop bits. The encoding format is as follows:

CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

Word length, Parity Select, and Stop Bit changes are not buffered and therefore become effective immediately.

Transmitter Control Bits (CR5 and CR6) – Two Transmitter Control bits provide for the control of the interrupt from the Transmit Data Register Empty condition, the Request-to-Send ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{RTS}$ = low, Transmitting Interrupt Disabled.
0	1	$\overline{RTS}$ = low, Transmitting Interrupt Enabled.
1	0	$\overline{RTS}$ = high, Transmitting Interrupt Disabled.
1	1	$\overline{RTS}$ = low, Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled.

Receive Interrupt Enable Bit (CR7) – The following interrupts will be enabled by a high level in bit position 7 of the Control Register (CR7): Receive Data Register Full, Overrun, or a low-to-high transition on the Data Carrier Detect ($\overline{DCD}$) signal line.

STATUS REGISTER

Information on the status of the ACIA is available to the MPU by reading the ACIA Status Register. This read-only register is selected when RS is low and $R/\overline{W}$ is high. Information stored in this register indicates the status of the Transmit Data Register, the Receive Data Register and error logic, and the peripheral/modem status inputs of the ACIA.

Receive Data Register Full (RDRF), Bit 0 – Receive Data Register Full indicates that received data has been transferred to the Receive Data Register. RDRF is cleared after an MPU read of the Receive Data Register or by a master reset. The cleared or empty state indicates that the contents of the Receive Data Register are not current. Data Carrier Detect being high also causes RDRF to indicate empty.

Transmit Data Register Empty (TDRE), Bit 1 – The Transmit Data Register Empty bit being set high indicates that the Transmit Data Register contents have been transferred and that new data may be entered. The low state indicates that the register is full and that transmission of a new character has not begun since the last write data command.

Data Carrier Detect ($\overline{DCD}$), Bit 2 – The Data Carrier Detect bit will be high when the $\overline{DCD}$ input from a modem has gone high to indicate that a carrier is not present. This bit going high causes an Interrupt Request to be generated when the Receive Interrupt Enable is set. It remains high after the $\overline{DCD}$ input is returned low until cleared by first reading the Status Register and then the Data Register or until a master reset occurs. If the $\overline{DCD}$ input remains high after read status and read data or master reset has occurred, the interrupt is cleared, the $\overline{DCD}$ status bit remains high and will follow the $\overline{DCD}$ input.

Clear-to-Send ($\overline{CTS}$), Bit 3 – The Clear-to-Send bit indicates the state of the Clear-to-Send input from a modem. A low $\overline{CTS}$ indicates that there is a Clear-to-Send from the modem. In the high state, the Transmit Data Register Empty bit is inhibited and the Clear-to-Send status bit will be high. Master reset does not affect the Clear-to-Send status bit.

Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the first stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun condition begins at the midpoint of the last bit of the second character received in succession without a read of the RDR having occurred. The Overrun does not occur in the Status Register until the valid character prior to Overrun has been read. The RDRF bit remains set until the Overrun is reset. Character synchronization is maintained during the Overrun condition. The Overrun indication is reset after the reading of data from the Receive Data Register or by a Master Reset.

Parity Error (PE), Bit 6 – The parity error flag indicates that the number of highs (ones) in the character does not agree with the preselected odd or even parity. Odd parity is defined to be when the total number of ones is odd. The parity error indication will be present as long as the data character is in the RDR. If no parity is selected, then both the transmitter parity generator output and the receiver partiy check results are inhibited.

Interrupt Request ($\overline{\text{IRQ}}$), Bit 7 – The $\overline{\text{IRQ}}$ bit indicates the state of the $\overline{\text{IRQ}}$ output. Any interrupt condition with its applicable enable will be indicated in this status bit. Anytime the $\overline{\text{IRQ}}$ output is low the $\overline{\text{IRQ}}$ bit will be high to indicate the interrupt or service request status. $\overline{\text{IRQ}}$ is cleared by a read operation to the Receive Data Register or a write operation to the Transmit Data Register.

MOTOROLA

MC6852
(1.0 MHz)
MC68A52
(1.5 MHz)
MC68B52
(2.0 MHz)

SYNCHRONOUS SERIAL DATA ADAPTER (SSDA)

The MC6852 Synchronous Serial Data Adapter provides a bidirectional serial interface for synchronous data information interchange. It contains interface logic for simultaneously transmitting and receiving standard synchronous communications characters in bus organized systems such as the M6800 Microprocessor systems.

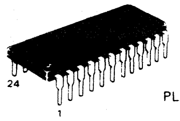
The bus interface of the MC6852 includes select, enable, read/write, interrupt, and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the synchronous data interface with synchronization, fill character insertion/deletion, and error checking. The functional configuration of the SSDA is programmed via the data bus during system initialization. Programmable control registers provide control for variable word lengths, transmit control, receive control, synchronization control, and interrupt control. Status, timing and control lines provide peripheral or modem control.

Typical applications include floppy disk controllers, cassette or cartridge tape controllers, data communications terminals, and numerical control systems.

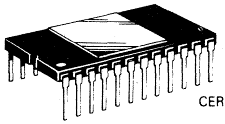
- Programmable Interrupts from Transmitter, Receiver, and Error Detection Logic
- Character Synchronization on One- or Two-Sync Codes
- External Synchronization Available for Parallel-Serial Operation
- Programmable Sync Code Register
- Up to 1.5 MHz Transmission
- Peripheral/Modem Control Functions
- Three Bytes of FIFO Buffering on Both Transmit and Receive
- 7-, 8-, or 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun, and Underflow Status

MOS
(N-CHANNEL, SILICON-GATE)

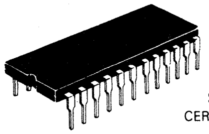
SYNCHRONOUS SERIAL DATA ADAPTER



P SUFFIX
PLASTIC PACKAGE
CASE 709

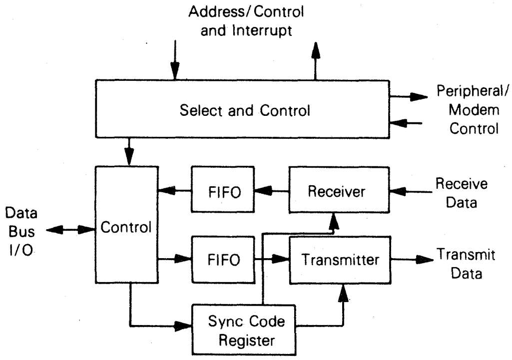


L SUFFIX
CERAMIC PACKAGE
CASE 716

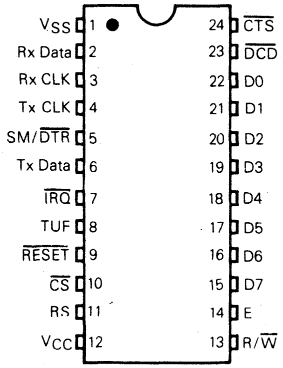


S SUFFIX
CERDIP PACKAGE
CASE 623

SYNCHRONOUS SERIAL DATA ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



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 MC6852, MC68A52, MC68B52 MC6852C, MC68A52C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advsied that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. 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 Package Ceramic Package Cerdip Package	θ_{JA}	 120 60 65	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS}+2.0	–	–	V
Input Low Voltage		V_{IL}	–	–	V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	Tx CLK, Rx CLK, Rx Data, Enable, $\overline{RESET}$, RS, R/$\overline{W}$, $\overline{CS}$, $\overline{DCD}$, $\overline{CTS}$	I_{in}	–	1.0	2.5	µA
Three-State (Off-State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.25 V)	D0-D7	I_{IZ}	–	2.0	10	µA
Output High Voltage (I_{Load} = −205 µA, Enable Pulse Width < 25 µs) (I_{Load} = −100 µA, Enable Pulse Width < 25 µs)	 D0-D7 TX Data, $\overline{DTR}$, TUF	V_{OH}	 V_{SS}+2.5 V_{SS}+2.4	 – –	 – –	V
Output Low Voltage (I_{Load} = 1.6 mA, Enable Pulse Width < 25 µs)		V_{OL}	–	–	V_{SS}+0.4	V
Output Leakage Current (Off-State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{OZ}	–	1.0	10	µA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	300	525	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 All Other Inputs	C_{in}	 – –	 – –	 12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	Tx Data, SM/$\overline{DTR}$, TUF $\overline{IRQ}$	C_{out}	–	–	10 5.0	pF

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

Characteristic	Symbol	MC6852		MC68A52		MC68B52		Unit
		Min	Max	Min	Max	Min	Max	
Serial Clock Pulse Width, Low (Figure 1)	PW_{CL}	700	–	400	–	280	–	ns
Serial Clock Pulse Width, High (Figure 2)	PW_{CH}	700	–	400	–	280	–	ns
Serial Clock Frequency (Rx CLK, Tx CLK)	f_C	–	600	–	1000	–	1500	kHz
Receive Data Setup Time (Figure 3, 7)	t_{RDSU}	350	–	200	–	160	–	ns
Receive Data Hold Time (Figure 3)	t_{RDH}	350	–	200	–	160	–	ns
Sync Match Delay Time (Figure 3)	t_{SM}	–	1.0	–	0.666	–	0.500	μs
Clock-to-Data Delay for Transmitter (Figure 4)	T_{DD}	–	1.0	–	0.666	–	0.500	μs
Transmitter Underflow (Figures 4, 6)	t_{TUF}	–	1.0	–	0.666	–	0.500	μs
$\overline{\text{DTR}}$ Delay Time (Figure 5)	t_{DTR}	–	1.0	–	0.666	–	0.500	μs
Interrupt Request Release Time (Figure 5)	t_{IR}	–	1.6	–	1.1	–	0.850	μs
$\overline{\text{RESET}}$ Pulse Width	t_{RESET}	1.0	–	0.666	–	0.500	–	μs
$\overline{\text{CTS}}$ Setup Time (Figure 6)	t_{CTS}	200	–	150	–	120	–	ns
$\overline{\text{DCD}}$ Setup Time (Figure 7)	t_{DCD}	500	–	350	–	250	–	ns
Input Rise and Fall Times (Except Enable)	t_r, t_f	–	1.0*	–	1.0*	–	1.0*	μs

*1.0 μs or 10% of the pulse width, whichever is smaller

FIGURE 1 — CLOCK PULSE WIDTH, LOW-STATE

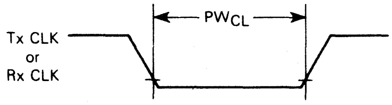


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

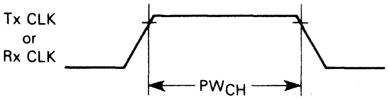
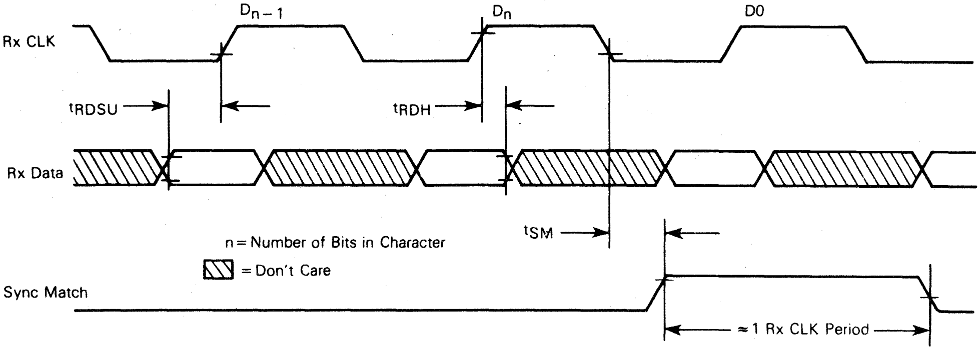
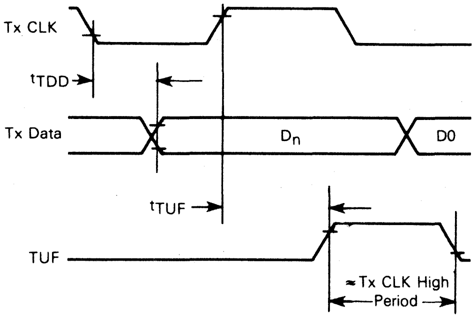


FIGURE 3 — RECEIVE DATA SETUP AND HOLD TIMES AND SYNC MATCH DELAY TIME



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 4 — TRANSMIT DATA OUTPUT DELAY AND TRANSMITTER UNDERFLOW DELAY TIME



n = Number of bits in character

FIGURE 5 — DATA TERMINAL READY AND INTERRUPT REQUEST RELEASE TIMES

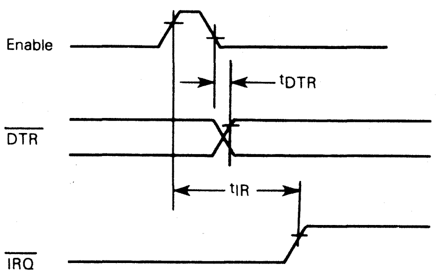


FIGURE 6 — CLEAR-TO-SEND SETUP TIME

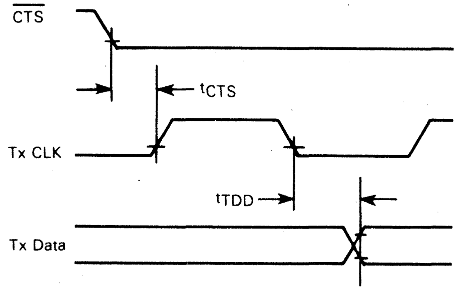
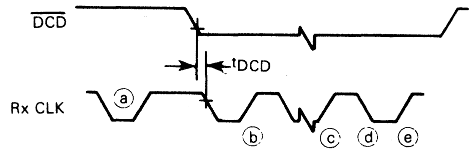


FIGURE 7 — DATA CARRIER DETECT SETUP TIME



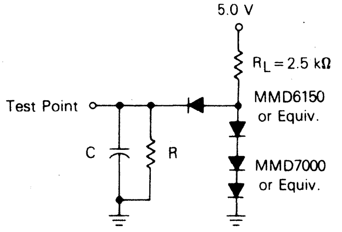
Notes:
- a. Must occur before $\overline{\text{DCD}}$ goes low.
- b. First data bit placed in Rx shift register.
- c. Last data bit of byte placed in Rx shift register.
- d. Rx data byte transferred from shift register to Rx FIFO.
- e. Clock edge required for generation of $\overline{\text{IRQ}}$ by RDA status.

Note: Refer to Figure 3 for the Rx data setup and hold times.

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.

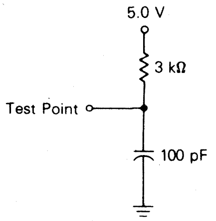
BUS TIMING TEST LOADS

Load A
(D0-D7, $\overline{\text{DTR}}$, Tx Data, TUF)



C = 130 pF for D0-D7
= 30 pF for $\overline{\text{DTR}}$, Tx Data, and TUF

Load B
($\overline{\text{IRQ}}$ Only)



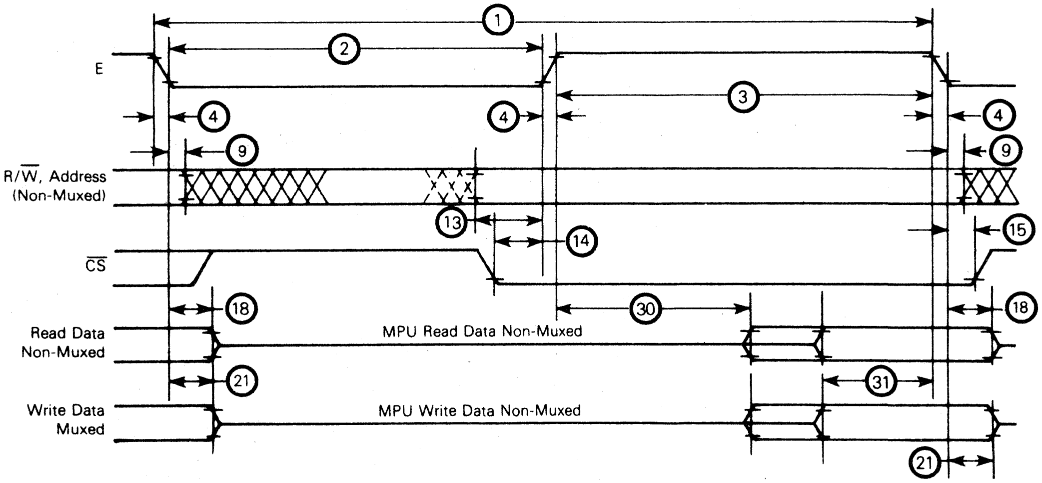
R = 11.7 kΩ for D0-D7
= 24 kΩ for $\overline{\text{DTR}}$, Tx Data, and TUF

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Indent Number	Characteristic	Symbol	MC6852		MC68A52		MC68B52		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50*	20	50*	30	50*	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

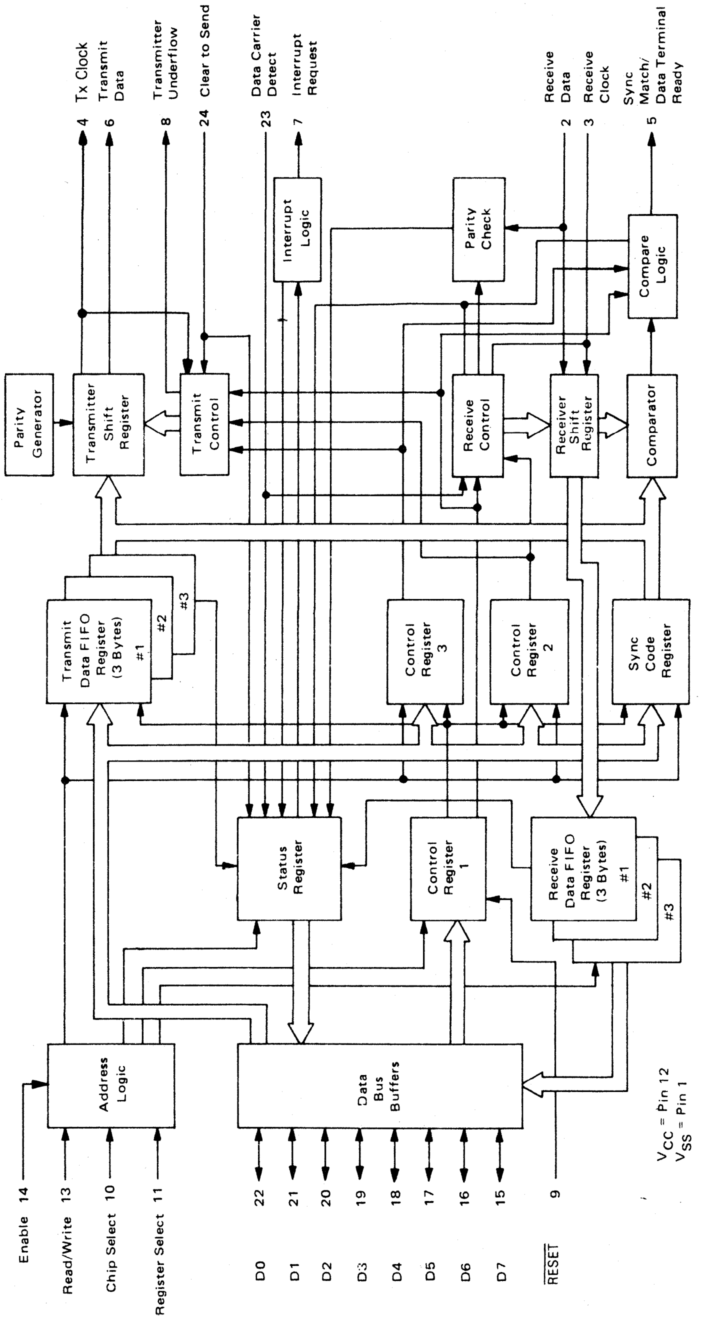
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

FIGURE 8 — BUS TIMING CHARACTERISTICS (READ/WRITE INFORMATION)



Notes:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

EXPANDED BLOCK DIAGRAM



DEVICE OPERATION

At the bus interface, the SSDA appears as two addressable memory locations. Internally, there are seven registers: two read-only and five write-only registers. The read-only registers are Status and Receive Data; the write-only registers are Control 1, Control 2, Control 3, Sync Code and Transmit Data. The serial interface consists of serial input and output lines with independent clocks, and four peripheral/modem control lines.

Data to be transmitted is transferred directly into the 3-byte Transmit Data First-In First-Out (FIFO) Register from the data bus. Availability of the input to the FIFO is indicated by the TDRA bit in the Status Register; once data is entered, it moves through the FIFO to the last empty location. Data at the output of the FIFO is automatically transferred from the FIFO to the Transmitter Shift Register as the shift register becomes available to transmit the next character. If data is not available from the FIFO (underflow condition), the Transmitter Shift Register is automatically loaded with either a sync code or an all "1's" character. The transmit seciton may be programmed to append even, odd, or no parity to the transmitted word. An external control line (Clear-to-Send) is provided to inhibit the transmitter without clearing the FIFO.

Serial data is accumulated in the receiver based on the synchronization mode selected. In the external sync mode, used for parallel-serial operation, the receiver is synchronized by the $\overline{DCD}$ (Data Carrier Detect) input (Figure 9) and transfers successive bytes of data to the input of the Receiver FIFO. The single-sync-character mode requires that a match occur between the Sync Code Register and one incoming character before data transfer to the FIFO begins. The two-sync-character mode requires that two sync codes be received in sequence to establish synchronization. Subsequent to synchronization in any mode, data is accumulated in the shift register, and parity is optionally checked. An indication of parity error is carried through the Receiver FIFO with each character to the last empty location. Availability of a word at the FIFO output is indicated by the RDA status bit in the Status Register, as is a parity error (PE).

The SSDA and its internal registers are selected by RS, $\overline{CS}$, Read/Write (R/$\overline{W}$) and Enable control lines. To configure the SSDA, Control Registers are selected and the appropriate bits set. The Status Register is addressable for reading status.

Other I/O lines, in addition to Clear-to-Send ($\overline{CTS}$) and Data Carrier Detect ($\overline{DCD}$), include SM/DTR (Sync Match/Data Terminal Ready) and Transmitter Underflow (TUF). The transmitter and receiver each have individual clock inputs allowing simultaneous operation under separate clock control. Signals to the microprocessor are the Data Bus and Interrupt Request ($\overline{IRQ}$).

INITIALIZATION

During a power-on sequence, the SSDA is reset via the $\overline{RESET}$ input and internally latched in a reset condition to prevent erroneous output transitions. The Receiver Shift Register is set to all "1's". The Sync Code Register, Control Register 2, and Control Register 3 should be programmed prior to the programmed release of the Transmitter and/or Receiver Reset bits; these bits in Control Register 1 should be cleared after the $\overline{RESET}$ line has gone high.

TRANSMITTER OPERATION

Data is transferred to the transmitter section in parallel form by means of the data bus and Transmit Data FIFO. The Transmit Data FIFO is a 3-byte register whose status is indicated by the Transmitter Data Register Available status bit (TDRA) and its associated interrupt enable bit. Data is transferred through the FIFO on negative edges of Enable (E) pulses. Two data transfer modes are provided in the SSDA. The 1-byte transfer mode provides for writing data to the transmitter section (and reading from the receiver section) one byte at a time. The 2-byte transfer mode provides for writing two data characters in succession.

Data will automatically transfer from the last register location in the Transmit Data FIFO (when it contains data) to the Transmitter Shift Register during the last half of the last bit of the previous character. A character is transferred into the Shift Register by the Transmitter Clock. Data is transmitted *LSB first*, and odd or even parity can be optionally appended. The unused bit positions in short word length characters, from the data bus, are "don't cares". (Note: The data bus inputs may be reversed for applications requiring the MSB to be transferred first, e.g., IBM format for floppy disks; however, care must be taken to properly program the control registers — Table 1 will have its bit positions reversed.)

When the Shift Register becomes empty, and data is *not* available for transfer from the Transmit Data FIFO, an "underflow" occurs, and a character is inserted into the transmitter data stream to maintain character synchronization. The character transmitted on underflow will be either a "Mark" (all "1's") or the contents of the Sync Code Register, depending upon the state of the Transmit Sync Code on Underflow control bit. The underflow condition is indicated by a pulse (=1 Tx CLK high period) on the Underflow output (when in Tx Sync on underflow mode). The Underflow output occurs coincident with the transfer of the last half of the last bit preceding the underflow character. The Underflow status bit is set until cleared by means of the Clear Underflow control bit. This output may be used in floppy disk systems to synchronize write operations and for appending CRCC.

Transmission is initiated by clearing the Transmitter Reset bit in Control Register 1. When the Transmitter Reset bit is cleared, the first *full* positive half-cycle of the Transmit Clock will initiate the transmit cycle, with the transmission of data or underflow characters beginning on the negative edge of the Transmit Clock pulse which started the cycle. If the Transmit Data FIFO was not loaded, an underflow character will be transmitted (see Figure 4).

The Clear-to-Send ($\overline{CTS}$) input provides for automatic control of the transmitter by means of external system hardware; e.g., the modem $\overline{CTS}$ output provides the control in a data communications system. The $\overline{CTS}$ input resets and inhibits the transmitter section when high, but does not reset the Transmit Data FIFO. The TDRA status bit is inhibited by $\overline{CTS}$ being high in either the one-sync character or two-sync character mode of operation. In the external sync mode, TDRA is unaffected by $\overline{CTS}$ in order to provide Transmit Data FIFO status for preloading and operating the transmitter under the control of the $\overline{CTS}$ input. When the Transmitter Reset bit (Tx Rs) is set, the Transmit Data FIFO is cleared and the TDRA status bit is cleared. After one E clock has occurred, the Transmit Data FIFO becomes available for new data with TDRA inhibited.

RECEIVER OPERATION

Data and a presynchronized clock are provided to the SSDA receiver section by means of the Receive Data (Rx Data) and Receive Clock (Rx CLK) inputs. The data is a continuous stream of binary data bits without means for identifying character boundaries within the stream. It is, therefore, necessary to achieve character synchronization for the data at the *beginning* of the data block. Once synchronization is achieved, it is assumed to be retained for all successive characters within the block.

Data communications systems utilize the detection of sync codes during the initial portion of the preamble to establish character synchronization. This requires the detection of a single code or two successive sync codes. Floppy disk and cartridge tape units require sixteen bits of defined preamble and cassettes require eight bits of preamble to establish the reference for the start of record. All three are functionally equivalent to the detection of sync codes. Systems which do not utilize code detection techniques require custom logic external to the SSDA for character synchronization and use of the parallel-to-serial (external sync) mode. (Note: The Receiver Shift Register is set to ones when reset.)

SYNCRHONIZATION

The SSDA provides three operating modes with respect to character synchronization: one-sync-character mode, two-sync-character mode, and external sync mode. The external sync mode requires synchronization and control of the receiving section through the Data Carrier Detect ($\overline{DCD}$) input (see Figure 7). This external synchronization could consist of direct line control from the transmitting end of the serial data link or from external logic designed to detect the start of the message block. The one-sync-character mode searches on a bit-by-bit basis until a match is achieved between the data in the Shift Register and the Sync Code Register. The match indicates character synchronization is complete and will be retained for the message block. In the two-sync-character mode, the receiver searches for the first sync code match on a bit-by-bit basis and then looks for a second *successive* sync code character prior to establishing character synchronization. If the second sync code character is not received, the bit-by-bit search for the first sync code is resumed.

Sync codes received prior to the completion of synchronization (one or two character) are not transferred to the Receive Data FIFO. Redundant sync codes during the preamble or sync codes which occur as "fill characters" can automatically be stripped from the data, when the Strip Sync control bit is set, to minimize system loading. The character synchronization will be retained until cleared by means of the Clear Sync bit, which also inhibits synchronization search when set.

RECEIVING DATA

Once synchronization has been achieved, subsequent characters are automatically transferred into the Receive Data FIFO and clocked through the FIFO to the last empty location by E pulses (MPU System $\phi 2$). The Receiver Data Available status bit (RDA) indicates when data is available to be read from the last FIFO location (#3) when in the 1-byte transfer mode. The 2-byte transfer mode causes the RDA status bit to indicate data is available when the last two FIFO register locations are full. Data being available in the Receive Data FIFO causes an interrupt request if the Receiver Interrupt Enable (RIE) bit is set. The MPU will then read the SSDA Status Register which will indicate that data is available for the MPU read from the Receive Data FIFO register. The IRQ and RDA status bits are reset by a read from the FIFO. If more than one character has been received and is resident in the Receive Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA and IRQ status bits will again be set. The read data operation for the 2-byte transfer mode requires an intervening E clock between reads to allow the FIFO data to shift. Optional parity is automatically checked as data is received, and the parity status condition is maintained with each character until the data is read from the Receive Data FIFO. Parity errors will cause an interrupt request if the Error Interrupt Enable (EIE) has been set. The parity bit is not transferred to the data bus but must be checked in the Status Register. NOTE: In the 2-byte transfer mode, parity should be checked prior to reading the second byte, since a FIFO read clears the error bit.

Other status bits which pertain to the receiver section are Receiver Overrun and Data Carrier Detect ($\overline{DCD}$). The Overrun status bit is automatically set when a transfer of a character to the Receive Data FIFO occurs and the first register of the Receive Data FIFO is full. Overrun causes an interrupt if Error Interrupt Enable (EIE) has been set. The transfer of the overrunning character into the FIFO causes the previous character in the FIFO input register location to be lost. The Overrun status bit is cleared by reading the Status Register (when the overrun condition is present), followed by a Receive data FIFO Register read. Overrun cannot occur and be cleared without providing an opportunity to detect its occurrence via the Status Register.

A positive transition on the $\overline{DCD}$ input causes an interrupt if the EIE control bit has been set. The interrupt caused by $\overline{DCD}$ is cleared by reading the Status Register when the $\overline{DCD}$ status bit is high, followed by a Receive data FIFO read. The $\overline{DCD}$ status bit will subsequently follow the state of the $\overline{DCD}$ input when it goes low.

INPUT/OUTPUT FUNCITONS

SSDA INTERFACE SIGNALS FOR MPU

The SSDA interfaces to the MC6800 MPU with an 8-bit bi-directional data bus, a chip-select line, a register-select line, an interrupt-request line, read/write line, an enable line, and a reset line. These signals, in conjunction with the MC6800 VMA output, permit the MPU to have complete control over the SSDA.

SSDA Bi-Directional Data (D0-D7) — The bi-directional data lines (D0-D7) allow for data transfer between the SSDA and the MPU. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an SSDA read operation.

SSDA Enable (E) — The Enable signal, E, is a high-impedance TTL-compatible input that enables the bus input/output data buffers, clocks data to and from the SSDA, and moves data through the FIFO Registers.

Read/Write (R/$\overline{\text{W}}$) – The Read/Write line is a high-impedance input that is TTL compatible and is used to control the direction of data flow through the SSDA's input/output data bus interface. When Read/Write is high (MPU read cycle), SSDA output drivers are turned on if the chip is selected and a selected register is read. When it is low, the SSDA output drivers are turned off and the MPU writes into a selected register. The Read/Write signal is also used to select read-only or write-only registers within the SSDA.

Chip Select ($\overline{\text{CS}}$) – This high-impedance TTL-compatible input line is used to address the SSDA. The SSDA is selected when $\overline{\text{CS}}$ is low. VMA should be used in generating the $\overline{\text{CS}}$ input to insure that false selects will not occur. Transfers of data to and from the SSDA are then performed under the control of the Enable signal, Read/Write, and Register Select.

Register Select (RS) – The Register Select line is a high-impedance input that is TTL compatible. A high level is used to select Control Registers C2 and C3, the Sync Code Register, and the Transmit/Receive Data Registers. A low level selects the Control 1 and Status Registers (see Table 1).

Interrupt Request ($\overline{\text{IRQ}}$) – Interrupt Request is a TTL compatible, open-drain (no internal pullup), active low output that is used to interrupt the MPU. The Interrupt Request remains low until cleared by the MPU.

$\overline{\text{RESET}}$ Input – The $\overline{\text{RESET}}$ input provides a means of resetting the SSDA from an external source. In the low state, the $\overline{\text{RESET}}$ input causes the following:

1. Receiver Reset (Rx Rs) and Transmitter Reset (Tx Rs) bits are set causing both the receiver and transmitter sections to be held in a reset condition.
2. Peripheral Control bits PC1 and PC2 are reset to zero, causing the SM/$\overline{\text{DTR}}$ output to be high.
3. The Error Interrupt Enable (EIE) bit is reset.
4. An internal synchronization mode is selected.
5. The Transmitter Data Register Available (TDRA) status bit is cleared and inhibited.
6. The Receiver Shift Register is set to 1's.

When $\overline{\text{RESET}}$ returns high (the inactive state), the transmitter and receiver sections will remain in the reset state until the Receiver Reset and Transmitter Reset bits are cleared via the data bus under software control. The control Register bits affected by $\overline{\text{RESET}}$ (Rx Rs, Tx Rs, PC1, PC2, EIE, and E/I Sync) cannot be changed when $\overline{\text{RESET}}$ is low.

CLOCK INPUTS

Separate high-impedance TTL-compatible inputs are provided for clocking of transmitted and received data.

Transmit Clock (Tx CLK) – The Transmit Clock input is used for the clocking of transmitted data. The transmitter shifts data on the negative transition of the clock.

Receive Clock (Rx CLK) – The Receive Clock input is used for clocking in received data. The clock and data must be synchronized externally. The receiver samples the data on the positive transition of the clock.

SERIAL INPUT/OUTPUT LINES

Receive Data (Rx Data) – The Receive Data line is a high-impedance TTL-compatible input through which data is received in a serial format.

Transmit Data (Tx Data) – The Transmit Data output line transfers serial data to a modem or other peripheral.

PERIPHERAL/MODEM CONTROL

The SSDA includes several functions that permit limited control of a peripheral or modem. The functions included are Clear-to-Send, Sync Match/Data Terminal Ready, Data Carrier Detect, and Transmitter Underflow.

Clear-to-Send ($\overline{\text{CTS}}$) – The $\overline{\text{CTS}}$ input provides a real-time inhibit to the transmitter section (the Tx Data FIFO is not disturbed). A positive $\overline{\text{CTS}}$ transition resets the Tx Shift Register and inhibits the TDRA status bit and its associated interrupt in both the one-sync-character and two-sync-character modes of operation. TDRA is not affected by the $\overline{\text{CTS}}$ input in the external sync mode.

The positive transition of $\overline{\text{CTS}}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{\text{CTS}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by writing a "1" in the Clear $\overline{\text{CTS}}$ bit in Control Register 3 or in the Transmitter Reset bit. The $\overline{\text{CTS}}$ status bit subsequently follows the $\overline{\text{CTS}}$ input when it goes low.

The $\overline{\text{CTS}}$ input provides character timing for transmitter data when in the external sync mode. Transmission is initiated on the negative transition of the first *full* positive clock pulse of the transmitter clock (Tx CLK) after the release of $\overline{\text{CTS}}$ (see Figure 6).

Data Carrier Detect ($\overline{\text{DCD}}$) – The $\overline{\text{DCD}}$ input provides a real-time inhbit to the receiver section (the Rx FIFO is not disturbed). A positive $\overline{\text{DCD}}$ transition resets and inhibts the receiver section except for the Receive FIFO and the RDRA status bit and its associated $\overline{\text{IRQ}}$.

The positive transition of $\overline{\text{DCD}}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{\text{DCD}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by reading the Status Register and then the Receiver FIFO, or by writing a "1" into the Receiver Reset bit. The $\overline{\text{DCD}}$ status bit subsequently follows the $\overline{\text{DCD}}$ input when it goes low. The $\overline{\text{DCD}}$ input provides character synchronization timing for the receiver during the external sync mode of operation. The receiver will be initialized and data will be sampled on the positive transition of the first *full* Receive Clock cycle after release of $\overline{\text{DCD}}$ (see Figure 7).

Sync Match/Data Terminal Ready (SM/$\overline{\text{DTR}}$) – The SM/DTR output provides four functions (see Table 1) depending on the state of the PC1 and PC2 control bits. When the Sync Match mode is selected (PC = "1", PC2 = "0"), the output provides a one-bit-wide pulse when a sync code is detected. This pulse occurs for each sync code match even if the receiver has already attained synchronization. The SM output is inhibited when PC2 = "1". The $\overline{\text{DTR}}$ mode (PC1 = "0") provides an output level corresponding to the complement of PC2 ($\overline{\text{DTR}}$ = "0" when PC2 = "1"). (See Table 1.)

TABLE 1 — SSDA PROGRAMMING MODEL

Register	Control Inputs		Address Control		Register Content							
	RS	R/W	AC2	AC1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Status (S)	0	1	X	X	Interrupt Request (IRQ)	Receiver Parity Error (PE)	Receiver Overrun (Rx Ovrn)	Transmitter Underflow (TUF)	$\overline{\text{Clear-to-Send}}$ ($\overline{\text{CTS}}$)	$\overline{\text{Data Carrier Detect}}$ ($\overline{\text{DCD}}$)	Transmitter Data Register Available (TDRA)	Receiver Data Available (RDA)
Control 1 (C1)	0	0	X	X	Address Control 2 (AC2)	Address Control 1 (AC1)	Receiver Interrupt Enable (RIE)	Transmitter Interrupt Enable (TIE)	Clear Sync	Strip Sync Characters (Strip Sync)	Transmitter Reset (Tx Rs)	Receiver Reset (Rx Rs)
Receive Data FIFO	1	1	X	X	D7	D6	D5	D4	D3	D2	D1	D0
Control 2 (C2)	1	0	0	0	Error Interrupt Enable (EIE)	Transmit Sync Code on Underflow (Tx Sync)	Word Length Select 3 (WS3)	Word Length Select 2 (WS2)	Word Length Select 1 (WS1)	1-Byte/2-Byte Transfer (1-Byte/2-Byte)	Peripheral Control 2 (PC2)	Peripheral Control 1 (PC1)
Control 3 (C3)	1	0	0	1	Not Used	Not Used	Not Used	Not Used	Clear Transmitter Underflow Status (CTUF)	Clear $\overline{\text{CTS}}$ Status (Clear $\overline{\text{CTS}}$)	One-Sync-Character/ Two-Sync Character Mode Control (1 Sync/ 2 Sync)	External/ Internal Sync Mode Control (E/I Sync)
Sync Code	1	0	1	0	D7	D6	D5	D4	D3	D2	D1	D0
Transmit Data FIFO	1	0	1	1	D7	D6	D5	D4	D3	D2	D1	D0

X = Don't care

STATUS REGISTER

IRQ	Bit 7	The IRQ flag is cleared when the source of the IRQ is cleared. The source is determined by the enables in the Control Registers: TIE, RIE, EIE.
	Bits 6–0	indicate the SSDA status at a point in time, and can be *reset* as follows:
PE	Bit 6	Read Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0).
Rx Ovrn	Bit 5	Read Status and then Rx Data FIFO, or a "1" into Rx Rs (C1 Bit 0).
TUF	Bit 4	A "1" into CTUF (C3 Bit 3) or into Tx Rs (C1 Bit 1).
$\overline{\text{CTS}}$	Bit 3	A "1" into Clear $\overline{\text{CTS}}$ (C3 Bit 2) or a "1" into Tx Rs (C1 Bit1)
$\overline{\text{DCD}}$	Bit 2	Read Status and then Rx Data FIFO or a "1" into Rx Rs (C1 Bit 0)
TDRA	Bit 1	Write into Tx Data FIFO.
RDA	Bit 0	Read Rx Data FIFO.

CONTROL REGISTER 1

AC2, AC1	Bits 7, 6	Used to access other registers, as shown above.
RIE	Bit 5	When "1", enables interrupt on RDA (S Bit 0).
TIE	Bit 4	When "1", enables interrupt on TDRA (S Bit 1).
Clear Sync	Bit 3	When "1", clears receiver character synchronization.
Strip Sync	Bit 2	When "1", strips all sync codes from the received data stream.
Tx Rs	Bit 1	When "1", resets and inhibits the transmitter section.
Rx Rs	Bit 0	When "1", resets and inhibits the receiver section.

CONTROL REGISTER 3

CTUF	Bit 3	When "1", clears TUF (S Bit 4), and IRQ if enabled.
Clear $\overline{\text{CTS}}$	Bit 2	When "1", clears $\overline{\text{CTS}}$ (S Bit 3), and IRQ if enabled.
1 Sync/2 Sync	Bit 1	When "1", selects the one-sync-character mode; when "0", selects the two-sync-character mode.
E/I Sync	Bit 0	When "1", selects the external sync mode; when "0", selects the internal sync mode.

CONTROL REGISTER 2

EIE	Bit 7	When "1", enables the PE, Rx Ovrn, TUF, $\overline{\text{CTS}}$, and $\overline{\text{DCD}}$ interrupt flags (S Bits 6 through 2).
Tx Sync	Bit 6	When "1", allows sync code contents to be transferred on underflow, and **enables the TUF Status bit and output. When "0", an all mark character is transmitted on underflow.**
WS3, 2, 1	Bits 5–3	Word Length Select

Bit 5 WS3	Bit 4 WS2	Bit 3 WS1	Word Length
0	0	0	6 Bits + Even Parity
0	0	1	6 Bits + Odd Parity
0	1	0	7 Bits
0	1	1	8 Bits
1	0	0	7 Bits + Even Parity
1	0	1	7 Bits + Odd Parity
1	1	0	8 Bits + Even Parity
1	1	1	8 Bits + Odd Parity

1-Byte/2-Byte	Bit 2	When "1", enables the TDRA and RDA bits to indicate when a 1-byte transfer can occur; when "0", the TDRA and RDA bits indicate when a 2-byte transfer can occur.
PC2, PC1	Bits 1–0	SM/$\overline{\text{DTR}}$ Output Control

Bit 1 PC2	Bit 0 PC1	SM/$\overline{\text{DTR}}$ Output at Pin 5
0	0	1
0	1	Pulse ⎍, 1-Bit Wide, on SM
1	0	0
1	1	SM Inhibited, 0

NOTE: When the SSDA is used in applications requiring the MSB of data to be received and transmitted first, the data bus inputs to the SSDA may be reversed (D0 to D7, etc.). Caution must be used when this is done since the **bit positions in this table will be reversed, and the parity should not be selected.**

Transmitter Underflow (TUF) – The Underflow output indicates the occurrence of a transfer of a "fill character" to the Transmitter Shift Register when the last location (#3) in the Transmit Data FIFO is emtpy. The Underflow output pulse is approximately one Tx CLK high period wide and occurs during the last half of the last bit of the character preceding the "Underflow" (see Figure 4). The Underflow output pulse does not occur when the Tx Sync bit is in the reset state.

SSDA REGISTERS

Seven registers in the SSDA can be accessed by means of the data bus. The registers are defined as read-only or write-only according to the direction of information flow. The Register Select input (RS) selects two registers in each state, one being read-only and the other write-only. The Read/Write input (R/$\overline{\text{W}}$) defines which of the two selected registers will actually be accessed. Four registers (two read-only and two write-only) can be accessed via the bus at any particular time. These registers and the required addressing are defined in Table 1.

CONTROL REGISTER 1 (C1)

Control Register 1 is an 8-bit write-only register that can be directly addressed from the data bus. Control Register 1 is accessed when RS = "0" and R/$\overline{\text{W}}$ = "0".

Receiver Reset (Rx Rs), C1 Bit 0 – The Receiver Reset control bit provides both a reset and inhibit function to the receiver section. When Rx Rs is set, it clears the receiver control logic, sync logic, error logic, Rx Data FIFO Control, Parity Error status bit, and $\overline{\text{DCD}}$ interrupt. The Receiver Shift Register is set to ones. The Rx Rs bit must be cleared after the occurrence of a low level on $\overline{\text{RESET}}$ in order to enable the receiver section of the SSDA.

Transmitter Reset (Tx Rs), C1 Bit 1 – The Transmitter Reset control bit provides both reset and inhibit to the transmitter section. When Tx Rs is set, it clears the transmitter control section, Transmitter Shift Register, Tx Data FIFO Control (the Tx Data FIFO can be reloaded after one E clock pulse), the Transmitter Underflow status bit, and the $\overline{\text{CTS}}$ interrupt, and inhibits the TDRA status bit (in the one-sync-character and two-sync-character modes).The Tx Rs bit must be cleared after the occurrence of a low level on $\overline{\text{RESET}}$ in order to enable the transmitter section of the SSDA. If the Tx FIFO is not preloaded, it must be loaded immediately after the Tx Rs release to prevent a transmitter underflow condition..

Strip Synchronization Characters (Strip Sync), C1 Bit 2 – If the Strip Sync bit is set, the SSDA will automatically strip all received characters which match the contents of the Sync Code Register. The characters used for synchronization (one or two characters of sync) are always stripped from the received data stream.

Clear Synchronization (Clear Sync), C1 Bit 3 – The Clear Sync control bit provides the capability of dropping receiver character synchronization and inhibiting resynchronization. The Clear Sync bit is set to clear and inhibit receiver synchronization in *all* modes and is reset to zero to enable resynchronization.

Transmitter Interrupt Enable (TIE), C1 Bit 4 – TIE enables both the Interrupt Request output ($\overline{\text{IRQ}}$) and Interrupt Request status bit to indicate a transmitter service request. When TIE is set and the TDRA status bit is high, the $\overline{\text{IRQ}}$ output will go low (the active state) and the IRQ status bit will go high.

Receiver Interrupt Enable (RIE), C1 Bit 5 – RIE enables both the Interrupt Request output ($\overline{\text{IRQ}}$) and the Interrupt Request status bit to indicate a receiver service request. When RIE is set and the RDA status bit is high, the $\overline{\text{IRQ}}$ output will go low (the active state) and the IRQ status bit will go high.

Address Control 1 (AC1) and Address Control 2 (AC2), C1 Bits 6 and 7 – AC1 and AC2 select one of the write-only registers – Control 2, Control 3, Sync Code, or Tx Data FIFO – as shown in Table 1, when RS = "1" and R/W = "0".

CONTROL REGISTER 2 (C2)

Control Register 2 is an 8-bit write-only register which can be programmed from the data bus when the Address Control bits in Control Register 1 (AC1 and AC2) are reset, RS = "1" and R/$\overline{\text{W}}$ = "0".

Peripheral Control (PC1) and Peripheral Control 2 (PC2), C2 Bits 0 and 1 – Two control bits, PC1 and PC2, determine the operating characteristics of the Sync Match/$\overline{\text{DTR}}$ output. PC1, when high, selects the Sync Match mode. PC2 provides the inhibit/enable control for the SM/$\overline{\text{DTR}}$ output in the Sync Match mode. A one-bit-wide pulse is generated at the output when PC2 is "0", and a match occurs between the contents of the Sync Code Register and the incoming data even if sync is inhibited (Clear Sync bit = "1"). The Sync Match pulse is referenced to the negative edge of Rx - CLK pulse causing the match (see Figure 3).

The Data Terminal Ready ($\overline{\text{DTR}}$) mode is selected when PC1 is low. When PC2 = "1" the SM/$\overline{\text{DTR}}$ output = "0" and vice versa. The operation of PC2 and PC1 is summarized in Table 1.

1-Byte/2-Byte Transfer (1-Byte/2-Byte), C2, Bit 2 – When 1-Byte/2-Byte is set, the TDRA and RDA status bits will indicate the availabitliy of their respective data FIFO registers for a single-byte data transfer. Alternately, if 1-Byte/2-Byte is reset, the TDRA and RDA status bits indicate when two bytes of data can be moved without a second status read. An intervening Enable pulse must occur between data transfers.

Word Length Selects (WS1, WS2, WS3), C2 Bits 3, 4, 5 – Word Length Select bits WS1, WS2, and WS3 select word lengths of 7, 8, or 9 bits including parity as shown in Table 1.

Transmit Sync Code on Underflow (Tx Sync), C2 Bit 6 – When Tx Sync is set, the transmitter will automatically send a sync character when data is not available for transmission. If Tx Sync is reset, the transmitter will transmit a Mark character (including the parity bit position) on underflow. When the underflow is detected, a pulse approximately one Tx CLK high period wide will occur on the underflow output if the Tx Sync bit is set. Internal parity generation is inhibited during underflow except for sync code fill character transmission in 8-bit plus parity word lengths.

Error Interrupt Enable (EIE), C2 Bit 7 – When EIE is set, the IRQ status bit will go high and the IRQ output will go low if:

1. A receiver overrun occurs. The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
2. DCD input has gone to a "1". The interrupt is cleared by reading the Status Register and reading the Rx Data FIFO.
3. A parity error exists for the character in the last location (#3) of the Rx Data FIFO. The interrupt is cleared by reading the Rx Data FIFO.
4. The CTS input has gone to a "1". The interrupt is cleared by writing a "1" in the Clear CTS bit, C3 bit 2, or by a Tx Reset.
5. The transmitter has underflowed (in the Tx Sync on Underflow mode). The interrupt is cleared by writing a "1" into the Clear Underflow, C3 bit 3, or Tx Reset.

When EIE is a "0", the IRQ status bit and the IRQ output are disabled for the above error conditions. A low level on the RESET input resets EIE to "0".

CONTROL REGISTER 3 (C3)

Control Register 3 is a 4-bit write-only register which can be programmed from the data bus whe RS = "1" and R/W = "0" and Address Control bit AC1 = "1" and AC2 = "0".

External/Internal Sync Mode Conrol (E/I Sync), C3, Bit 0 – When the E/I Sync Mode bit is high, the SSDA is in the external sync mode and the receiver synchronization logic is disabled. Synchronization can be achieved by means of the DCD input or by starting Rx CLK at the midpoint of data bit 0 of a cahracter with DCD low. Both the transmitter and receiver sections operate as parallel – serial converters in the External Sync mode. The Clear Sync bit in Control Register 1 acts as a receiver sync inhibit when high to provide a bus controllable inhibit. The Sync Code Register can serve as a transmitter fill character register and a receiver match register in this mode. A "low" on the RESET input resets the E/I Sync Mode bit placing the SSDA in the internal sync mode.

One-Sync-Character/Two-Sync-Character Mode Control (1-Sync/2-Sync), C3 Bit 1 – When the 1-Sync/2-Sync bit is set, the SSDA will synchronize on a single match between the received data and the contents of the Sync Code Register. When the 1-Sync/2-Sync bit is reset, two *successive* sync characters must be received prior to receiver synhnchronization. If the second sync character is not detected, the bit-by-bit search resumes from the first bit in the second character. See the description of the Sync Code Register for more details.

Clear CTS Status (Clear CTS), C3 Bit 2 – When a "1" is written into the Clear CTS bit, the stored status and interrupt are cleared. Subsequently, the CTS status bit reflects the state of the CTS input. The Clear CTS control bit does not affect the CTS input nor its inhibit of the transmitter section. The Clear CTS command bit is self-clearing, and writing a "0" into this bit is a nonfunctional operation.

Clear Transmit Underflow Status (CTUF), C3 Bit 3 – When a "1" is written into the CTUF status bit, the CTUF bit and its associated interrupt are reset. The CTUF command bit is self-clearing and writing a "0" into this bit is a nonfunctional operation.

SYNC CODE REGISTER

The Sync Code Register is an 8-bit register for storing the programmable sync code required for received data character synchronization in the one-sync-character and two-sync-character modes. The Sync Code Register also provides for stripping the sync/fill characters from the received data (a programmable option) as well as automatic insertion of fill characters in the transmitted data stream. The Sync Code Register is not utilized for receiver character synchronization in the external sync mode; however, it provides storage of receiver match and transmit fill characters.

The Sync Code Register can be loaded when AC2 and AC1 are a "1" and "0", respectively, and R/W = "0" and RS = "1".

The Sync Code Register may be changed after the detection of a match with the received data (the first sync code having been detected) to synchronize with a double-word sync pattern. (This sync code change must occur prior to the completion of the second character.) The sync match (SM) output can be used to interrupt the MPU system to indicate that the first eight bits have matched. The service routine would then change the sync match register to the second half of the pattern. Alternately, the one-sync-character mode can be used for sync codes for 16 or more bits by using software to check the second and subsequent bytes after reading them from the FIFO.

The detection of the sync code can be programmed to appear on the Sync Match/DTR output by writing a "1" in PC1 (C2 bit 0) and a "0" in PC2 (C2 bit 1). The Sync Match output will go high for one bit time beginning at the character interface between the sync code and the next character (see Figure 3).

PARITY FOR SYNC CHARACTER

Transmitter

Transmitter does not generate parity for the sync character except 9-bit mode.

9-bit (8-bit + parity)...8-bit sync character + parity
8-bit (7-bit + parity)...8-bit sync character (no parity)
7-bit (6-bit + parity)...7-bit sync character (no parity)

Receiver

At Synchronization

Receiver automatically strips the sync character(s) (two sync characters if '2 sync' mode is selected) which is used to establish synchronization. Parity is not checked for these sync characters.

After Synchronization Is Established

When 'strip sync' bit is selected, the sync characters (fill characters) are stripped and parity is not checked for the stripped sync (fill) characters. When "strip sync" bit is not selected (low), the sync character is assumed to be normal data and it is transferred into FIFO after parity checking. (When non-parity format is selected, parity is not checked.)

Strip Sync (C1, Bit 2)	WS0-WS2 (Data Format) (C2, Bits 3-5)	
1	X	No transfer of sync code No parity Check of sync code
0	With Parity	*Transfer data and sync codes Parity check
0	Without Parity	*Transfer data and sync codes No parity check

*Subsequent to synchronization.

It is necessary to consider parity in the selected sync character in the following cases. Data Format is (6+ parity), (7+ parity), strip sync is not selected (low), and when sync code is used as a fill character after synchronization.

The transmitter sends a sync character without parity, but the receiver checks the parity as if it is normal data. Therefore, the sync character should be chosen to match the parity check selected for the receiver in this special case. See the following section for unused bit assignment in short-word length.

RECEIVE DATA FIRST-IN FIRST-OUT REGISTER (Rx Data FIFO)

The Receive Data FIFO Register consists of three 8-bit registers which are used for buffer storage of received data. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register. The transfer from register to register occurs on E pulses. The RDA status bit will be high when data is available in the last location of the Rx Data FIFO.

In an Overrun condition, the overrunning character will be transferred into the full first stage of the FIFO register and will cause the loss of that data character. Successive overruns continue to overwrite the first register of the FIFO. This destruction of data is indicated by means of the Overrun status bit. The Overrun bit will be set when the overrun occurs and remains set until the Status Register is read, followed by a read of the Rx Data FIFO.

Unused data bits for short word lengths (including the parity bit) will appear as "0's" on the data bus when the Rx Data FIFO is read.

TRANSMIT DATA FIRST-IN FIRST-OUT REGISTER (Tx Data FIFO)

The Transmit Data FIFO Register consists of thee 8-bit registers which are used for buffer storage of data to be transmitted. Each 8-bit register has an internal status bit which monitors its full or empty condition. Data is always transferred from a full register to an adjacent empty register. The transfer is clocked by E pulses.

The TDRA status bit will be high if the Tx Data FIFO is available for data.

Unused data bits for short word lengths will be handled as "don't cares." The parity bit is not transferred over the data bus since the SSDA generates parity at transmission.

When an Underflow occurs, the Underflow character will be either the contents of the Sync Code Register or an all "1's" character. The underflow will be stored in the Status Register until cleared and will appear on the Underflow output as a pulse approximatley one Tx CLK high period wide.

STATUS REGISTER (S)

The Status Register is an 8-bit read-only register which provides the real-time status of the SSDA and the associated serial data channel. Reading the Status Register is a non-destructive process. The method of clearing status bits depends upon the function each bit represents and is discussed for each bit in the register.

Receiver Data Available (RDA), S Bit 0 – The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. The receiver data being present in the last register (#3) of the FIFO causes RDA to be high for the 1-byte transfer mode. The RDA bit being high indicates that the last two registers (#2 and #3) are full when in the 2-byte transfer mode. The second character can be read without a second status read (to determine that the character is available). An E pulse must occur between reads of the Rx Data FIFO to allow the FIFO to shift. Status must be read on a word-by-word basis if receiver data error checking is important. The RDA status bit is reset automatically when data is not available.

Transmitter Data Register Available (TDRA), S Bit 1 – The TDRA status bit indicates that data can be loaded into the Tx Data FIFO Register. The first register (#1) of the Tx Data IFFO being empty will be indicated by a high level in the TDRA status bit in the 1-byte transfer mode. The first two registers (#1 and #2) must be empty for TDRA to be high when in the 2-byte transfer mode. The Tx Data FIFO can be loaded with two bytes without an intervening status read; however, one E pulse must occur between loads. TDRA is inhibited by the Tx Reset or $\overline{\text{RESET}}$. When Tx Reset is set, the Tx Data FIFO is cleared and then released on the next E clock pulse. The Tx Data FIFO can then be loaded with up to three characters of data, even though TDRA is inhibited. This feature allows preloading data prior to the release of Tx Reset. A high level on the $\overline{\text{CTS}}$ input inhibits the TDRA status bit in either sync mode of operation (one-sync-character or two-sync-character). $\overline{\text{CTS}}$ does not affect TDRA in the external sync mode. This enables the SSDA to operate under the control of the $\overline{\text{CTS}}$ input with TDRA indicating the status of the Tx Data FIFO. The $\overline{\text{CTS}}$ input does not clear the Tx Data FIFO in any operating mode.

Data Carrier Detect ($\overline{\text{DCD}}$), S Bit 2 – A positive transition on the DCD input is stored in the SSDA until cleared by reading both Status and Rx Data FIFO. A "1" written into Rx Rs also clears the stored $\overline{\text{DCD}}$ status. The $\overline{\text{DCD}}$ status bit, when set, indicates that the $\overline{\text{DCD}}$ input has gone high. The reading of Status followed by reading of the Receive Data FIFO allows Bit 2 of subsequent Status reads to indicate the state of the $\overline{\text{DCD}}$ input until the next positive transition.

Clear-to-Send ($\overline{\text{CTS}}$), S Bit 3 — A positive transition on the $\overline{\text{CTS}}$ input is stored in the SSDA until cleared by writing a "1" into the Clear $\overline{\text{CTS}}$ control bit or the Tx Rs bit. The $\overline{\text{CTS}}$ status bit, when set, indicates that the $\overline{\text{CTS}}$ input has gone high. The Clear $\overline{\text{CTS}}$ command (a "1" into C3 Bit 2) allows Bit 3 of subsequent Status reads to indicate the state of the $\overline{\text{CTS}}$ input until the next positive transition.

Transmitter Underflow (TUF), S Bit 4 — When data is not available for the transmitter, an underflow occurs and is so indicated in the Status Register (in the Tx Sync on underflow mode). The underflow status bit is cleared by writing a "1" into the Clear Underflow (CTUF) control bit or the Tx Rs bit. TUF indicates that a sync character will be transmitted as the next character. A TUF is indicated on the output *only* when the contents of the Sync Code Register is to be transferred (transmit sync code on underflow = "1").

Receiver Overrun (Rx Ovrn), S Bit 5 — Overrun indicates data has been received when the Rx Data FIFO is full, resulting in data loss. The Rx Ovrn status bit is set when overrun occurs. The Rx Ovrn status bit is cleared by reading Status followed by reading the Rx Data FIFO or by setting the Rx Rs control bit.

Receiver Parity Error (PE), S Bit 6 — The parity error status bit indicates that parity for the character in the last register of the Rx Data FIFO did not agree with selected parity. The parity error is cleared when the character to which it pertains is read from the Rx Data FIFO or when Rx Rs occurs. The $\overline{\text{DCD}}$ input does not clear the Parity Error or Rx Data FIFO status bits.

Interrupt Request ($\overline{\text{IRQ}}$), S Bit 7 — The Interrupt Request status bit indicates when the $\overline{\text{IRQ}}$ output is in the active state ($\overline{\text{IRQ}}$ output = "0"). The $\overline{\text{IRQ}}$ status bit is subject to the same interrupt enables (RIE, TIE, and EIE) as the $\overline{\text{IRQ}}$ output. The $\overline{\text{IRQ}}$ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests.

MOTOROLA

MC6854
(1.0 MHz)
MC68A54
(1.5 MHz)
MC68B54
(2.0 MHz)

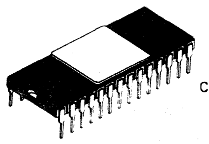
ADVANCED DATA-LINK CONTROLLER (ADLC)

The MC6854 ADLC performs the complex MPU/data communication link function for the "Advanced Data Communication Control Procedure" (ADCCP), High-Level Data-Link Control (HDLC) and Synchronous Data-Link Control (SDLC) standards. The ADLC provides key interface requirements with improved software efficiency. The ADLC is designed to provide the data communications interface for both primary and secondary stations in stand-alone, polling, and loop configurations.

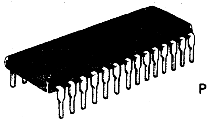
- M6800 Compatible
- Protocol Features
  - Automatic Flag Detection and Synchronization
  - Zero Insertion and Deletion
  - Extendable Address, Control and Logical Control Fields (Optional)
  - Variable Word Length Information Field — 5-, 6-, 7-, or 8-Bits
  - Automatic Frame Check Sequence Generation and Check
  - Abort Detection and Transmision
  - Idle Detection and Transmission
- Loop Mode Operation
- Loop Back Self-Test Mode
- NRZ/NRZI Modes
- Quad Data Buffers for Each Rx and Tx
- Prioritized Status Register (Optional)
- MODEM/DMA/Loop Interface

MOS
(N-CHANNEL, SILICON GATE)

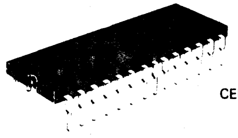
ADVANCED DATA-LINK CONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

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 MC6854, MC68A54, MC68B54 MC6854C, MC68A54C	T_A	(T_L to T_H) 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

THERMAL CHRACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip	θ_{JA}	 115 60 65	°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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

PIN DESCRIPTION

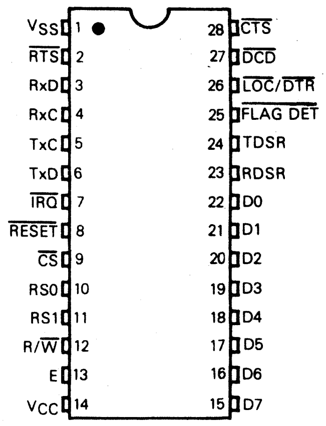
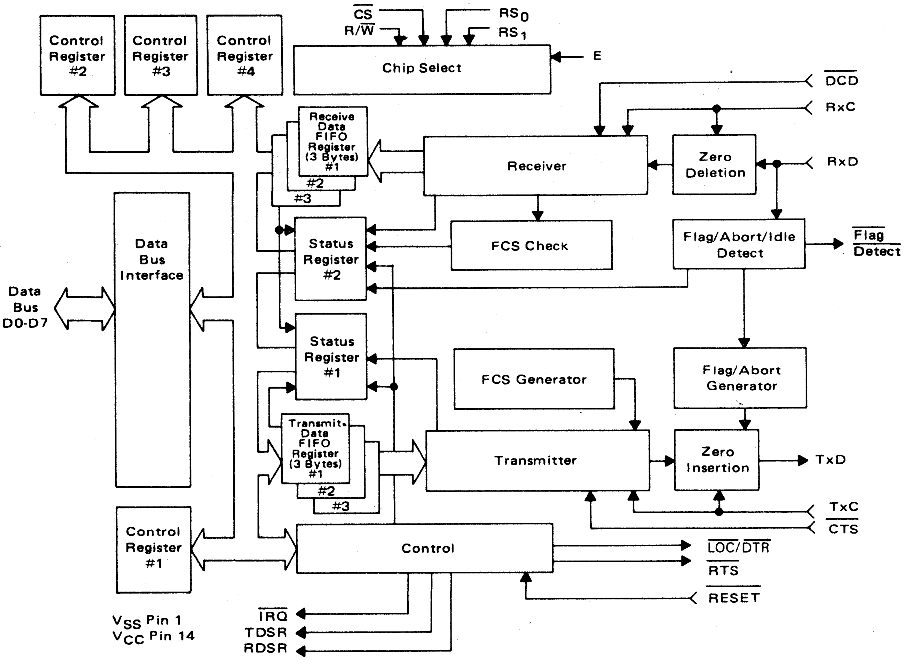


FIGURE 1 — ADLC GENERAL BLOCK DIAGRAM



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.

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

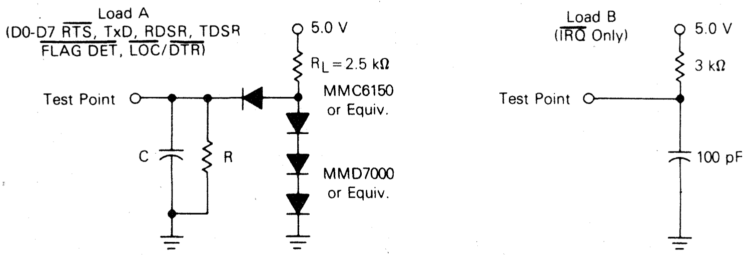
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	V_{SS} + 2.0	–	–	V
Input Low Voltage		V_{IL}	–	–	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	All Inputs Except D0-D7	I_{in}	–	1.0	2.5	μA
Three-State (Off-State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = 5.25 V)	D0-D7	I_{IZ}	–	2.0	10	μA
DC Output High Voltage (I_{Load} = −205 μA) (V_{Load} = −100 μA)	D0-D7 All Others	V_{OH}	V_{SS} + 2.4 V_{SS} + 2.4	– –	– –	V
DC Output Low Voltage (I_{Load} = 1.6 mA)		V_{OL}	–	–	V_{SS} + 0.4	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQ}$	I_{OZ}	–	1.0	10	μA
Internal Power Dissipation (measured at $T_A = T_L$)		P_{INT}	–	–	850	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7 All Other Inputs	C_{in}	– –	– --	12.5 7.5	pF
	$\overline{IRQ}$ All Others	C_{out}	– –	– –	5.0 10	pF

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

Characteristic	Symbol	MC6854		MC68A54		MC68B54		Unit
		Min	Max	Min	Max	Min	Max	
Clock Pulse Width, Low (RxC, TxC)	PW_{CL}	700	–	450	–	280	–	ns
Clock Pulse Width, High (RxC, TxC)	PW_{CH}	700	–	450	–	280	–	ns
Serial Clock Frequency (RxC, TxC)	f_{SC}	–	0.66	–	1.0	–	1.5	MHz
Receive Data Setup Time	t_{RDSU}	150	–	100	–	50	–	ns
Receive Data Hold Time	t_{RDH}	60	–	60	–	60	–	ns
Request-to-Send Delay Time	t_{RTS}	–	680	–	460	–	340	ns
Clock-to-Data Delay for Transmitter	t_{TDD}	–	300	–	250	–	200	ns
Flag Detect Delay Time	t_{FD}	–	680	–	460	–	340	ns
$\overline{DTR}$ Delay Time	t_{DTR}	–	680	–	460	–	340	ns
Loop On-Line Control Delay Time	t_{LOC}	–	680	–	460	–	340	ns
RDSR Delay Time	t_{RDSR}	–	540	–	400	–	340	ns
TDSR Delay Time	t_{TDSR}	–	540	–	400	–	340	ns
Interrupt Request Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	μs
$\overline{RESET}$ Pulse Width	t_{RESET}	1.0	–	0.65	–	0.40	–	μs
Input Rise and Fall Times (Except Enable) (0.8 V to 2.0 V)	t_r, t_f	–	1.0*	–	1.0*	–	1.0*	μs

*1.0 μs or 10% of the pulse width, whichever is smaller.

FIGURE 2 — BUS TIMING TEST LOADS



C = 130 pF for D0-D7
= 30 pF for others

R = 11.7 kΩ for D0-D7
= 24 kΩ for others

FIGURE 3 RECEIVER DATA SETUP/HOLD, FLAG DETECT AND LOOP ON-LINE CONTROL DELAY TIMING

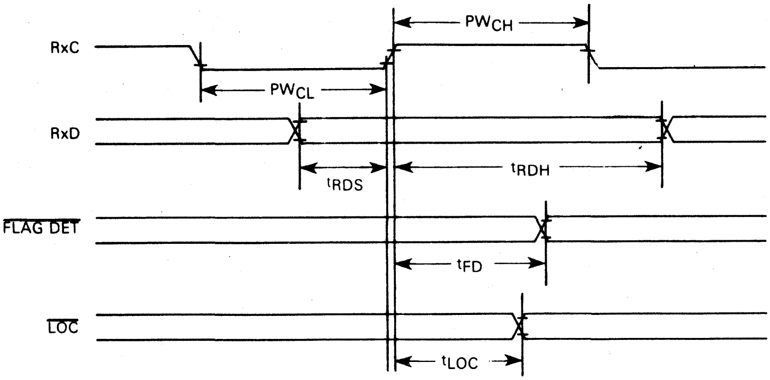


FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND REQUEST-TO-SEND DELAY TIMING

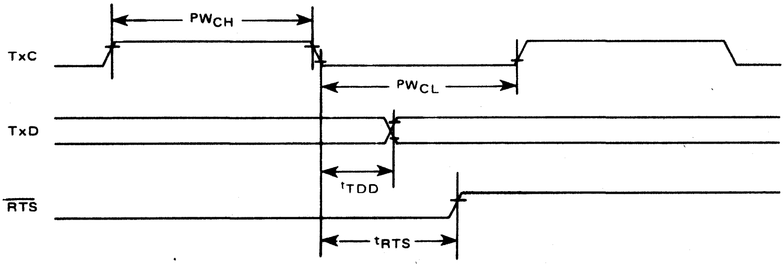
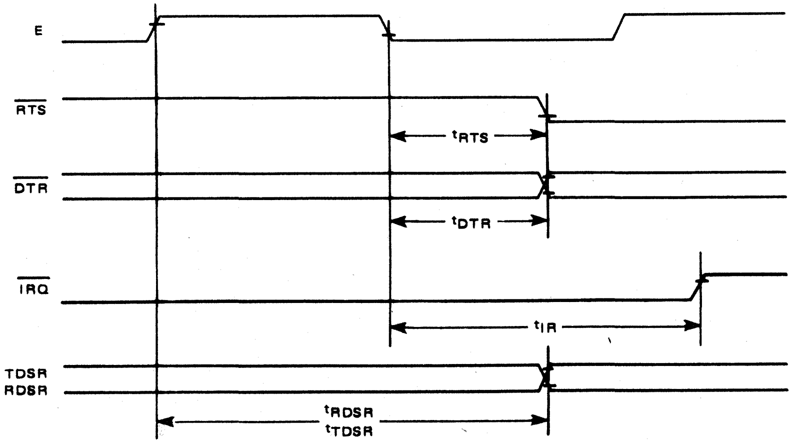


FIGURE 5 — TDSR/RDSR DELAYS, $\overline{\text{IRQ}}$ RELEASE DELAY, $\overline{\text{RTS}}$ AND $\overline{\text{DTR}}$ DELAY TIMING

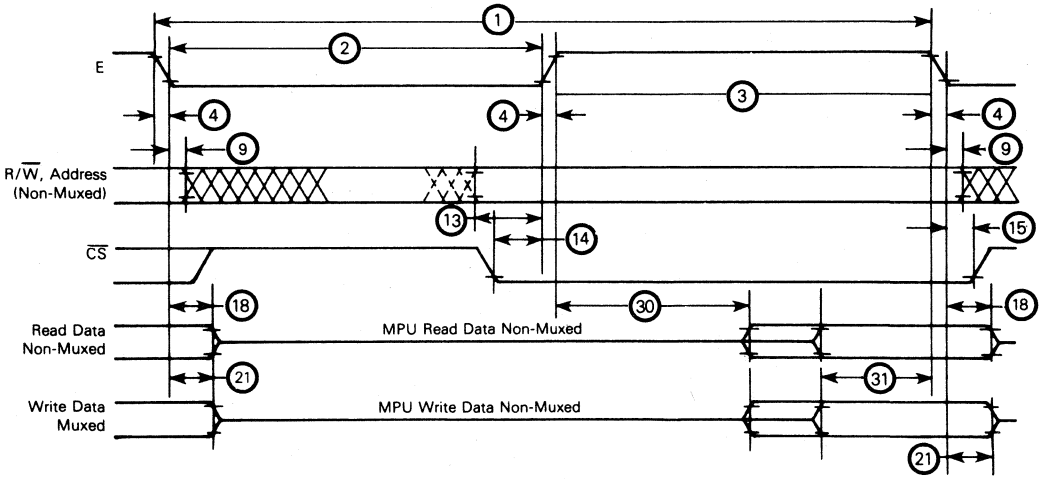


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6854		MC68A54		MC68B54		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 6 — BUS TIMING



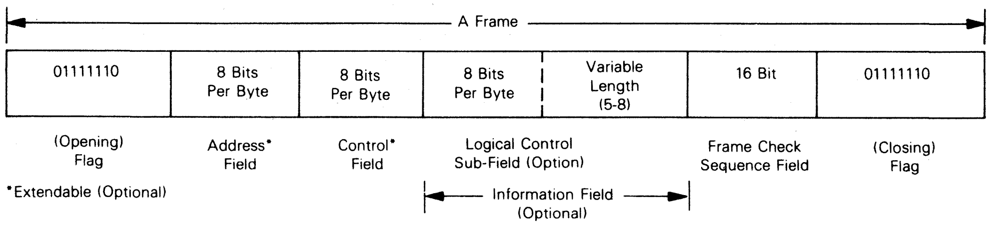
Notes:

1. Voltage levels are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless othewise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FRAME FORMAT

The ADLC transmits and receives data (information or control) in a format called a frame. All frames start with an opening flag (F) and end with a closing flag (F). Between the opening flag and closing flag, a frame contains an address field, control field, information field (optional) and frame check sequence field.

FIGURE 7 — DATA FORMAT OF A FRAME



Flag (F) — The flag is the unique binary pattern (01111110). It provides the frame boundary and a reference for the position of each field of the frame.

The ADLC transmitter generates a flag pattern internally and the opening flag and closing flags are appended to a frame automatically. Two successive frames can share one flag for a closing flag of the first frame and for the opening flag of the next frame, if the "FF"/"F" control bit in the control register is reset.

The receiver searches for a flag on a bit-by-bit basis and recognizes a flag at any time. The receiver establishes the frame synchronization with every flag. The flags mark the frame boundary and reference for each field but they are not transferred to the Rx FIFO. The detection of a flag is indicated by the Flag Detect output and by a status bit in the status register.

Order of Bit Transmission — Address, control and information field bytes are transferred between the MPU and the ADLC in parallel by means of the data bus. The bit on D0 (data bus bit 0, pin 22) is serially transmitted first, and the first serially received bit is transferred to the MPU on D0. The FCS field is transmitted and received MSB first.

Address (A) Field — The 8 bits following the opening flag are the address (A) field. The A-field can be extendable if the Auto-Address Extend Mode is selected in control register #3. In the Address Extend Mode, the first bit (bit 0) in every address octet becomes the extend control bit. When the bit is "0", the ADLC assumes another address octet will follow, and when the bit is "1", the address extension is terminated. A "null" address (all "0's") does not extend. In the receiver, the Address Present status bit distinguishes the address field from other fields. When an address byte is available to be read in the receive FIFO register, the Address Present status bit is set and causes an interrupt (if enabled). The Address Present bit is set for every address octet when the Address Extend Mode is used.

Control (C) Field — The 8 bits following the address field is the control (link control) field. When the Extended Control Field bit in control register #3 is selected, the C-field is extended to 16 bits.

Information (I) Field — The I-field follows the C-field and precedes the FCS field. The I-field contains "data" to be transferred but is not always necessarily contained in every frame. The word length of the I-field can be selected from 5 to 8 bits per byte by control bits in control register #4. The I-field will continue until it is terminated by the FCS and closing flag. The receiver has the capability to handle a "partial" last byte. The last information byte can be any word length between 1 and 8 bits. If the last byte in the I-field is less than the selected word length, the receiver will right justify the received bits, fill the remaining bits of the receiver shift register with zeros, and transfer a full byte to the Rx FIFO. Regardless of selected byte length, the ADLC will transfer 8 bits of data to the data bus. Unused bits for word lengths of 5, 6, and 7 will be zeroed.

Logical Control (LC) Field — When the Logical Control Field Select bit, in control register #3, is selected the ADLC separates the I-field into two sub-fields. The first sub-field is the Logical Control field and the following sub-field is the "data" portion of the I-field. The logical control field is 8 bits and follows the C-field, which is extendable by octets, if it is selected. The last bit (bit 7) is the extend control bit, and if it is a "1", the LC-field is extended one octet.

NOTE

Hereafter the word "Information field" or "I-field" is used as the data portion of the information field, and excludes the logical control field. This is done in order to keep the consistency of the meaning of "Information field" as specified in SDLC, HDLC, and ADCCP standards.

Frame Check Sequence (FCS) Field – The 16 bits preceding the closing flag is the FCS field. The FCS is the "cyclic redundancy check character (CRCC)." The polynomial $x^{16} + x^{12} + x^5 + 1$ is used both for the transmitter and receiver. Both the transmitter and receiver polynomial registers are initialized to all "1's" prior to calculation of the FCS. The transmitter calculates the FCS on all bits of the address, control, logical control (if selected), and information fields, and transmits the complement of the resulting remainder as FCS. The receiver performs the similar computation on all bits of the address, control, logical control (if selected), information, and received FCS fields and compares the result to F0B8 (Hexadecimal). When the result matches F0B8, the Frame Valid status bit is set in the status register. If the result does not match, the Error status bit is set. The FCS generation, transmission, and checking are performed automatically by the ADLC transmitter and receiver. The FCS field is not transferred to the Rx FIFO.

Invalid Frame – Any valid frames should have at least the A-field, C-field, and FCS field between the opening flag and the closing flag. When invalid frames are received, the ADLC handles them as follows:

1. A short frame which has less than 25 bits between flags – the ADLC ignores the short frame and its reception is not reported to the MPU.
2. A frame less than 32 bits between the flags, or a frame 32 bits or more with an extended A-field or C-field that is not completed. – This frame is transferred into the Rx FIFO. The FCS/IF Error status bit indicates the reception of the invalid frame at the end of the frame.
3. Aborted Frame – The frame which is aborted by receiving an abort or DCD failure is also an invalid frame. Refer to "Abort" and "DCD status bit".

Zero Insertion and Zero Deletion – The Zero insertion and deletion, which allows the content of the frame to be transparent, are performed by the ADLC automatically. A binary 0 is inserted by the transmitter after any succession of five "1's" within a frame (A, C, LC, I, and FCS field). The receiver deletes a binary 0 that follows successive five continuous "1's" within a frame.

Abort – The function of prematurely terminating a data link is called "abort." The transmitter aborts a frame by sending at least eight consecutive "1's" immediately after the Tx Abort control bit in control register #4 is set to a "1". (Tx FIFO is also cleared by the Tx Abort control bit at the same time.) The abort can be extended up to (at least) 16 consecutive "1's", if the Abort Extend control bit in the control register #4 is set when an abort is sent. This feature is useful to force mark idle transmission. Reception of seven or more consecutive "1's" is interpreted as an abort by the receiver. The receiver responds to a received abort as follows:

1. An abort in an "out of frame" condition – an abort during the idle or time fill has no meaning. The abort reception is indicated in the status register as long as the abort condition continues; but neither an interrupt nor a stored condition occurs. The abort indication disappears after 15 or more consecutive "1's" are received (Received Idle status is set.)
2. An abort "in frame" after less than 26 bits are received after an opening flag – under this condition, any field of the aborted frame has not transferred to the MPU yet. The ADLC clears the aborted frame data in the FIFO and clears flag synchronization. Neither an interrupt nor a stored status occurs. The status indication is the same as (1) above.
3. An abort "in frame" after 26 bits or more are received after an opening flag – under this condition, some fields of the aborted frame might have been transferred onto the data bus. The abort status is stored in the receiver status register and the data of the aborted frame in the ADLC is cleared. The synchronization is also cleared.

Idle and Time Fill – When the transmitter is in an "out of frame" condition (the transmitter is not transmitting a frame), it is in an idle state. Either a series of contiguous flags (time fill) or a mark idle (consecutive "1's" on a bit-by-bit basis) is selected for the transmission in an idle state by the Flag/Mark Idle control bit. When the receiver receives 15 or more consecutive "1's", the Receive Idle status bit is set and causes an interrupt. The flags and mark idle are not transferred to the Rx FIFO.

OPERATION

INITIALIZATION

During a power-on sequence, the ADLC is reset via the $\overline{\text{RESET}}$ input and internally latched in a reset condition to prevent erroneous output transitions. The four control registers must be programmed prior to the release of the reset condition. The release of the reset condition is performed via software by writing a "0" into the Rx RS control bit (receiver) and/or Tx RS control bit (transmitter). The release of the reset condition must be done after the $\overline{\text{RESET}}$ input has gone high.

At any time during operation, writing a "1" into the Rx RS control bit or Tx RS control bit causes the reset condition of the receiver or the transmitter.

TRANSMITTER OPERATION

The Tx FIFO register cannot be pre-loaded when the transmitter is in a reset state. After the reset release, the Flag/Mark Idle control bit selects either the mark idle state (inactive idle) or the Flag "time fill" (active idle) state. This active or inactive mark idle state will continue until data is loaded into the Tx FIFO.

The availability of the Tx FIFO is indicated by the TDRA status bit under the control of the 2-Byte/1-Byte control bit. TDRA status is inhibited by the Tx RS bit or $\overline{\text{CTS}}$ input being high. When the 1-Byte mode is selected, one byte of the FIFO is available for data transfer when TDRA goes high. When the 2-Byte mode is selected, two successive bytes can be transferred when TDRA goes high.

The first byte (Address field) should be written into the Tx FIFO at the "Frame Continue" address. Then the transmission of a frame automatically starts. If the transmitter is in a mark idle state, the transfer of an address causes an opening flag within two or three transmitter clock cycles. If the transmitter has been in a time fill state, the current time fill flag being transmitted is assumed as an opening flag and the address field will follow it.

FIGURE 8a — ADLC TRANSMITTER STATE DIAGRAM
(C_ib_i refers to control register bit)

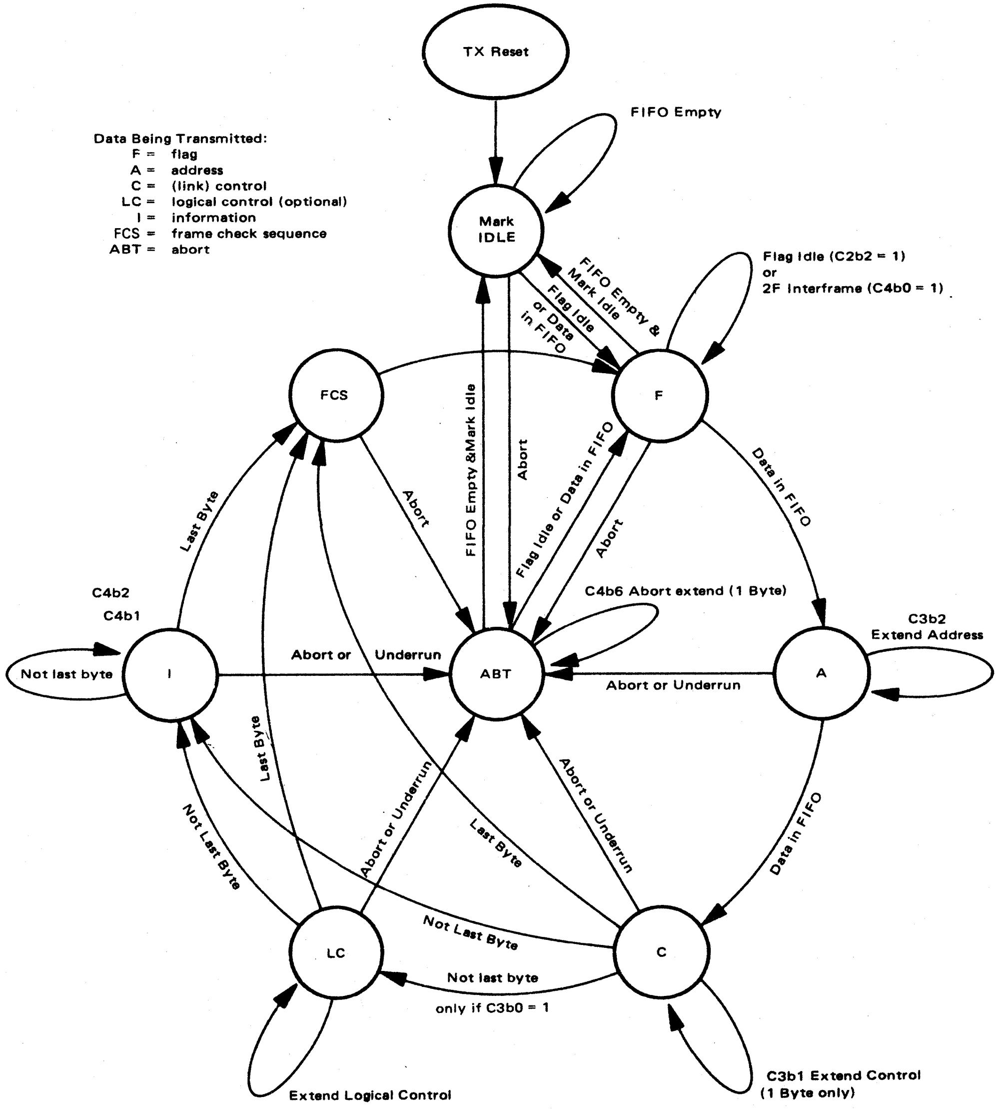
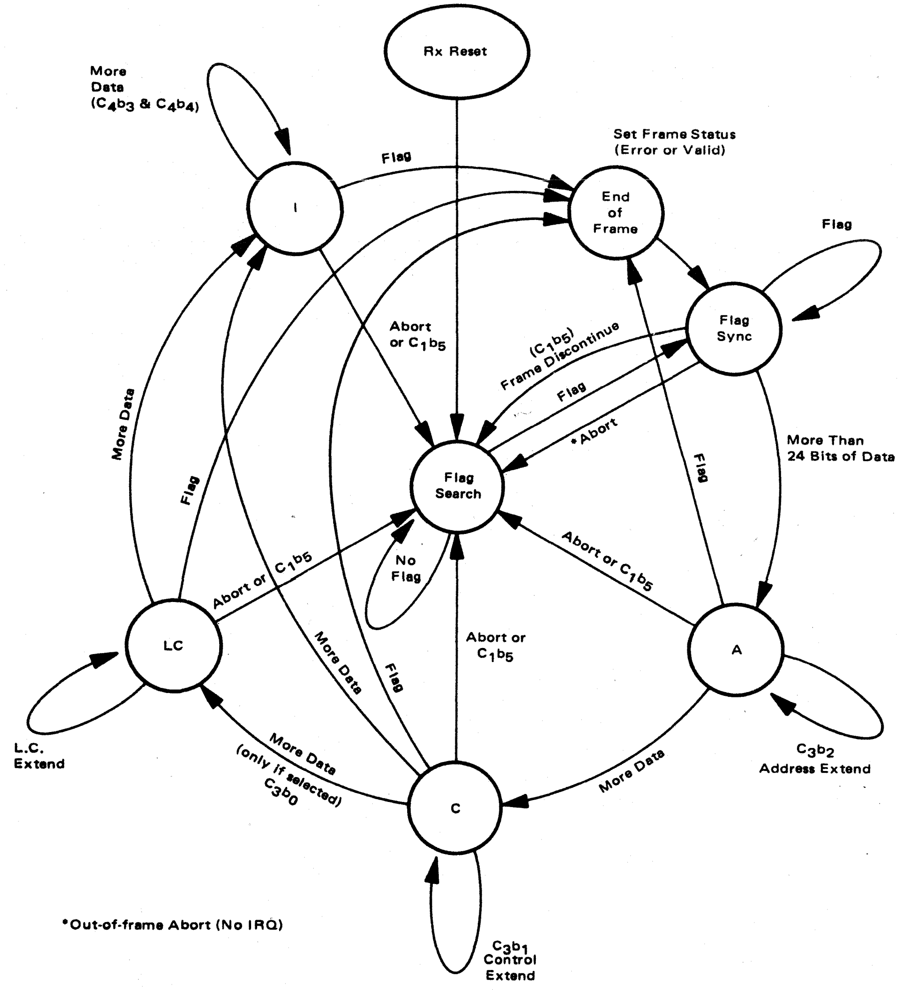


FIGURE 8b — ADLC RECEIVER STATE DIAGRAM



A frame continues as long as data is written into the Tx FIFO at the "Frame Continue" address. The ADLC internally keeps track of the field sequence in the frame. The frame format is described in the "FRAME FORMAT" section.

The frame is terminated by one of two methods. The most efficient way to terminate the frames from a software standpoint is to write the last data character into the Transmit FIFO "Frame Terminate" address (RS1, RS0 = 11) rather than the Transmit FIFO "Frame Continue" address (RS1, RS0 = 10). An alternate method is to follow the last write of data in the Tx FIFO "Frame Continue" address with the setting of the Transmit Last Data control bit. Either method causes the last character to be transmitted and the FCS field to automatically be appended along with a closing flag. Data for a new frame can be loaded into the Tx FIFO immediately after the old frame data, if TDRA is high. The closing Flag can serve as the opening Flag of the next frame or separate opening and closing Flags may be transmitted. If a new frame is not ready to be transmitted, the ADLC will automatically transmit the Active (Flag) or Inactive (Mark) Idle condition.

If the Tx FIFO becomes empty at any time during frame transmission (the FIFO has no data to transfer into transmitter shift register during transmission of the last half of the next to last bit of a word), an underrun will occur and the transmitter automatically terminates the frame by transmitting an abort. The underrun state is indicated by the Tx Underrun status bit.

Any time the Tx ABORT Control bit is set, the transmitter immediately aborts the frame (transmits at least 8 consecutive "1's") and clears the Tx FIFO. If the Abort Extend Control bit is set at the time, an idle (at least 16 consecutive "1's") is transmitted. An abort or idle in an "out of frame" condition can be useful to gain 8 or 16 bits of delay. (For an example, see "Programming Considerations.")

The $\overline{CTS}$ (Clear-to-Send) input and $\overline{RTS}$ (Request-to-Send) output are provided for a MODEM or other hardware interface.

The TDRA/FC status bit (when selected to be Frame Complete Status) can cause an interrupt upon frame completion (i.e., a flag or abort completion).

Details regarding the inputs and outputs, status bits, control bits, and FIFO operation are described in their respective sections.

RECEIVER OPERATION

Data and a pre-synchronized clock are provided to the ADLC receiver section by means of the Receive Data (RxD) and Receive Clock (RxC) inputs. The data is a continuous stream of binary bits with the characteristic that a maximum of five "1's" can occur in succession unless Abort, Flag, or Idling condition occurs. The receiver continuously (on a bit-by-bit basis) searches for Flags and Aborts.

When a flag is detected, the receiver establishes frame synchronization to the flag timing. If a series of flags is received, the receiver resynchronizes to each flag.

If the frame is terminated before the internal buffer time expires (the frame data is less than 25 bits after an opening flag), the frame is simply ignored. Noise on the data input (RxD) during time fill can cause this kind of invalid frame.

The received serial data enters a 32-bit shift register (clocked by RxC) before it is transferred into the Rx Data FIFO. Synchronization is established when a Flag is detected in the first eight locations of the shift register. Once synchronization has been achieved, data is clocked through to the last byte location of the shift register where it is transferred byte-per-byte into the Rx Data FIFO. The Rx Data FIFO is clocked by E to cause received data to move through the FIFO to the last empty register location. The Receiver Data Available status bit (RDA) indicates when data is present in the last register (Register #3) for the 1-Byte Transfer Mode. The 2-Byte Transfer Mode causes the RDA status bit to indicate data is available when the last two FIFO register locations (Registers #2 and #3) are full. If the data character present in the FIFO is an address octet, the status register will exhibit an Address Present status condition. Data being available in the Rx Data FIFO causes an interrupt to be initiated (assuming the receiver interrupt is enabled, RIE="1"). The MPU will read the ADLC Status Register as a result of the interrupt or in its turn in a polling sequence. RDA or Address Present will indicate that receiver data is available and the MPU should subsequently read the Rx Data FIFO register. The interrupt and status bit will then be reset automatically. If more than one character had been received and was resident in the Rx Data FIFO, subsequent E clocks will cause the FIFO to update and the RDA status bit and interrupt will again be SET. In the 2-Byte Transfer Mode both data bytes may be read on consecutive E cycles. Address Present provides for 1 byte transfers only.

The sequence of each field in the received frame is automatically handled by the ADLC. The frame format is described in the "FRAME FORMAT" section.

When a closing flag is received, the frame is terminated. The 16 bits preceding the closing flag are regarded as the FCS and are not transferred to the MPU. Whatever data is present in the most-significant byte portion of the receiver buffer register it is right justified and transferred to the Rx FIFO. The frame boundary pointer, which is explained in the "Rx FIFO REGISTER" section, is set simultaneously in the Rx FIFO. The frame boundary pointer sets the Frame Valid status bit (when the frame was completed with no error) or the FCS/IF Error Status bit (when the frame was completed with error) when the last byte of the frame appears at the last location of the Rx FIFO. As long as the Frame Valid or FCS/IF Error status bit is set, the data transfer from the second location of the Rx FIFO to the last location of the Rx FIFO is inhibited.

Any time the Frame Discontinue control bit is set, the ADLC discards the current frame data in the ADLC without dropping flag synchronization. This feature can be used to ignore a frame which is addressed to another station.

The reception of an abort or idle is explained in the "FRAME FORMAT" section. The details regarding the inputs, outputs, status bits, control bits, and Rx FIFO operation are described in their respective sections.

LOOP MODE OPERATION

The ADLC in the loop mode, not only performs the transmission and receiving of data frames in the manner previously described, but also has additional features for gaining and relinquishing loop control. In Figure 9a, a configuration is shown which depicts loop mode operation. The system configuration shows a primary station and several secondary stations. The loop is always under control of the primary station. When the primary wants to receive data, it transmits a Poll sequence and allows frame transmission to secondary stations on the loop. Each secondary is in series and adds one bit of delay to the loop. Secondary A in the figure receives data from the primary via its Rx Data Input, delays the data 1 bit, and transmits it to secondary B via its Tx Data Output. Secondaries B, C, and D operate in a similar manner. Therefore, data passes through each secondary and is received back by the primary controller.

Certain protocol rules must be followed in the manner by which the secondary station places itself on-loop (connects its transmitter output to the loop), goes active on the loop (starts transmitting its own station's data on the loop), and goes off the loop (disconnects its transmitter output). Otherwise loop data to other stations down loop would be interfered. The data stream always flows the same way and the order in which secondary terminals are serviced is determined by the hardware configuration. The primary controller times the delay through the loop. Should it exceed n+1 bit times, where n is the number of secondary terminals on the loop, it will indicate a loop failure. Control is transferred to a secondary by transmitting a "Go Ahead" signal following the closing Flag of a polling frame (request for a response from the secondary) from the primary station. The "Go Ahead" from the primary is a "0" and seven "1's" followed by mark

FIGURE 9a — TYPICAL LOOP CONFIGURATION

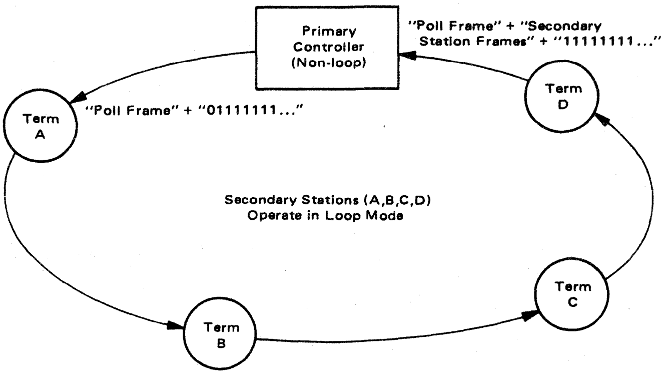
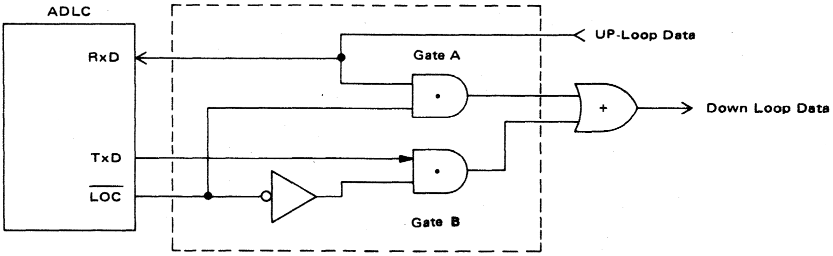


FIGURE 9b — EXAMPLE OF EXTERNAL LOOP LOGIC



idling. The primary can abort its response request by interrupting its idle with flags. The secondary should immediately stop transmission and return control back to the primary. When the secondary completes its frame, a closing flag is transmitted followed by all "1's". The primary detects the final 01111111...("Go Ahead" to the primary) and control is given back to the primary. Note that, if a down-loop secondary (e.g., station D) needs to insert information following an up-loop station (e.g., station A), the go ahead to station D is the last "0" of the closing flag from station A followed by "1's".

The ADLC in the primary station should operate in a non-loop full-duplex mode. The ADLC in the secondaries should operate in a loop mode, monitoring up-loop data on its receiver data input. The ADLC can recognize the necessary sequences in the data stream to automatically go on/off the loop and to insert its own station data. The procedure is the following and is summarized in Table 1.

(1) Go On-Loop — When the ADLC powers up, the terminal station will be off line. The first task is to become an active terminal on the loop. The ADLC must be connected to a Loop Link via an external switch as shown in Figure 9a. After a hardware reset, the ADLC $\overline{\text{LOC}}/\overline{\text{DTR}}$ Output will be in the high state and the up-loop receive data repeated through gate A to the down Loop stations. Any Up-Loop transmission will be received by the ADLC. The Loop Mode/Non-Loop Mode Control bit (bit 5 in Control Register 3) must be set to place the ADLC in the Loop Mode. The ADLC now monitors its Rx Data input for a string of seven consecutive "1's" which will allow a station to go on line. The Loop operation may be monitored by use of the Loop Status bit in Status Register 1. After power up and reset, this bit is a zero. When seven consecutive "1's" are received by the ADLC the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output will go to a low level, disabling gate A (refer to Figure 9b), enabling gate B and connecting the ADLC Tx Data output to the down Loop stations. The up Loop data is now repeated to the down Loop stations via the ADLC. A 1-bit delay is inserted in the data (in NRZI mode, there will be a 2-bit delay) as it circulates through the ADLC. The ADLC is now on-line and the Loop Status bit in Status Register 1 will be at a one.

(2) Go Active after Poll — The receiver section will monitor the up-link data for a general or addressed poll command and the Tx FIFO should be loaded with data so that when the go ahead sequence of a zero followed by seven "1's" (01111111---) is detected, transmission can be initiated immediately. When the polling frame is detected, the Go-Active-On-Poll control bit must be set (bit 6 in Control

TABLE 1 — SUMMARY OF LOOP MODE OPERATION

STATE	RX SECTION	TX SECTION	LOOP STATUS BIT
OFF-LOOP	Rx section receives data from loop and searches for 7 "1's" (when On-Loop Control bit set) to go ON-LOOP.	Inactive 1) NRZ MODE. Tx data output is maintained "high" (mark). 2) NRZI MODE. Tx data output reflects the Rx data input state delayed by one bit time. (Not normally connected to loop.) The NRZI data is internally decoded to provide error-free transitions to On-Loop mode.	"0"
ON-LOOP	1) When Go-Active on poll bit is set, Rx section searches for 01111111 pattern (the EOP or 'Go Ahead') to become the active terminal on the loop. 2) When On-Loop control bit is reset, Rx section searches for 8 "1's" to go OFF-Loop.	Inactive 1) NRZ MODE. Tx data output reflects Rx data input state delayed one bit time. 2) NRZI MODE. Tx data output reflects Rx data input state delayed 2 bit times.	"1"
ACTIVE	Rx section searches for flag (an interrupt from the loop controller) at Rx data input. Received flag causes $\overline{\text{FD}}$ output to go low. IRO is generated if $\overline{\text{RIE}}$ and FDSE control bits are set.	Tx data originates within ADLC until Go Active on Poll bit is reset and a flag or Abort is completed. Then returns to ON-Loop state.	"0"

Register 3). A maximum of seven bit times are available to set this control bit after the closing flag of the poll. When the Go-Ahead is detected by the receiver, the ADLC will automatically change the seventh one to a zero so that the repeated sequence out gate B in Figure 9b is now an opening flag sequence (011111110). Transmission now continues from the Tx FIFO with data (address, control, etc.) as previously described. When the ADLC has gone active-on-poll, the Loop Status bit in Status Register 1 will go to a zero. The receiver searches for a flag, which indicates that the primary station is interrupting the current operation.

(3) Go Inactive when On-Loop — The Go-Active-On-Poll control bit may be $\overline{\text{RESET}}$ at any time during transmission. When the frame is complete (the closing Flag or abort is transmitted), the Loop is automatically released and the station reverts back to being just a 1-bit delay in the Loop, repeating up-link data. If the Go-Active-On-Poll control bit is not reset by software and the final frame is transmitted (Flag/Mark Idle bit = 0), then the transmitter will mark idle and will not release the loop to up-loop data. A Tx Abort command would have to be used in this case in order to go inactive when on the loop. Also, if the Tx FIFO was not preloaded with data (address, control, etc.) prior to changing the "Go Ahead Character" to a Flag, the ADLC will either transmit flags (active idle character) until data is loaded (when Flag/Mark Idle Control bit is high) or will go into an underrun condition and transmit an Abort (when Flag/Mark Idle control bit is low). When an abort is transmitted, the Go-Active-on-Poll control bit is reset automatically and the ADLC reverts to its repeating mode, (TxD = delayed RxD). When the ADLC transmitter lets go of the loop, the Loop Status bit will return to a "1", indicating normal on-loop retransmission of up-loop data.

(4) Go Off-Loop — The ADLC can drop off the Loop (go off-line) similar to the way it went on-line. When the Loop On-Line control bit is reset the ADLC receiver section looks for eight successive "1's" before allowing the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output to return high (the inactive state). Gate A in Figure 9b will be enabled and gate B disabled allowing the loop to maintain continuity without disturbance. The Loop Status bit will show an off-line condition (logical zero).

SIGNAL DESCRIPTIONS

All inputs of ADLC are high-impedance and TTL-compatible level inputs. All outputs of the ADLC are compatible with standard TTL. Interrupt Request ($\overline{\text{IRQ}}$), however, is an open-drain output (no internal pullup).

INTERFACE FOR MPU

Bidirectional Data Bus (D0-D7) — These data bus I/O ports allow the data transfer between ADLC and system bus. The data bus drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an ADLC read operation.

Enable Clock (E) — E activates the address inputs ($\overline{\text{CS}}$, RS0, and RS1) and R/$\overline{\text{W}}$ input and enables the data transfer on the data bus. E also moves data through the Tx FIFO and Rx FIFO. E should be a free-running clock such as the MC6800 MPU system clock.

Chip Select ($\overline{\text{CS}}$) — An ADLC read or write operation is enabled only when the $\overline{\text{CS}}$ input is low and the E clock input is high. (E•$\overline{\text{CS}}$).

Register Selects (RS0, RS1) – When the Register Select inputs are enabled by (E•$\overline{\text{CS}}$), they select internal registers in conjunction with the Read/Write input and Address Control bit (control register 1, bit 0). Register addressing is defined in Table 2.

Read/Write Control Line (R/$\overline{\text{W}}$) – The R/$\overline{\text{W}}$ input controls the direction of data flow on the data bus when it is enabled by (E•$\overline{\text{CS}}$). When R/$\overline{\text{W}}$ is high, the I/O Buffer acts as an output driver and as an input buffer when low. It also selects the Read Only and Write Only registers within the ADLC.

Reset Input ($\overline{\text{RESET}}$) – The $\overline{\text{RESET}}$ input provides a means of resetting the ADLC from a hardware source. In the "low state," the $\overline{\text{RESET}}$ input causes the following:

- Rx Reset and Tx Reset are SET causing both the Receiver and Transmitter sections to be held in a reset condition.
- Resets the following control bits: Transmit Abort, $\overline{\text{RTS}}$, Loop Mode, and Loop On-Line/$\overline{\text{DTR}}$.
- Clears all stored status condition of the status registers.
- Outputs: $\overline{\text{RTS}}$ and $\overline{\text{LOC}}/\overline{\text{DTR}}$ go high. TxD goes to the mark state ("1's" are transmitted).

When RESET returns "high" (the inactive state) the transmitter and receiver sections will remain in the reset state until Tx Reset and Rx Reset are cleared via the data bus under software control. The Control Register bits affected by $\overline{\text{RESET}}$ cannot be changed when $\overline{\text{RESET}}$ is "low."

Interrupt Request Output ($\overline{\text{IRQ}}$) – $\overline{\text{IRQ}}$ will be low if an interrupt situation exists and the appropriate interrupt enable has been set. The interrupt remains as long as the cause for the interrupt is present and the enable is set. $\overline{\text{IRQ}}$ will be low as long as the $\overline{\text{IRQ}}$ status bit is set and is high if the $\overline{\text{IRQ}}$ status bit is not set.

CLOCK AND DATA OF TRANSMITTER AND RECEIVER

Transmitter Clock Input (TxC) – The transmitter shifts data on the negative transition of the TxC clock input. When the Loop Mode or Test Mode is selected, TxC should be the same frequency and phase as the RxC clock. The data rate of the transmitter should not exceed the E frequency.

Receiver Clock Input (RxC) – The receiver samples the data on the positive transition of the RxC clock. RxC should be synchronized with receive data externally.

Transmit Data Output (TxD) – The serial data from the transmitter is coded in NRZ or NRZI (Zero Complement) data format.

Receiver Data Input (RxD) – The serial data to be received by the ADLC can be coded in NRZ or NRZI (Zero Complement) data format. The data rate of the receiver should not exceed the E frequency. If a partial byte reception is possible at the end of a frame, the maximum data rate of the receiver is indicated by the following relationship:

$$f_{RxC} \leq \frac{1}{2t_E + 300\text{ ns}}$$

where t_E is the period of E.

PERIPHERAL/MODEM CONTROL

Request-to-Send Output ($\overline{\text{RTS}}$) – The Request-to-Send output is controlled by the Request-to-Send control bit in conjunction with the state of the transmitter section. When the $\overline{\text{RTS}}$ bit goes high, the $\overline{\text{RTS}}$ output is forced low. When the $\overline{\text{RTS}}$ bit returns low, the $\overline{\text{RTS}}$ output remains low until the end of the frame and there is no further data in the Tx FIFO for a new frame. The positive transition of $\overline{\text{RTS}}$ occurs after the completion of a Flag, an Abort, or when the RTS control bit is reset during a mark idling state. When the $\overline{\text{RESET}}$ input is low, the $\overline{\text{RTS}}$ output goes high.

Clear-to-Send Input ($\overline{\text{CTS}}$) – The $\overline{\text{CTS}}$ input provides a real-time inhibit to the TDRA status bit and its associated interrupt. The positive transition of $\overline{\text{CTS}}$ is stored within the ADLC to ensure its occurrence will be acknowledged by the system. The stored $\overline{\text{CTS}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by writing a "1" in the Clear Tx Status bit or in the Transmitter Reset bit.

Data-Carrier-Detect Inupt ($\overline{\text{DCD}}$) – The $\overline{\text{DCD}}$ input provides a real-time inhibit to the receiver section. A high level on the $\overline{\text{DCD}}$ input resets and inhibits the receiver register, but data in the Rx FIFO from a previous frame is not disturbed. The positive transition of $\overline{\text{DCD}}$ is stored within the ADLC to ensure that its occurrence will be acknowledged by the system. The stored $\overline{\text{DCD}}$ information and its associated $\overline{\text{IRQ}}$ (if enabled) are cleared by means of the Clear Rx Status Control bit or by the Rx Reset bit.

Loop On-Line Control/Data Terminal Ready Output ($\overline{\text{LOC}}/\overline{\text{DTR}}$) – The $\overline{\text{LOC}}/\overline{\text{DTR}}$ output serves as a $\overline{\text{DTR}}$ output in the non-loop mode or as a Loop Control output in the loop mode. When $\overline{\text{LOC}}/\overline{\text{DTR}}$ output performs the $\overline{\text{DTR}}$ function, it is turned on and off by means of the $\overline{\text{LOC}}/\overline{\text{DTR}}$ control bit. When the $\overline{\text{LOC}}/\overline{\text{DTR}}$ control bit is high the $\overline{\text{DTR}}$ output will be low. In the loop mode the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output provides the means of controlling the external loop interface hardware to go On-line or Off-line. When the $\overline{\text{LOC}}/\overline{\text{DTR}}$ control bit is SET and the loop has "idled" for 7 bit times or more (RxD = 01111111...), the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output will go low (on-line). The $\overline{\text{RESET}}$ input being low will cause the $\overline{\text{LOC}}/\overline{\text{DTR}}$ output to be high.

Flag Detect Output ($\overline{\text{FD}}$) – An output to indicate the reception of a flag and initiate an external time-out counter for the loop mode operation. The $\overline{\text{FD}}$ output goes low for 1 bit time beginning at the last bit of the flag character, as sampled by the receiver clock (RxC).

DMA INTERFACE

Receiver Data Service Request Output (RDSR) – The RDSR Output is provided primarily for use in DMA Mode operation and indicates (when high) that the Rx FIFO requests service (RSDR output reflects the RDA status bit regardless of the state of the RDSR mode control bit in CR1). If the prioritized Status Mode is selected, RDSR will be inhibited when any other receiver status conditions are present. RDSR goes low when the Rx FIFO is read.

Transmitter Data Service Request Output (TDSR) — The TDSR Output is provided for DMA mode operation and indicates (when high) that the Tx FIFO request service regardless of the state of the TDSR Mode Control bit in CR1. TDSR goes low when the Tx FIFO is loaded. TDSR is inhibited by: the Tx RS control bit being SET, $\overline{\text{RESET}}$ being low, or CTS being high. If the prioritized status mode is used, Tx Underrun also inhibits TDSR. TDSR reflects the TDRA status bit except in the FC mode. In the FC mode the TDSR line is inhibited.

ADLC REGISTERS

Eight registers in the ADLC can be accessed by means of the MPU data and address buses. The registers are defined as read-only or write-only according to the direction of information flow. The addresses of these registers are defined in Table 2. The transitter FIFO register can be accessed by two different addresses, the "Frame Terminate" address and the "Frame Continue" address. (The function of these addresses are discussed in the FIFO section.)

TABLE 2 — REGISTER ADDRESSING

Register Selected	R/W	RS1	RS0	Address Control Bit (C_1b_0)
Write Control Register #1	0	0	0	X
Write Control Register #2	0	0	1	0
Write Control Register #3	0	0	1	1
Write Transmit FIFO (Frame Continue)	0	1	0	X
Write Transmit FIFO (Frame Terminate)	0	1	1	0
Write Control Register #4	0	1	1	1
Read Status Register #1	1	0	0	X
Read Status Register #2	1	0	1	X
Read Receiver FIFO	1	1	X	X

RECEIVER DATA FIRST-IN FIRST-OUT REGISTER

Rx FIFO — The Rx FIFO consists of three 8-bit registers which are used for the buffer storage of received data. Data bytes are always transferred from a full register to an adjacent empty register; and both phases of the E input clock are used for the data transfer. Each register has pointer bits which point the frame boundary. When these pointers appear at the last FIFO location, they update the Address Present, Frame Valid, or FCS/IF Error status bits.

The RDA status bit indicates the state of the Rx FIFO. When RDA status bit is "1", the Rx FIFO is ready to be read. The RDA status is controlled by the 2-Byte/1-Byte control bit. When overrun occurs, the data in the first byte of the Rx FIFO are not longer valid.

Both the Rx Reset bit and $\overline{\text{RESET}}$ input clear the Rx FIFO. Abort ("in Frame") and a high level on the $\overline{\text{DCD}}$ input also clears the Rx FIFO, but the last bytes of the previous frame, which are separated by the frame boundary pointer, are not disturbed.

TRANSMITTER DATA FIRST-IN FIRST-OUT REGISTER

Tx FIFO — The Tx FIFO consists of three 8-bit registers which are used for buffer storage of data to be transmitted. Data is always transferred from a full register to an empty adjacent register; the transfer occurs on both phases of the E input clock. The Tx FIFO can be addressed by two different register addresses, the "Frame Continue" address and the "Frame Terminate" address. Each register has pointer bits which point to the frame boundary. When a data byte is written at the "Frame Continue" address, the pointer of the first FIFO register is set. When a data byte is written at the "Frame Terminate" address, the pointer of the first FIFO register is reset. Rx RS control bit or Tx Abort control bit resets all pointers. The pointer will shift through the FIFO. When a positive transition is detected at the third location of FIFO, the transmitter initiates a frame with an open flag. When the negative transition is detected at the third location of FIFO, the transmitter closes a frame, appending the FCS and closing Flag to the last byte.

The Tx last control bit can be used instead of using the "Frame Terminate" address. When the Tx last control bit is set with a "1", the logic searches the last byte location in the FIFO and resets the pointer in the FIFO register.

The status of Tx FIFO is indicated by the TDRA status bit. When TDRA is "1", the Tx FIFO is available for loading data. The TDRA status is controlled by the 2-Byte/1-Byte control bit. The Tx FIFO is reset by both Tx Reset and $\overline{\text{RESET}}$ input. During this reset condition or when $\overline{\text{CTS}}$ input is high, the TDRA status bit is suppressed and data loading is inhibited.

ADLC INTERNAL REGISTER STRUCTURE

		RS1 RS0 = 00	RS1 RS0 = 01	RS1 RS0 = 10	RS1 RS0 = 11
	Bit #	Status Register #1	Status Register #2	Receiver Data Register	
Read Only Registers	0	RDA	Address Present	Bit 0	Same as RS1, RS0 = 10
	1	Status #2 Read Request	Frame Valid	Bit 1	
	2	Loop	Inactive Idle Received	Bit 2	
	3	Flag Detected (When Enabled)	Abort Received	Bit 3	
	4	$\overline{\text{CTS}}$	FCS Error	Bit 4	
	5	Tx Underrun	$\overline{\text{DCD}}$	Bit 5	
	6	TDRA/Frame Complete	Rx Overrun	Bit 6	
	7	IRQ Present	RDA (Receiver Data Available)	Bit 7	

	Bit #	Control Register #1	Control Register #2 (C_1b_0 = 0)	Control Register #3 (C_1b_0 = 1)	Transmitter Data (Continue Data)	Transmitter Data (Last Data) (C_1b_0 = 0)	Control Register #4 (C_1b_0 = 1)
Write Only Registers	0	Address Control (AC)	Prioritized Status Enable	Logical Control Field Select	Bit 0	Bit 0	Double Flag/Single Flag Interframe Control
	1	Receiver Interrupt Enable (RIE)	2 Byte/1 Byte Transfer	Extended Control Field Select	Bit 1	Bit 1	Word Length Select Transmit #1
	2	Transmitter Interrupt Enable (TIE)	Flag/Mark Idle	Auto, Address Extension Mode	Bit 2	Bit 2	Word Length Select Transmit #2
	3	RDSR Mode (DMA)	Frame Complete/ TDRA Select	01/11 Idle	Bit 3	Bit 3	Word Length Select Receive #1
	4	TDSR Mode (DMA)	Transmit Last Data	Flag Detected Status Enable	Bit 4	Bit 4	Word Length Select Receive #2
	5	Rx Frame Discontinue	CLR Rx Status	Loop/Non-Loop Mode	Bit 5	Bit 5	Transmit Abort
	6	Rx RESET	CLR Tx Status	Go Active on Poll/Test	Bit 6	Bit 6	Abort Extend
	7	Tx RESET	RTS Control	Loop On-Line Control DTR	Bit 7	Bit 7	NRZI/NRZ

CONTROL REGISTERS

CONTROL REGISTER 1 (CR1)

				7	6	5	4	3	2	1	0
RS1	RS0	$R/\overline{W}$	AC	TxRS	RxRS	Discontinue	TDSR Mode	RDSR Mode	TIE	RIE	AC
0	0	0	X								

b0 — Address Control (AC) — AC provides another RS (Register Select) signal internally. The AC bit is used in conjunction with RS0, RS1, and $R/\overline{W}$ inputs to select particular registers, as shown in Table 2.

b1 — Receiver Interrupt Enable (RIE) — RIE enables/disables the interrupt request caused by the receiver section. 1...enable, 0...disable.

b2 — Transmitter Interrupt Eanble (TIE) — TIE enables/disables the interrupt request caused by the transmitter. 1...enable, 0...disable.

b3 — Receiver Data Service Request Mode (RDSR MODE) — The RDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When RDSR MODE is set, an interrupt request caused by RDA status is inhibited, and the ADLC does not request data transfer via the $\overline{\text{IRQ}}$ output.

b4 — Transmitter Data Service Request Mode (TDSR MODE) — The TDSR MODE bit provides the capability of operation with a bus system in the DMA mode when used in conjunction with the prioritized status mode. When TDSR MODE is set, an interrupt request caused by TDRA status is inhibited, and the ADLC does not request a data transfer via the $\overline{\text{IRQ}}$ output.

b5 — Rx Frame Discontinue (DISCONTINUE) — When the DISCONTINUE bit is set, the currently received frame is ignored and the ADLC discards the data of the current frame. The DISCONTINUE bit only discontinues the currently received frame and has no affect on subsequent frames, even if a following frame has entered the receiver section. The DISCONTINUE bit is automatically reset when the last byte of the frame is discarded. When the ignored frame is aborted by receiving an Abort or DCD failure, the DISCONTINUE bit is also reset.

b6 — Receiver Reset (Rx RS) — When the Rx RS bit is "1", the receiver section stays in the reset condition. All reciever sections, including the Rx FIFO register and the receiver status bits in both status registers, are reset. (During reset, the stored DCD status is reset but the DCD status bit follows the $\overline{\text{DCD}}$ input.) Rx RS is set by forcing a low level on the $\overline{\text{RESET}}$ input or by writing a "1" into the bit from the data bus. Rx RS must be reset by writing a "0" from the data bus after $\overline{\text{RESET}}$ has gone high.

b7 — Transmitter Reset (Tx RS) — When the Tx RS bit is "1", the transmitter section stays in the reset condition and transmits marks ("1's"). All transmitter sections, including the Tx FIFO and the transmitter status bits, are reset (FIFO cannot be loaded). During reset, the stored CTS status is reset but the CTS status bit follows the $\overline{\text{CTS}}$ input. Tx RS is set by forcing a low level on the $\overline{\text{RESET}}$ input or by writing a "1" from the data bus. It must be reset by writing a "0" after $\overline{\text{RESET}}$ has gone high.

CONTROL REGISTER 2 (CR2)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	RTS	CLR TxST	CLR RxST	Tx Last	FC/TDRA Select	F/M Idle	2/1 Byte	PSE
0	1	0	0								

b0 — Prioritized Status Enable (PSE) — When the PSE bit is SET, the status bits in both status registers are prioritized as defined in the Status Register section. When PSE is low, the status bits indicate current status without bit suppression by other status bits. The exception to this rule is the $\overline{\text{CTS}}$ status bit which always supresses the TDRA status.

b1 — 2-Byte/1-Byte Transfer (2/1 Byte) — When the 2/1 Byte bit is RESET the TDRA and RDA status bits then will indicate the availability of their respective data FIFO registers for a single-byte data transfer. Similarly, if 2/1 Byte is set, the TDRA and RDA status bit indicate when two bytes of data can be moved without a second status read.

b2 — Flag/Mark Idle Select (F/M Idle) — The F/M Idle bit selects Flag characters or bit-by-bit Mark Idle for the time fill or the idle state of the transmitter. When Mark Idle is selected, Go-Ahead code can be generated for loop operation in conjunction with the 01/11 Idle control bit (C_3b_3). 1...Flag time fill, 0...Mark Idle.

b3 — Frame Complete/TDRA Select (FC/TDRA Select) — The FC/TDRA Select bit selects TDRA status or FC status for the TDRA/FC status bit indication. 1...FC status, 0...TDRA status.

b4 — Transmit Last Data (Tx Last) — Tx Last bit provides another method to terminate a frame. This bit should be set after loading the last data byte and before the Tx FIFO empties. When the Tx Last bit is set, the ADLC assumes the byte is the last byte and terminates the frame by appending CRCC and a closing Flag. This control bit is useful for DMA operation. Tx Last bit automatically returns to the "0" state.

b5 — Clear Receiver Status (CLR Rx ST) — When a "1" is written into the CLR Rx ST bit, a reset signal is generated for the receiver status bits in status registers #1 and #2 (except AP and RDA bits). The reset signal is enabled only for the bits which have been present during the last "read status" operation. The CLR Rx ST bit automatically returns to the "0" state.

b6 — Clear Transmitter Status (CLR Tx ST) — When a "1" is written into CLR Tx ST bit, a reset signal is generated for the transmitter status bits in status register #1 (except TDRA). The reset signal is enabled for the bits which have been present during the last "read status" operation. The CLR Tx ST bit automatically returns to the "0" state.

b7 — Request-to-Send Control (RTS) — The RTS bit, when high, causes the $\overline{\text{RTS}}$ output to be low (the active state). When the RTS bit returns low and data is being transmitted, the $\overline{\text{RTS}}$ output remains low until the last character of the frame (the closing Flag or Abort) has been completed and the Tx FIFO is empty. If the transmitter is idling when the RTS bit returns low, the $\overline{\text{RTS}}$ output will go high (the inactive state) within two bit times.

CONTROL REGISTER 3 (CR3)

				7	6	5	4	3	2	1	0
RS1	RS0	R/$\overline{W}$	AC	LOC/ DTR	GAP/ TST	Loop	FDSE	01/11 Idle	AEX	CEX	LCF
0	1	0	1								

b0 — Logical Control Field Select (LCF) — The LCF select bit causes the first byte(s) of data belonging to the information field to remain 8-bit characters until the logical control field is complete. The logical control field (when selected) is an automatically extendable field which is extended when bit 7 of a logical control character is a "1." When the LCF Select bit is reset the ADLC assumes no logical control field is present for either the transmit or received data channels. When the logical control field is terminated, the word length of the information data is then defined by WLS_1 and WLS_2.

b1 — Extended Control Field Select (C_{EX}) — When the C_{EX} bit is a "1", the control field is extended and asusmed to be 16 bits. When C_{EX} is "0", the control field is assumed to be 8 bits.

b2 — Auto/Address Extend Mode (A_{EX}) — The A_{EX} bit when "low" allows full 8 bits of the address octet to be utilized for addressing because address extension is inhibited. When the A_{EX} bit is "high," bit 0 of address octet equal to "0" causes the Address field to be extended by one octet. The exception to this automatic address field extension is when the first address octet is all "0's" (the Null Address).

b3 — 01/11 Idle (01/11 Idle) — The 01/11 Idle Control bit determines whether the inactive (Mark) idle condition begins with a "0" or not. If the 01/11 Idle Control is SET, the closing flag (or Abort) will be followed by a 011111...pattern. This is required of the controller for the "Go Ahead" character in the Loop Mode. When 01/11 is RESET, the idling condition will be all "1's".

b4 — Flag Detect Status Enable (FDSE) — The FDSE bit enables the FD status bit in Status Register #1 to indicate the occurrence of a received Flag character. The status indication will be accompanied by an interrupt if RIE is SET. Flag detection will cause the Flag Detect output to go low for 1 bit time regardless of the state of FDSE.

b5 — LOOP/NON-LOOP Mode (LOOP) — When the LOOP bit is set, loop mode operation is selected and the GAP/TST control bit, LOC/DTR control bit and LOC/DTR output are selected to perform the loop control functions. When LOOP is reset, the ADLC operates in the point-to-point data communications mode.

b6 — Go Active On Poll/Test (GAP/TST) — *In the Loop Mode* — The GAP/TST bit is used to respond to the poll sequence and to begin transmission. When GAP/TST is set, the receiver searches for the "Go Ahead" (or End of Poll, EOP). The receiver "Go ahead" is converted to an opening Flag and the ADLC starts its own transmission. When GAP/TST is reset during the transmission, the end of the frame (the completion of Flag or Abort) causes the termination of the "go-active-on-poll" operation and the Rx Data to Tx Data link is re-established. The ADLC then returns to the "loop-on-line" state.

In the Non-Loop Mode — The GAP/TST bit is used for self-test purposes. If GAP/TST bit is set, the TxD output is connected to the RxD input internally, and provides a "loop-back" feature. For normal operation, the GAP/TST bit should be reset.

b7 — Loop On-Line Control/DTR Control (LOC/DTR) — *In the Loop Mode* — The LOC/DTR bit is used to go on-line or to go off-line. When LOC/DTR is set, the ADLC goes to the on-line state after 7 consecutive "1's" occur at the RxD input. When LOC/DTR is reset, the ADLC goes to the "off-line" state after eight consecutive "1's" occur at the RxD input.

In the Non-Loop Mode — The LOC/DTR bit directly controls the Loop On-Line/$\overline{DTR}$ output state. 1...$\overline{DTR}$ output goes to low level, 0...$\overline{DTR}$ output goes to high level.

CONTROL REGISTER 4 (CR4)

				7	6	5	4	3	2	1	0
RS1	RS0	$R/\overline{W}$	AC	NRZI/NRZ	ABT_{EX}	ABT	Rx		Tx		"FF"/F
1	1	0	1				WLS_2	WLS_1	WLS_2	WLS_1	

b0 — Double Flag/Single Flag Interframe Control ("FF"/"F") — The "FF"/"F" Control bit determines whether the transmitter will transmit separate closing and opening Flags when frames are transmitted successively. When the "FF"/"F" control bit is low, the closing flag of the first frame will serve as the opening flag of the second frame. When the bit is high, independent opening and closing flags will be transmitted.

b1, b2 — Transmitter Word Length Select (Tx WLS1 and WLS2) — Tx WLS1 and WLS2 are used to select the word length of the transmitter information field. The encoding format is shown in Table 3.

b3, b4 — Receiver Word Length Select (Rx WLS1 and WLS2) — Rx WLS1 and WLS2 are used to select the word length of the receiver information field. The encoding format is shown in Table 3.

TABLE 3 — I-FIELD CHARACTER LENGTH SELECT

WLS_1	WLS_2	I-Field Character Length
0	0	5 bits
1	0	6 bits
0	1	7 bits
1	1	8 bits

b5 — Transmit Abort (ABT) — The ABT bit causes an Abort (at least 8 bits of "1" in succession) to be transmitted. The Abort is initiated and the Tx FIFO is cleared when the control bit goes high. Once Abort begins, the Tx Abort control bit assumes the low state.

b6 — Abort Extend (ABT_{EX}) — If ABT_{EX} is set, the abort code initiated by ABT is extended up to at least 16 bits of consecutive "1's", the mark Idle State.

b7 — NRZI (Zero Complement)/NRZ Select (NRZI/NRZ) — NRZI/NRZ bit selects the transmit/receive data format to be NRZI or NRZ in both Loop Mode or Non-Loop mode operation. When the NRZI Mode is selected, a 1-bit delay is added to the transmitted data (TxD) to allow for NRZI encoding. 1...NRZI, 0...NRZ.

NOTE

NRZI coding — The serial data remains in the same state to send a binary "1" and switches to the opposite state to send a binary "0".

STATUS REGISTER

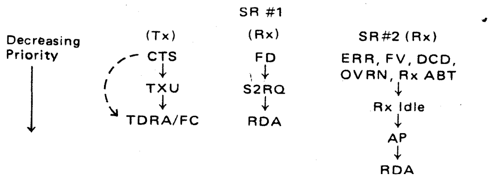
The Status Register #1 is the main status register. The IRQ bit indicates whether the ADLC requests service or not. The S2RQ bit indicates whether any bits in status register #2 request any service. TDRA and RDA, because they are most often used, are located in bit positions that are more convenient to test. RDA reflects the state of the RDA bit in status register #2.

The Status Register #2 provides the detailed status information contained in the S2RQ bit and these bits reflect receiver status. The FD bit is the only receiver status which is not indicated in status register #2.

The prioritized status mode provides maximum efficiency in searching the status bits and indicates only the most important action required to service the ADLC. The priority trees of both status registers are provided in Figure 10.

Reading the status register is a non-destructive process. The method of clearing status depends upon the bit's function and is discussed for each bit in the register.

FIGURE 10 — STATUS REGISTER PRIORITY TREE (PSE = 1)



*Prioritized even when PSE = 0

NOTE: Status bit above will inhibit one below it.

STATUS REGISTER 1 (SR1)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				IRQ	TDRA/FC	TXU	CTS	FD	LOOP	S2RQ	RDA
0	0	1	X								

b0 — Receiver Data Available (RDA) — The RDA status bit reflects the state of the RDA status bit in status register #2. It provides the means of achieving data transfers of received data in the full Duplex Mode without having to read both status registers.

b1 — Status Register #2 Read Request (S2RQ) — All the status bits (stored conditions) of status register #2 (except RDA bit) are logically ORed and indicated by the S2RQ status bit. Therefore, S2RQ indicates that status register #2 needs to be read. When S2RQ is "0", it is not necessary to read status register #2. The bit is cleared when the appropriate bits in status register #2 are cleared or when Rx Reset is used.

b2 — Loop Status (LOOP) — The LOOP status bit is used to monitor the loop operation of the ADLC. This bit does not cause an IRQ. When Non-Loop Mode is selected, LOOP bit stays "0". When Loop Mode is selected, the LOOP status bit goes to "1" during "On-Loop" condition. When ADLC is in an "Off-Loop" condition or "Go-Active-On-Poll" condition, the LOOP status bit is a "0".

b3 — Flag Detected (FD) — The FD Status bit indicates that a flag has been received if the Flag Detect Enable control bit has been set. The bit goes high at the last bit of the Flag Character received (when the Flag Detect Output goes low) and is stored until cleared by Clear Rx Status or Rx Reset.

b4 — Clear-to-Send (CTS) — The $\overline{CTS}$ input positive transition is stored in the status register and causes an IRQ (if Enabled). The stored CTS condition and its IRQ are cleared by Clear Tx Status control bit or Tx Reset bit. After the stored status is reset, the CTS status bit reflects the state of the $\overline{CTS}$ input.

b5 — Transmitter Underrun (TxU) — When the transmitter runs out of data during a frame transmission, an underrun occurs and the frame is automatically terminated by transmitting an Abort. The underrun condition is indicated by the TxU status bit. TxU can be cleared by means of the Clear Tx Status Control bit or by Tx Reset.

b6 — Transmitter Data Register Available/Frame Complete (TDRA/FC) — The TDRA Status bit serves two purposes depending upon the state of the Frame Complete/TDRA Select control bit. When this bit serves as a TDRA status bit, it indicates that data (to be transmitted) can be loaded into the Tx Data FIFO register. The first register (Register #1) of the Tx Data FIFO being empty (TDRA = "1") will be indicated by the TDRA Status bit in the "1-Byte Transfer Mode." The first two registers (Registers #1 and #2) must be empty for TDRA to be high when in the "2-Byte Transfer Mode." TDRA is inhibited by Tx Reset, or $\overline{CTS}$ being high.

When the Frame Complete Mode of operation is selected, the TDRA/FC status bit goes high when an abort is transmitted or when a flag is transmitted with no data in the Tx FIFO. The bit remains high until cleared by resetting the TDRA/FC control bit or setting the Tx Reset bit.

b7 — Interrpt Request (IRQ) — The Interrupt Request status bit indicates when the $\overline{IRQ}$ output is in the active state ($\overline{IRQ}$ Output = "0"). The IRQ status bit is subject to the same interrupt enables (RIE, TIE) as the $\overline{IRQ}$ output, i.e., with both transmitter and receiver interrupts enabled, the IRQ status bit is a logical ORed indication of Status Register 1 status bits. The IRQ bit only reflects the set status bits which have interrupts enabled. The IRQ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests.

STATUS REGISTER 2 (SR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				RDA	OVRN	DCD	ERR	Rx ABT	Rx Idle	FV	AP
0	1	1	X								

b0 — Address Present (AP) — The AP status bit provides the frame boundary and indicates an Address octet is available in the Rx Data FIFO register. In the Extended Addressing Mode, the AP bit continues to indicate addresses until the Address field is complete. The Address present status bit is cleared by reading data or by Rx Reset.

b1 — Frame Valid (FV) — The FV status bit provides the frame boundary indication to the MPU and also indicates that a frame is complete with no error. The FV status bit is set when the last data byte of a frame is transferred into the last location of the Rx FIFO (available to be read by MPU). Once FV status is set, the ADLC stops further data transfer into the last location of the Rx FIFO (in order to prevent the mixing of two frames) until the status bit is cleared by the Clear Rx Status bit or Rx Reset.

b2 — Inactive Idle Received (Rx Idle) — The Rx Idle status bit indicates that a minimum of 15 consecutive "1's" have been received. The event is stored within the status register and can cause an interrupt. The interrupt and stored condition are cleared by the Clear Rx Status Control bit. The Status bit is the Logical OR of the receiver idling detector (which continues to reflect idling until a "0" is received) and the stored inactive idle condition.

b3 — Abort Received (RxABT) — The RxABT status bit indicates that seven or more consecutive "1's" have been received. Abort has no meaning under out-of-frame conditions; therefore, no interrupt nor storing of the status will occur unless a Flag has been detected prior to the Abort. An Abort Received when "in frame" is stored in the status register and causes an IRQ. The status bit is the logical OR of the stored conditions and the Rx Abort detect logic, which is cleared after 15 consecutive "1's" have occurred. The stored Abort condition is cleared by the Clear Rx Status Control bit or Rx Reset.

b4 — Frame Check Sequence/Invalid Frame Error (ERR) — When a frame is complete with a cyclic redundancy check (CRC) error or a short frame error (the frame does not have complete Address and Control fields), the ERR status bit is set instead of the Frame Valid status bit. Other functions, frame boundry indication and control function, are exactly the same as for the Frame Valid status bit. Refer to the FV status bit.

b5 — $\overline{\text{Data}}$ Carrier Detect (DCD) — A positive transition on the $\overline{\text{DCD}}$ input is stored in the status register and causes an IRQ (if enabled). The stored DCD condition and its IRQ are cleared by the Clear Rx Status Control bit or RX Reset. After stored status is reset, the DCD status bit follows the state of the input. Both the stored DCD condition and the $\overline{\text{DCD}}$ input cause the reset of the receiver section when they are high.

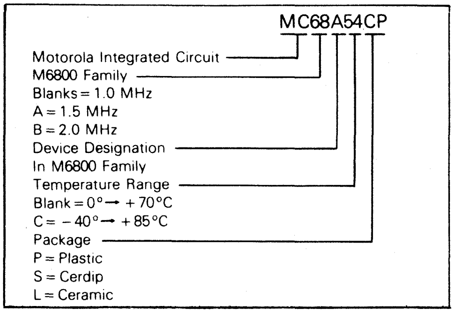
b6 — Receiver Overrun (OVRN) — OVRN status indicates that receiver data has been transferred into the Rx FIFO when it is full, resulting in data loss. The OVRN status is cleared by the Clear Rx Status bit or Rx Reset. Continued overrunning only destroys data in the first FIFO Register.

b7 — Receiver Data Available (RDA) — The Receiver Data Available status bit indicates when receiver data can be read from the Rx Data FIFO. When the prioritized status mode is used, the RDA bit indicates that non-address and non-last data are available in the Rx FIFO. The receiver data being present in the last register of the FIFO causes RDA to be high for the "1-Byte Transfer Mode." The RDA bit being high indicates that the last two registers are full when in the "2-Byte Transfer Mode." The RDA status bit is reset automatically when data is not available.

PROGRAMMING CONSIDERATIONS

1. **Status Priority** — When the prioritized status mode is used, it is best to test for the lowest priority conditions first. The lowest priority conditions typically occur more frequently and are the most likely conditions to exist when the processor is interrupted.
2. **Stored vs Present Status** — Certain status bits (DCD, CTS, Rx Abort, and Rx Idle) indicate a status which is the logical OR of a stored and a present condition. It is the stored status that causes an interrupt and which is cleared by a Status Clear control bit. After being cleared, the status register will reflect the present condition of an input or a receiver input sequence.
3. **Clearing Status Registers** — In order to clear an interrupt with the two Status Clear control bits, a particular status condition must be read before it can be cleared. In the prioritized mode, clearing a higher priority condition might result in another $\overline{IRQ}$ caused by a lower priority condition whose status was suppressed when a status register was first read. This guarantees that a status condition is never inadvertently cleared.
4. **Clearing the Rx FIFO** — An Rx Reset will effectively clear the contents of all three Rx FIFO bytes. However, the FIFO may contain data from two different frames when abort or DCD failure occurs. When this happens, the data from a previously closed frame (a frame whose closing flag has been received) will not be destroyed.
5. **Servicing the Rx FIFO in a 2-Byte Mode** — The procedure for reading the last bytes of data is the same, regardless of whether the frame contains an even or an odd number of bytes. Continue to read 2 bytes until an interrupt cocurs that is caused by an end-of-frame status (FV or ERR). When this occurs, indicating the last byte either has been read or is ready to be read, switch temporarily to the 1-byte mode with no prioritized status (control register 2). Test RDA to indicate whether a 1-byte read should be performed. Then clear the frame end status.
6. **Frame Complete Status and $\overline{RTS}$ Release** — In many cases, a MODEM will require a delay for releasing $\overline{RTS}$. An 8-bit or 16-bit delay can be added to the ADLC $\overline{RTS}$ output by using an Abort. At the end of a transmission, frame complete status will indicate the frame completion. After frame complete status goes high, write "1" into the Abt control bit (and Abt Extend bit if a 16-bit delay is required). After the Abt control bit is set, write "0" into the $\overline{RTS}$ control bit. The transmitter will transmit eight or sixteen "1's" and the $\overline{RTS}$ output will then go high (inactive).
7. **Note to users not using the MC6800** — (a) Care should be taken when performing a write followed by a read on successive E pulses at a high frequency rate. Time must be allowed for status changes to occur. If this is done, the time that E is low between successive write/read E pulses should be at least 500 ns. (b) The ADLC is a completely static part. However, the E frequency should be high enough to move data through the FIFOs and to service the peripheral requirements. Also, the period between successive E pulses should be less than the period of RxC or TxC in order to maintain synchronization between the data bus and the peripherals.
8. **Clear-to-Send ($\overline{CTS}$)** — The $\overline{CTS}$ input, when high, provides a real-time inhibit to the TDRA status bit and its associated interrupt. All other status bits will be operational. Since it inhibits TDRA, $\overline{CTS}$ also inhibts the TDSR DMA request. The $\overline{CTS}$ input being high does not affect any other part of the transmitter. Information in the Tx FIFO and Tx Shift Register will, therefore, continue to be transmitted as long as the Tx CLK is running.

ORDERING INFORMATION



Level 1 "S" = 10 Temp Cycles — (−25 to 150°C),
Hi Temp testing at T_A max.
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2.

Speed	Device	Temperature Range
1.0 MHz	MC6854P,L,S MC6854CP,CL,CS	0 to 70°C −40 to +85°C
1.5 MHz	MC68A54P,L,S MC68A54CP,CL,CS	0 to +70°C −40 to +85°C
2.0 MHz	MC68B54P,L,S	0 to +70°C

MOTOROLA

MC6859

Advance Information

DATA SECURITY DEVICE

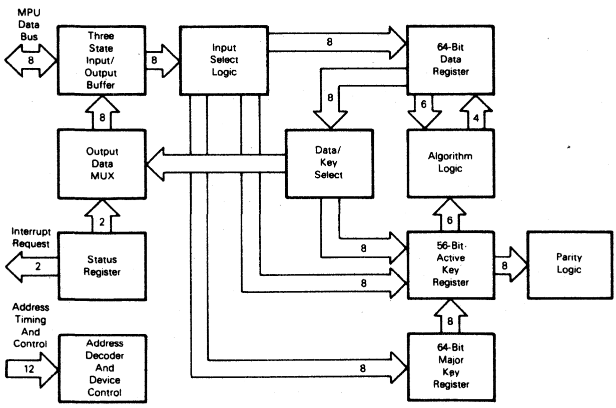
The MC6859 Data Security Device (DSD) is a monolithic MOS integrated circuit designed to be integrated into a wide range of equipment requiring protection of data by the employment of cryptographic measures.

The cryptographic algorithm utilized by the device is the Data Encryption Standard (DES) as adopted by the U.S. Department of Commerce, National Bureau of Standards (NBS), in publication FIPS PUB 46 (1-15-1977).

Through the use of flexible on-chip control and status circuitry and external control lines, the DSD provides direct capability of adapting the functional implementation of the DES algorithm for various specific system requirements for data protection.

- Direct Compatibility with the M6800 Microprocessor Family
- Data Encryption Standard Algorithm
- Two Separate Interrupt Output Lines for Program Controlled Interrupt Capability
- Up to 400 KBPS Throughput Rate of 64-Bit Block Cipher (Exclusive of Software Overhead)
- TTL Compatible
- Single +5 V Power Supply

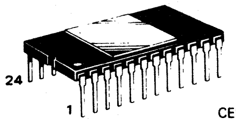
DATA SECURITY DEVICE BLOCK DIAGRAM



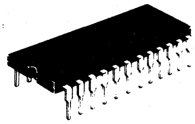
MOS

DEPLETION LOAD
(N-CHANNEL, SILICON-GATE)

DATA SECURITY DEVICE

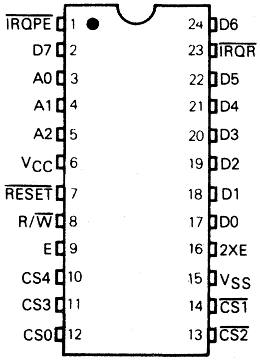


L SUFFIX
CERAMIC PACKAGE
CASE 716



S SUFFIX
CERDIP PACKAGE
CASE 623

PIN ASSIGNMENT



NOTICE

This product may not be exported without prior approval from the U.S. Department of State, Office of Munitions Control.

MAXIMUM RATINGS

Characteristics	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	T_L to T_H	°C
MC6859		0 to 70	
MC6859C		−40 to +85	
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Ceramic Package	θ_{JA}	60	°C/W
Cerdip Package		65	

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precuations be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. 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:

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0 to 5.25 V)	D0-D7	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA) (See Figure 2)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
Output Low Voltage		V_{OL}				V
(I_{Load} = 1.6 mA)	D0-D7		–	–	$V_{SS}+0.4$	
(I_{Load} = 3.2 mA) (See Figure 2)	$\overline{IRQPE}$, $\overline{IRQR}$		–	–	$V_{SS}+0.6$	
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{IRQPE}$, $\overline{IRQR}$	I_{OZ}	–	1.0	10	μA
Internal Power Dissipation (Measured at $T_A = T_L$)		P_{INT}	–	1000	–	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Others		–	–	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{IRQPE}$, $\overline{IRQR}$	C_{out}	–	–	50	pF

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Indent Number	Characteristic	Symbol	MC6859		MC68A59		MC68B59		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	100	20	100	20	100	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}*	165	–	80	–	60	–	ns

*See Mask Sets for mask set AK6 Data Setup time.

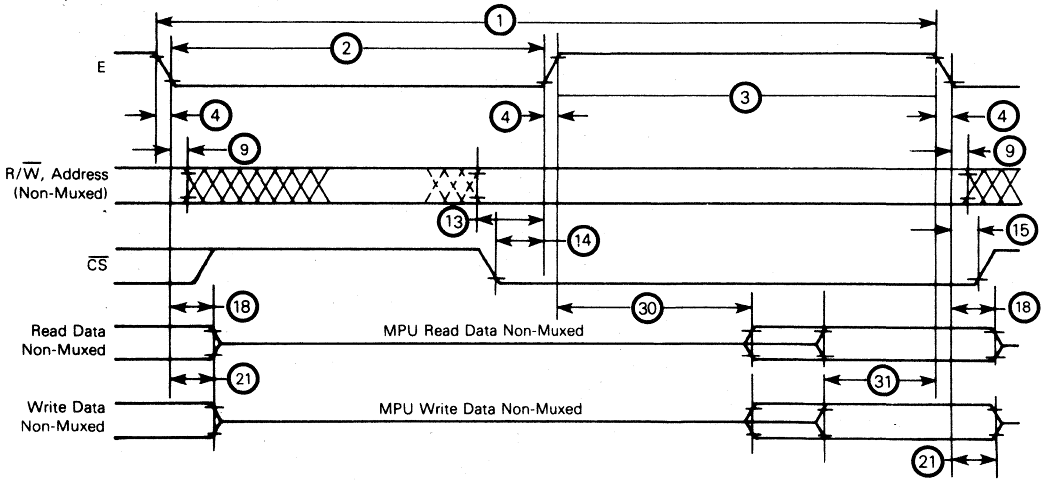


FIGURE 1 — BUS TIMING

Notes:
1. Voltage levels shown are $V_L \leq 0.4$ 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.

FIGURE 2 — BUS TIMING TEST LOADS

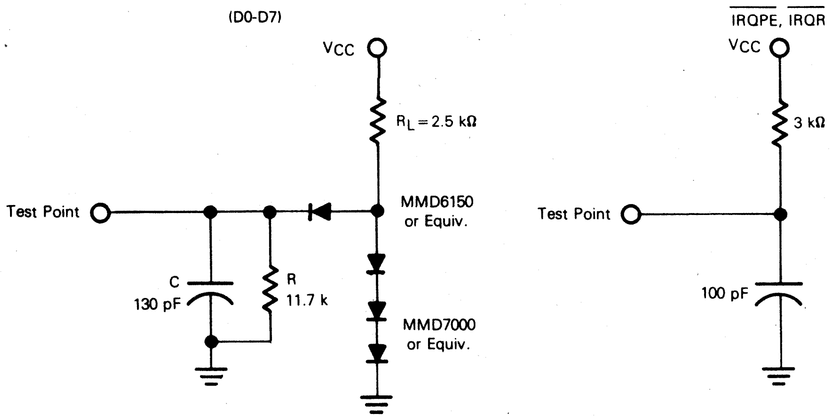
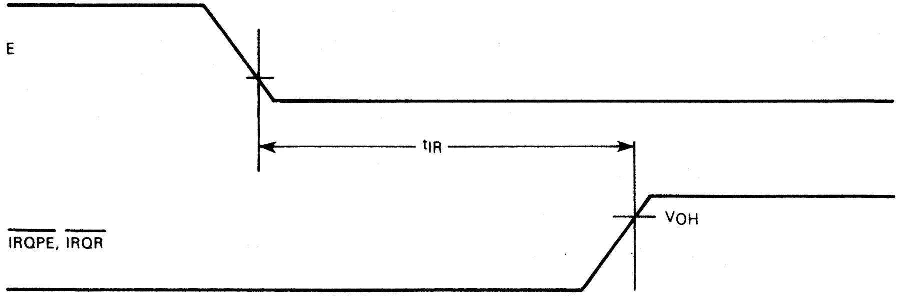


FIGURE 3 — INTERRUPT RELEASE TIME



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

BUS INTERFACE

The MC6859 Data Security Device (DSD) interfaces to the M6800 bus via an 8-bit bidirectional data bus, five chip select lines, a read/write (R/$\overline{\text{W}}$) line, an external $\overline{\text{RESET}}$ line, three register select lines, an Enable (System $\phi 2$) line, a 2XEnable (2XE) clock line, and two interrupt request lines. These signals permit the M6800 MPU to control the DSD and perform data transfers between the two.

Bidirectional Data Bus (D0-D7) — The bidirectional data lines (D0-D7) allow the transfer of information between the MPU and DSD. The data bus input/output drivers are three-state devices which remain in the high-impedance (off) state except when the MPU performs a DSD read or write operation.

Chip Select (CS0, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, CS3, and CS4) — These five signals are used to activate the data bus interface and allow DSD data transfers. When CS0 = CS3 = CS4 = 1 and $\overline{\text{CS1}}$ = $\overline{\text{CS2}}$ = 0, the device is selected.

Read/Write (R/$\overline{\text{W}}$) — With the DSD selected, this input controls the direction of data transfer on the data bus. When R/$\overline{\text{W}}$ is high, data in the DSD is read by the MPU on the trailing edge of E. A low state on the R/$\overline{\text{W}}$ line enables data transfer from the MPU on the trailing edge of the 2XE signal on the AK6 mask set and on the trailing edge of E for all other mask sets. (See Mask Sets.)

Enable (E) and 2XEnable (2XE) — The rising edge of the Enable input initiates data transfer from the DSD to the MPU during a read cycle. The falling edge of the Enable input latches MPU data into the DSD during a write cycle. The 2XE input is used in processing the encryption/decryption algorithm for all mask sets. E and 2XE are completely asynchronous. See section on Mask Sets for exceptions on prior revision of the DSD.

Reset ($\overline{\text{RESET}}$) — This input signal is used to initialize the internal control logic, status flags, and counters of the DSD. The contents of the active key register and major key register remain unchanged. The $\overline{\text{RESET}}$ function should be coupled with the system power-on reset to provide orderly system initialization. It may also be used as a master reset to the chip during system operation.

To abort the encryption algorithm before the required 320 clock cycles (2XE) have occurred, it is necessary to provide a $\overline{\text{RESET}}$ signal or a software reset command to the DSD. When this occurs, information in the data register and active key register is no longer valid. The contents of the major key register are unaffected.

Address Lines (A0, A1, A2) — These inputs are used in conjunction with the R/$\overline{\text{W}}$ line to select one of eleven possible DSD operations, as shown in Tables 1 and 2. The DSD is accessed via MPU read and write operations in much the same manner as a memory device.

NOTE:

Instructions performing operations directly on memory should not be used when the DSD is accessed. Since the DSD uses the R/$\overline{\text{W}}$ line as an additional register select input, read-modify-write type instructions will conflict with normal operation of the Data Security Device.

Modes — Operational and control modes are invoked by addressing DSD registers at the addresses in Tables 1 and 2.

TABLE 1 — OPERATIONAL MODES

Control Address		Operational Mode
A0, A1, A2	R/$\overline{\text{W}}$	
0 0 0	W	Write Data/"C" Key Operation (1st 7 bytes)
*1 0 1	W	Encipher Data
*0 0 1	W	Decipher Data
0 0 1	R	Read Data
0 1 0	R	Read Status

TABLE 2 — CONTROL MODES

Control Address		Control Mode
A0, A1, A2	R/$\overline{\text{W}}$	
1 0 0	W	Reset/Initialize
0 1 0	W	Enter Major Key
1 1 0	W	Enter Plain Secondary Key
*0 1 1	W	Decipher Secondary Key
*1 1 1	W	Encipher Secondary Key
1 0 0	R	Transfer Major Key

*Instruction initiated after eighth byte of Key Block entry.

Interrupt Requests — These open drain outputs are used to convey internal DSD status information to the MPU.

Ready Interrupt Request ($\overline{\text{IRQR}}$) — This active low output signals the MPU that the DSD is ready to initiate another operation. The $\overline{\text{IRQR}}$ signal will be inactive during encryption/decryption or key transfer.

Parity Error Interrupt Request ($\overline{\text{IRQPE}}$) — This active low output is used to signal the MPU that the DSD has detected a parity error. The $\overline{\text{IRQPE}}$ signal will remain low until a hardware or software reset is received.

DSD FUNCTIONAL DESCRIPTION

The MC6859 Data Security Device appears to an MPU system as an interface adapter device. An example of a system with the encryption function is shown in Figure 4.

Internal construction of the DSD is illustrated by the block diagram. The device consists of a single 8-bit data bus buffer with three-state operation, through which data may be entered into:

1) the 56-bit active key register
2) the 64-bit major key register
3) the 64-bit data register

Output data from the status register or the data register is also switched through the data bus buffers.

At the bus interface, the DSD data register appears as eight addressable memory locations to the MPU, through which the operational mode of the chip may be selected, chip status monitored, key or data written into the device, and data read from the device.

OPERATING MODES

As shown in Table 1, the operation of the DSD is split into five major modes:

1) status readout
2) loading of data or encrypted key
3) data encryption
4) data decryption
5) data readout

These and additional control modes are activated by three address input lines and a read/write input line.

Read Status — Only two bits are used in the status readout, D7 = Parity Error (PE) and D6 = $\overline{\text{READY}}$. The remaining six bits are always read as logic zeros. A read of the status register does not change these bits.

The PE flag is set when a parity error is detected while loading either a major or secondary key or when the active key is checked during algorithm operation. The PE flag remains set and the $\overline{\text{IRQPE}}$ signal will remain low until a hardware/software reset is received.

The $\overline{\text{READY}}$ flag is set and the $\overline{\text{IRQR}}$ output goes high whenever the device is processing a block of data. The flag is cleared, pulling the $\overline{\text{IRQR}}$ output low, whenever the DSD is not encoding/decoding data or transferring major key. $\overline{\text{IRQR}}$ may be tied to $\overline{\text{IRQ}}$ of a M6800 family processor for interrupt-driven encryption if no other peripherals share the $\overline{\text{IRQ}}$ line.

Encipher Data — To encipher an 8 byte block of data, the first seven bytes are written to the Write Data/"C" Key register. The eighth byte is written to the Encipher Data register. This automatically initiates the encryption process. Data is always processed using the current Active Key. During algorithm operation, the DSD constantly performs parity checking on the contents of the active key register. The busy flag will be set during encryption and then reset when the algorithm has finished. Completion requires 320 cycles of 2XE. During this time the DSD will ignore all external commands except status read, hardware reset and software reset.

Decipher Data — This process is identical to encipher data except that the eighth byte is written to the Decipher Data register. During decipher or encipher only a read status register, hardware reset, or software reset will be recognized. All other commands will be ignored.

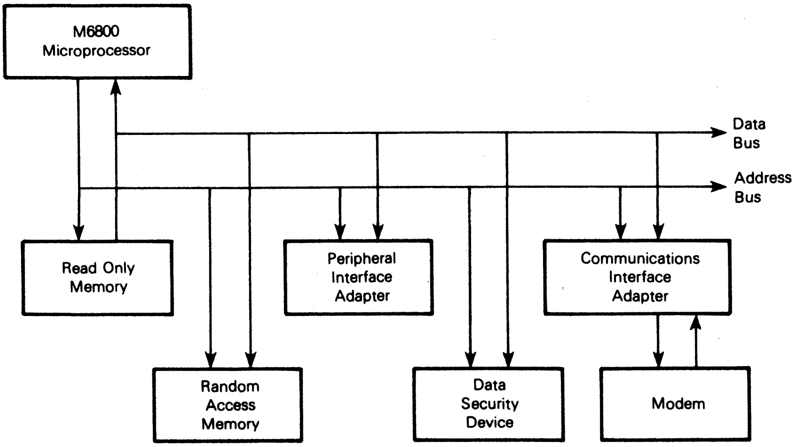
Read Data — This command is normally executed upon completion of the encipher/decipher algorithm (indicated by $\overline{\text{READY}}$ = 0). A read prior to completion of busy will result in all zeros being read from D0-D7. As each byte of data is read, zeros are automatically shifted into the data register to ensure data security.

CONTROL MODES

Shown in Table 2 are the control modes which facilitate programming of the primary and secondary keys.

Reset/Initialize — The DSD may be software reset by writing the reset/initialize command at any time the data bus is ignored. Like the hardware reset, this command initializes the internal control logic, status flags, and counters without altering the contents of the active key register or the major key register. If a hardware or software reset is issued during the algorithm processing, the information in the data register and active key register will no longer be valid. However, the contents of the major key register are not affected.

FIGURE 4 — M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM



Load Major Key – An unencrypted key will be entered into both the active key register and the major key register when eight consecutive bytes are written into the Enter Major Key Register. Parity error checking is automatically performed.

Load Plain Secondary Key – An unencrypted key may be loaded into the active key register and simultaneously checked for parity errors by writing eight consecutive bytes into the Enter Plain Secondary Key Register. The Major Key Register is unaffected.

Encipher Secondary Key – After a secondary key is loaded, it can be enciphered or deciphered (the source of an encrypted key is usually another DSD). A secondary key may be enciphered by loading the first seven bytes of plain text to the Write Data/"C" Key register. The eighth byte is entered to the Encipher Secondary Key register. This causes the secondary key to be enciphered using the current major key and automatically loaded into the Active Key register and checked for parity. This operation requires 328 cycles of 2XE.

Decipher Secondary Key – This function is similar to the Encipher Secondary Key operation. The first seven bytes of the key are loaded into the Write Data/"C" Key register. The eighth byte is entered by addressing the Decipher Secondary Key register. The secondary key is then deciphered using the current major key and automatically loaded into the Active Key register and checked for parity. This operation requires 328 cycles of 2XE.

Transfer Major Key – The contents of the Major Key register will be transferred to the Active Key register by a read of the Transfer Major Key register. The data bus is ignored. The Major Key register remains unchanged. This operation requires eight cycles of 2XE.

KEY CONVENTIONS

The key used for coding is a 56-bit data word plus eight bits of odd parity. In the DSD seven bits of key and the parity bit make up a key character. Eight key characters make up the total key information required by the DSD if parity errors are to be checked via the PE signal. If parity is not needed for some reason, then the parity bit need not be calculated and can be left as a zero. An example key with parity is shown in Table 3.

TABLE 3 – EXAMPLE KEY

Key Character	Hex Value	Binary Value	Parity
Byte 1	7C	0 1 1 1 1 1 0	0
Byte 2	A1	1 0 1 0 0 0 0	1
Byte 3	10	0 0 0 1 0 0 0	0
Byte 4	45	0 1 0 0 0 1 0	1
Byte 5	4A	0 1 0 0 1 0 1	0
Byte 6	1A	0 0 0 1 1 0 1	0
Byte 7	6E	0 1 1 0 1 1 1	0
Byte 8	57	0 1 0 1 0 1 1	1
Data Lines		**D7 D6 D5 D4 D3 D2 D1**	**D0**

TYPICAL SYSTEM OPERATION

For a communications link between a sender and one or more receivers, the following typical sequence might be used to transmit confidential data:

1) A software reset is issued to each DSD by its MPU.
2) The sending MPU loads a major key (eight bytes) into its DSD. This will serve as the active key if a secondary key is not entered.
3) The receiving station must also load this same major key before data transmission can begin. If the current major (or secondary) key is not known in advance, it can be transmitted by the sending MPU, but may not be encoded as the receiving MPU system has no key to decode it by. The MPU at the receiving station must be programmed with the mode and format being used for data transmission so its DSD can process the data correctly. At this point both the transmitting and receiving stations are ready for data transfer.
4) The sending MPU writes eight bytes of data into its DSD which enciphers them.
5) The sending MPU retrieves eight bytes of encrypted data from its DSD and transmits them to the receiving MPU.
6) The receiving MPU writes these eight bytes of data into its DSD to be deciphered.
7) The receiving MPU retrieves eight bytes of data from its DSD in the original plain text form.

Steps four through seven are repeated for each 8-byte block of data to be transmitted. If the major key or secondary key is to be changed, steps two and three must also be carried out.

SECURITY CONSIDERATIONS

The security of a system employing the NBS Data Encryption Standard (DES) depends only upon the key used, not the availability of the algorithm or of equipment used to implement the algorithm. The key is the most critical piece of information in the system and security of the key itself must be maintained both inside and outside the system.

Guidelines to be used in selecting a key are:

- Consider the key to be a single 56-bit number
- Avoid bias in selecting the key
- Change key as frequently as practical

One way to help ensure the security of the key is to make frequent use of secondary keys. Secondary keys can be generated by the sender and distributed selectively to one or more receivers. Since the MC6859 can encipher or decipher secondary keys using the major key, the sender can transmit the secondary key in encrypted form to further ensure system security. However, the receiver must be aware that a secondary key is being transmitted and must decrypt the key if it was sent in encrypted form.

Assuming that secrecy of the key is maintained, it is nearly impossible for an unauthorized user to decode an intercepted message into its original form. Since the DES algorithm utilizes a 56-bit active key, there are 2^{56} (or about 7×10^{16}) possible encrypted messages which must be searched to retrieve the original message. In addition, if the key were changed regularly only a small portion of the message would be retrieved for each successful exhaustive search. Therefore, the basic "block cipher" technique described in the Typical System Operation section is adequate for today's data security applications.

If additional security is required for some reason, several techniques can be used to increase data security. These include:

- Perform multiple encryption and/or decryption using the same key or different keys
- Reverse the algorithm (decipher-transmit-encipher)
- Utilize cipher feedback or other feedback techniques

The process of multiple encryption or decryption is an easy way to effectively increase the size of the key to any desired length. For example, the sender might successively encipher, decipher, and encipher a block of data using one key for the encipher operations and another for the decipher operation. The receiver would then have to decipher, encipher, and decipher the data using the same pair of keys. This technique would greatly increase data security while reducing throughput by a factor of three. Many such multiple encryption combinations are possible.

An easy way to increase security without reducing throughput is to perform the DES algorithm "in reverse." In other words, data or keys can be deciphered by the sender and then enciphered by the receiver to yield the original message. This technique works because the enciphering and deciphering algorithms are "mirror images" of each other.

Many different feedback techniques are available as alternatives to the basic 64-bit block cipher. One of these, known as cipher feedback (CFB), is described below. CFB is a byte-oriented implementation in that only one byte is transmitted at a time. Thus, throughput is reduced by a factor of eight (excluding software overhead). Implementation of the CFB technique is more dependent upon the system configuration than is the block cipher.

CFB ENCIPHER

The basic flow of the CFB encipher procedure is shown in Figure 5.

An initial eight byte fill of the RAM buffer must be done prior to accepting plain text bytes for enciphering. This information can be considered to be a data subset of the key, but may be any combination of eight-bit bytes as long as the deciphering device uses the same initial fill.

After the block of data in the RAM buffer is enciphered, one byte of enciphered data is read from the DSD. This byte is the key byte (Kt+1). The plain text byte (Pt+1) is exclusive ORed with the key byte and the result is the cipher text byte (Ct+1). The cipher text byte is shifted into the bottom of the RAM buffer and now is the newest byte in the block. The oldest previous byte is discarded. The cipher text byte is now available for use. The new RAM buffer block is loaded into the DSD for enciphering and yields the next key for further processing.

CFB DECIPHER

The basic flow of the decipher CFB operation is shown in Figure 6.

The same initial fill as used for enciphering must be used to initialize the decipher RAM buffer. The same key used to encipher must also be used to load the DSD active key register prior to receiving cipher text bytes. When a cipher text byte is received it is exclusive ORed with the key byte generated by the DSD and the result is the plain text data byte. The received cipher text byte is shifted into the RAM buffer and becomes the newest RAM buffer byte. The oldest RAM buffer byte is discarded and the eight byte RAM buffer is loaded into the DSD for block deciphering. One byte of the DSD data register is read out and this byte becomes the key byte for the next cipher text byte received.

FIGURE 5 — CFB ENCIPHER DATA FLOW (TRANSMITTING)

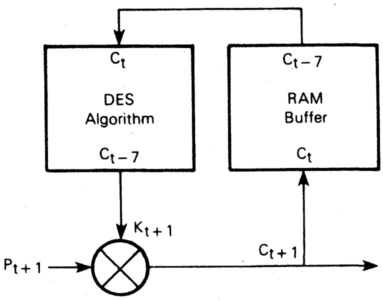
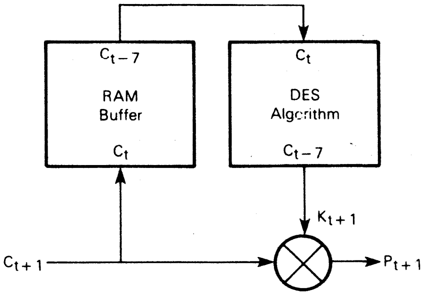


FIGURE 6 — CFB ENCIPHER DATA FLOW (RECEIVING)



To purchase a copy of the NBS Data Encryption Standard ask for the Federal Information Processing Standards (FIPS) Publication, FIPSP 46 at the following address:

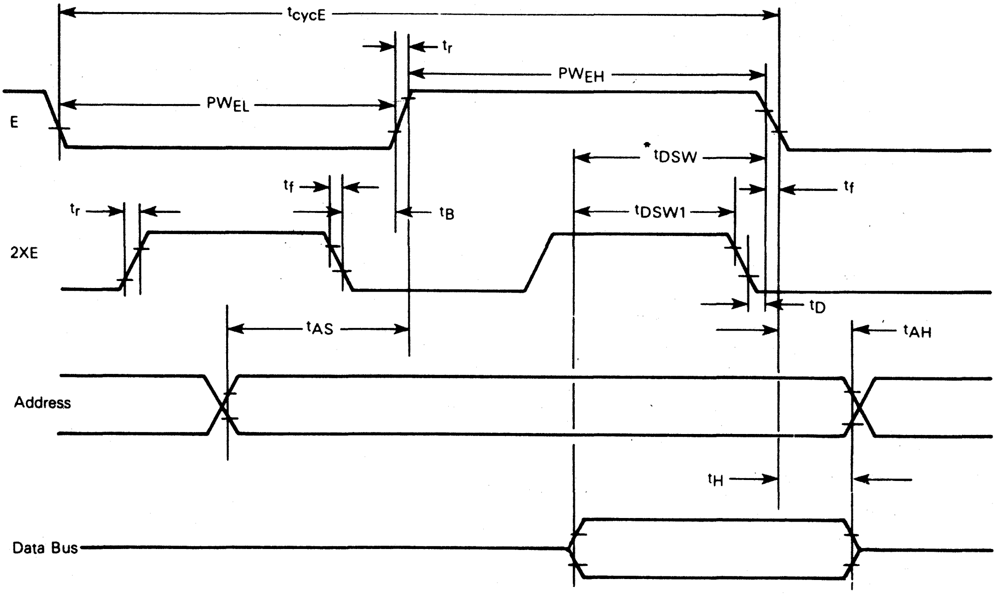
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

MASK SETS

Devices marked "AK6XXXX", where "AK6" is the mask set designation, latch MPU data into the DSD on the falling edge of 2XEnable during a write cycle as shown in Figure 7. E and 2XE must be synchronized. Devices marked with a mask set designation other than "AK6" latch MPU data into the DSD on the falling edge of Enable during a write operation as described in this data sheet. This is the only operational difference between devices manufactured using the old "AK6" mask set and those using the more recent mask sets.

The change to the "AK6" mask set allows the user to treat the DSD like any other M6800 peripheral since all data transfers are referenced to the Enable clock. E and 2XE are asynchronous on all mask sets except AK6. For this reason devices will no longer be produced using the "AK6" mask set. The "AK6" mask set can be identified by the "AK6" designation preceding the data code on top of the package.

FIGURE 7 — BUS WRITE TIMING CHARACTERISTICS
(WRITE INFORMATION INTO DSD)
(AK6 MASK SET ONLY)



*Data is latched into the internal registers on the falling edge of 2XE and while Enable is high. Therefore, for system considerations $t_{DSW} = t_{DSW1} + t_D + 2X\ t_f$. Minimize t_D to ensure operation at 1 MHz. t_{DSW1} is the data setup time for the "AK6" mask set.

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.

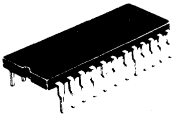
MOTOROLA

MC6860

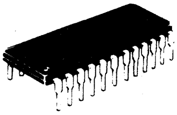
MOS

(N-CHANNEL, SILICON-GATE)

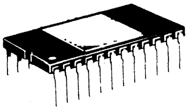
0-600 bps
DIGITAL MODEM



S SUFFIX
CERDIP PACKAGE
CASE 623



P SUFFIX
PLASTIC PACKAGE
CASE 709



L SUFFIX
CERAMIC PACKAGE
CASE 716

0-600 bps DIGITAL MODEM

The MC6860 is a MOS subsystem designed to be integrated into a wide range of equipment utilizing serial data communications.

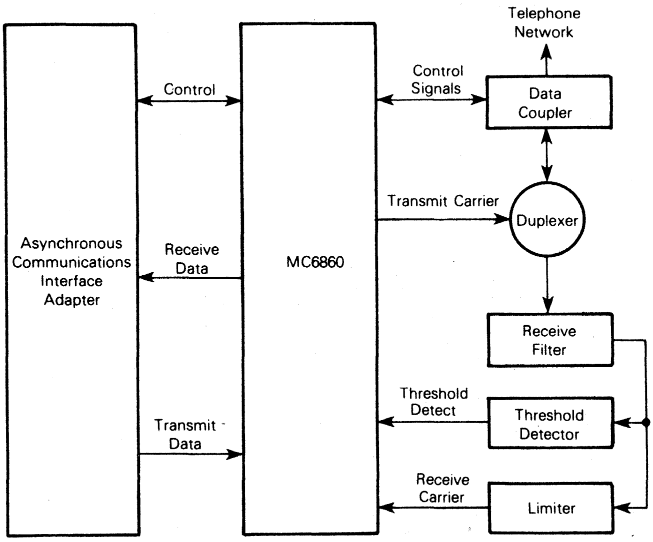
The modem provides the necessary modulation, demodulation and supervisory control functions to implement a serial data communications link, over a voice grade channel, utilizing frequency shift keying (FSK) at bit rates up to 600 bps. The MC6860 can be implemented into a wide range of data handling systems, including stand alone modems, data storage devices, remote data communication terminals and I/O interfaces for minicomputers.

N-channel silicon-gate technology permits the MC6860 to operate using a single-voltage supply and be fully TTL compatible.

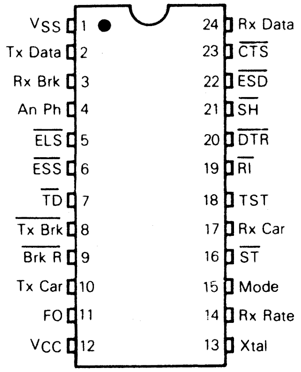
The modem is compatible with the M6800 microcomputer family, interfacing directly with the Asynchronous Communications Interface Adapter to provide low-speed data communications capability.

- Originate and Answer Mode
- Crystal or External Reference Control
- Modem Self Test
- Terminal Interfaces TTL-Compatible
- Full-Duplex or Half-Duplex Operation
- Automatic Answer and Disconnect
- Compatible Functions for 100 Series Data Sets
- Compatible Functions for 1001A/B Data Couplers

FIGURE 1 — TYPICAL MC6860 SYSTEM CONFIGURATION



PIN ASSIGNMENT



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 MC6860 MC6860S, MC6860C	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 Cerdip Plastic Ceramic	θ_{JA}	 65 120 60	°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.

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:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS

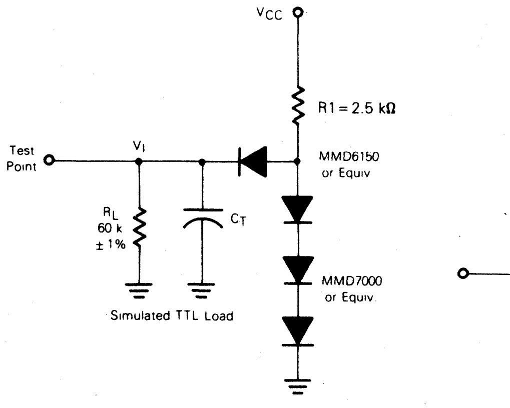
(V_{CC} = 5.0 ±5% Vdc, all voltages referenced to V_{SS} = 0, $T_A = T_L$ to T_H, all outputs loaded as shown in Figure 2 unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage, All Inputs Except Crystal	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage, All Inputs Except Crystal	V_{IL}	V_{SS}	–	0.80	V
Crystal Input Voltage (Crystal Input Driven from an External Reference, Input Coupling Capacitor = 200 pF, Duty Cycle = 50 ± 5%)	V_{in}	1.5	–	2.0	$V_{p\text{-}p}$
Input Current ($V_{in} = V_{SS}$) All Inputs Except Rx Car, Tx Data, $\overline{TD}$, $\overline{TST}$, $\overline{RI}$, $\overline{SH}$ $\overline{RI}$, $\overline{SH}$ Inputs	I_{in}	– –	– –	–0.2 –1.6	mA
Input Leakage Current (V_{in} = 7.0 V, $V_{CC} = V_{SS}$, T_A = 25°C)	I_{IL}	–	–	1.0	µA
Output High Voltage, All Outputs Except An Ph and Tx Car (I_{OH1} = –0.04 mA, Load A)	V_{OH1}	2.4	–	V_{CC}	V
Output Low Voltage, All Outputs Except An Ph and Tx Car (I_{OL1} = 1.6 mA, Load A)	V_{OL1}	V_{SS}	–	0.40	V
Output High Current, An Ph (V_{OH2} = 0.8 V, Load B)	I_{OH2}	0.30	–	–	mA
Output Low Voltage, An Ph (I_{OL2} = 0, Load B)	V_{OL2}	V_{SS}	–	0.30	V
Input Capacitance (f = 0.1 MHz, T_A = 25°C)	C_{in}	–	5.0	–	pF
Output Capacitance (f = 0.1 MHz, T_A = 25°C)	C_{out}	–	10	–	pF
Transmit Carrier Output Voltage (Load C)	V_{CO}	0.20	0.35	0.50	V(RMS)
Transmit Carrier Output 2nd Harmonic (Load C)	V_{2H}	–25	–32	–	dB
Input Transition Times, All Inputs Except Crystal (Operating in the Crystal Input Mode; from 10% to 90% Points)	t_r t_f	– –	– –	1.0* 1.0*	µs
Input Transition Times, Crystal Input (Operating in External Input Reference Mode)	t_r t_f	– –	– –	30 30	ns
Output Transition Times, All Outputs Except Tx Car (From 10% to 90% Points)	t_r t_f	– –	– –	5.0 5.0	µs
Internal Power Dissipation (All Inputs at V_{SS} and All Outputs Open) (Measured at $T_A = T_L$)	P_{INT}	–	–	340	mW

*Maximum Input Transition Times are ≤ 0.1 × Pulse Width or the specified maximum of 1.0 µs, whichever is smaller.

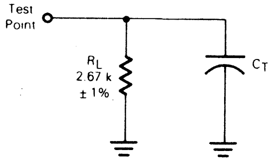
FIGURE 2 — OUTPUT TEST LOADS

Load A — TTL Output Load for Receive Break, Digital Carrier, Mode, Clear-to-Send, and Receive Data Outputs



C_T = 20 pF = total parasitic capacitance, which includes probe, wiring, and load capacitance

Load B — Answer Phone Load



Load C — Transmit Carrier Load

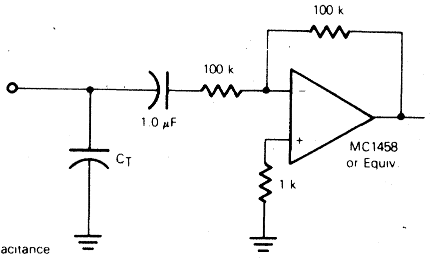
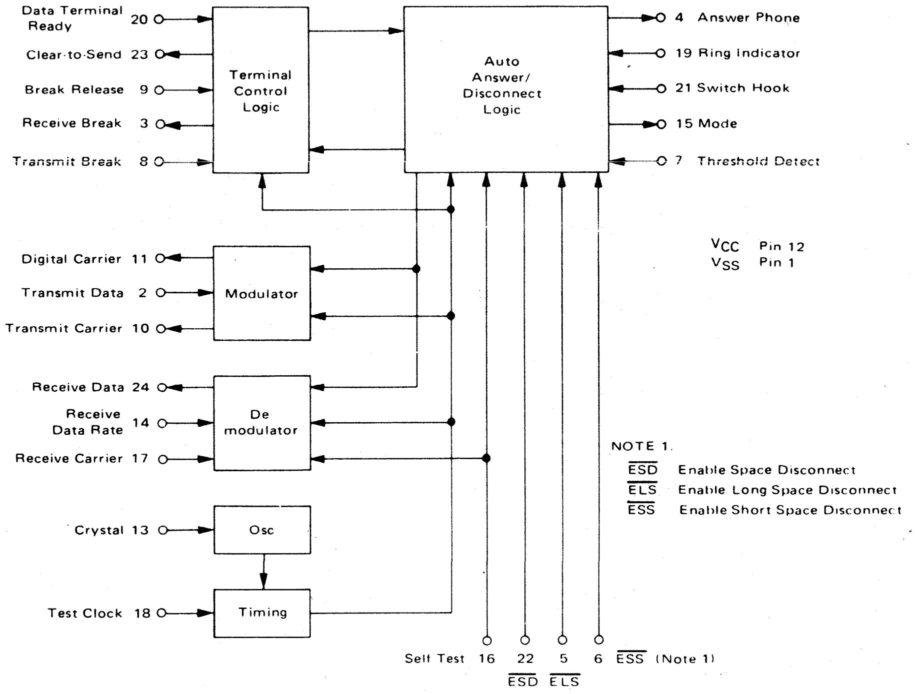


FIGURE 3 — BLOCK DIAGRAM



DEVICE OPERATION*

GENERAL

Figure 1 shows the modem and its interconnections. The data to be transmitted is presented in serial format to the modulator for conversion to FSK signals for transmission on the telephone line (refer to Figure 3). The modulator output is buffered before driving the line.

The FSK signal from the remote modem is received via the telephone line and filtered to remove extraneous signals such as the local Transmit Carrier. This filtering can be either a bandpass which passes only the desired band of frequencies or a notch which rejects the known interfering signal. The desired signal is then limited to preserve the axis crossings and fed to the demodulator where the data is recovered from the received FSK carrier.

The Supervisory Control provides the necessary commands and responses for handshaking with the remote modem, along with the interface signals to the data coupler and communication terminal. If the modem is a built-in unit, all input-output (I/O) logic need not be RS-232 compatible. The use of MC1488 and MC1489A line drivers and receivers will provide a RS-232 interface conforming to the EIA specification.

ANSWER MODE

Automatic answering is first initiated by a receipt of a Ring Indicator ($\overline{\mathrm{RI}}$) signal. This can be either a low level for at least 51 ms as would come from a CBS data coupler, or at least 20 cycles of a 20-47 Hz ringing singal (low level ≥50% of the duty cycle) as would come from a CBT data coupler. The presence of the Ring Indicator signal places the modem in the Answer Mode; if the Data Terminal Ready line is low, indicating the communication terminal is ready to send or receive data, the Answer Phone output goes high. This output is designed to drive a transistor switch which will activate

*See Tables 1 and 2 for delay time tolerances.

the Off Hook (OH) and Data Transmission (DA) relays in the data coupler. Upon answering the phone the 2225-Hz Transmit Carrier is turned on.

The originate modem at the other end detects this 2225-Hz signal and after a 450 ms delay (used to disable any echo suppressors in the telephone network) transmits a 1270-Hz signal which the local answering modem detects, provided the amplitude and frequency requirements are met. The amplitude threshold is set external to the modem chip. If the signal level is sufficient the $\overline{\mathrm{TD}}$ input should be low for 20 μs at least once every 32 ms. The absence of a threshold indication for a period greater than 51 ms denotes the loss of Receive Carrier and the modem begins hang-up procedures. Hang-up will occur 17 s after $\overline{\mathrm{RI}}$ has been released provided the handshaking routine is not re-established. The frequeny tolerance during handshaking is ± 100 Hz from the Mark frequency.

After the 1270-Hz signal has been received for 150 ms, the Receive Data is unclamped from a Mark condition and data can be received. The Clear-to-Send output goes low 450 ms after the receipt of carrier and data presented to the answer modem is transmitted. Refer to Figure 4.

AUTOMATIC DISCONNECT

Upon receipt of a space of 150 ms or greater duration, the modem clamps the Receive Break high. This condition exists until a Break Release command is issued at the receiving station. Upon receipt of a 0.3 s space, with Enable Short Space Disconnect at the most negative voltage (low), the modem automatically hangs up. If Enable Long Space Disconnect is low, the modem requires 1.5 s of continuous space to hang up. Refer to Figure 5.

ORIGINATE MODE

Upon receipt of a Switch Hook ($\overline{\mathrm{SH}}$) command the modem function is placed in the Originate Mode. If the Data Terminal Ready input is enabled (low) the modem will provide a logic high output at Answer Phone. The modem is now ready to receive the 2225-Hz signal from the remote answering modem. It will continue to look for this signal until 17 s after $\overline{\mathrm{SH}}$ has been released. Disconnect occurs if the handshaking routine is not established.

Upon receiving 2225 ± 100 Hz for 150 ms at an acceptable amplitude, the receive Data output is unclamped from a Mark condition and data reception can be accomplished. 450 ms after receiving a 2225-Hz signal, a 1270-Hz signal is transmitted to the remote modem. 750 ms after receiving the 2225-Hz signal, the Clear-to-Send output is taken low and data can now be transmitted as well as received. Refer to Figure 6.

INITIATE DISCONNECT

In order to command the remote modem to automatically hang up, a disconnect signal is sent by the local modem. This is accomplished by pulsing the normally low Data Terminal Ready into a high state for greater than 34 ms. The local modem then sends a 3 s continuous space and hangs up provided the Enable Space Disconnect is low. If the remote modem hangs up before 3 s, loss of Threshold Detect will cause loss of Clear-to-Send, which marks the line in Answer Mode and turns the carrier off in the Originate Mode.

If $\overline{\mathrm{ESD}}$ is high the modem will transmit data until hang-up occurs 3 s later. Receive Break is clamped 150 ms following the Data Terminal Ready interrupt. Refer to Figure 7.

INPUT/OUTPUT FUNCTIONS

Figure 8 shows the I/O interface for the low speed modem. The following is a description of each individual signal:

Receiver Carrier (Rx Car)

The Receive Carrier is the FSK input to the demodulator. The local Transmit Carrier must be balanced or filtered out and the remaining signal hard limited. The conditioned receive carrier is measured by the MC6860. Any half-cycle period greater than or equal to 429 ± 1.0 μs for the low band or 235 ± 1.0 μs for the high band is detected as a space. Resultant peak phase jitter is as follows:

Data Rate Bits per Second	Answer Mode ϕ_J (Peak %)	Originate Mode ϕ_J (Peak %)
300	7.0	3.7
200	4.7	2.5
150	3.5	1.8
110	2.6	1.4

Ring Indicator ($\overline{\mathrm{RI}}$)

The modem function will recognize the receipt of a call from the CBT data coupler if at least 20 cycles of the 20-47 Hz ringing singal (low level ≥ 50% of the duty cycle) are present. The CBS data coupler $\overline{\mathrm{RI}}$ signal must be level-converted to TTL according to the EIA RS-232 specification before interfacing it with the modem function. The receipt of a call from the CBS data coupler is recognized if the $\overline{\mathrm{RI}}$ signal is present for at least 51 ms. This input is held high except during ringing. An $\overline{\mathrm{RI}}$ signal automatically places the modem function in the Answer Mode.

Switch Hook ($\overline{\mathrm{SH}}$)

$\overline{\mathrm{SH}}$ interfaces directly with the CBT data coupler and via the EIA RS-232 level conversion for the CBS data coupler. An $\overline{\mathrm{SH}}$ signal automatically places the modem function in the Originate Mode.

$\overline{\mathrm{SH}}$ is low during origination of a call. The modem will automatically hang up 17 s after releasing $\overline{\mathrm{SH}}$ if the handshaking routine has not been accomplished.

Threshold Detect ($\overline{\mathrm{TD}}$)

This input is derived from an external threshold detector. If the signal level is sufficient, the $\overline{\mathrm{TD}}$ input must be low for 20 μs at least once every 32 ms to maintain normal operation. An insufficient signal level indicates the absence of the Receive Carrier; an absence for less than 32 ms will not cause channel establishment to be lost; however, data during this interval will be invalid.

If the signal is present and the level is acceptable at all times, then the threshold input can be low permanently.

Loss of threshold for 51 ms or longer results in a loss of Clear-to-Send. The Transmit Carrier of the originate modem is clamped off and a constant Mark is transmitted from the answer modem.

TIMING DIAGRAMS

FIGURE 4 — ANSWER MODE

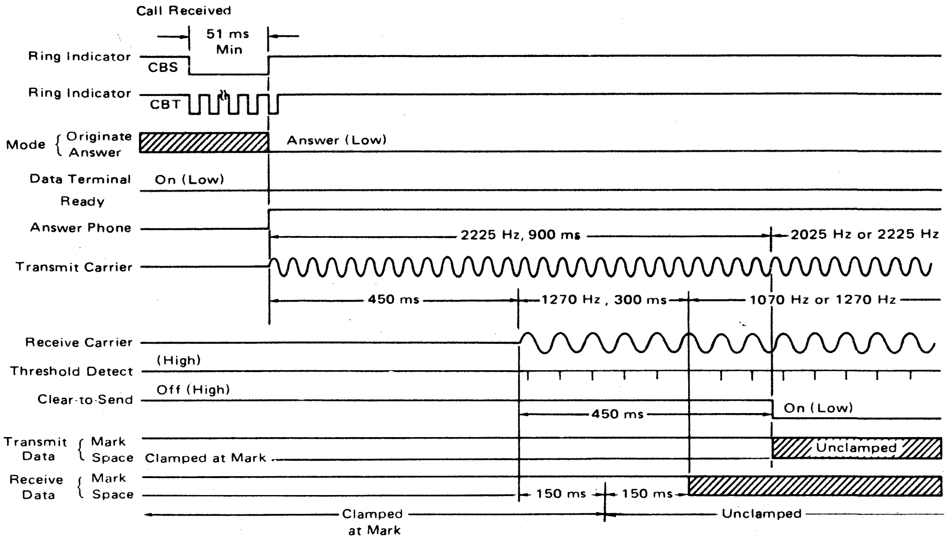


FIGURE 5 — AUTOMATIC DISCONNECT — LONG OR SHORT SPACE

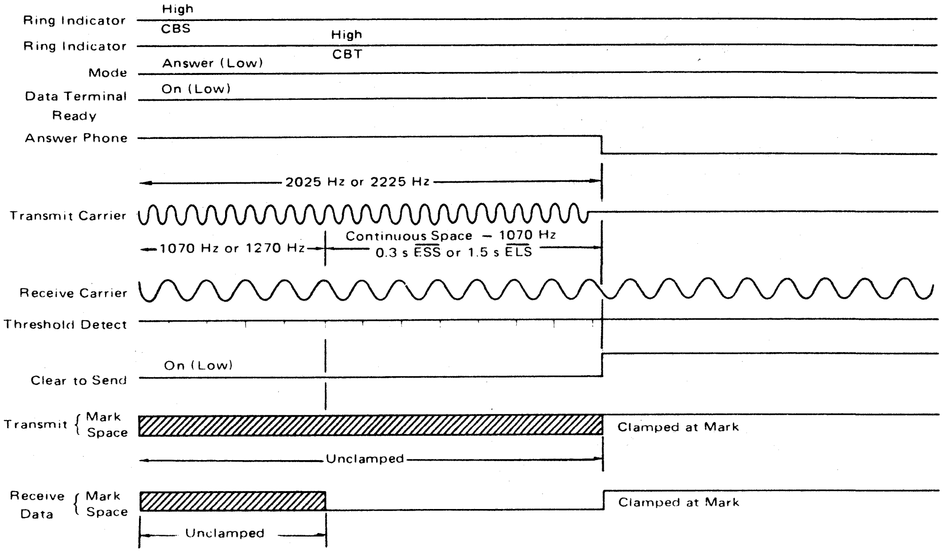


FIGURE 6 — ORIGINATE MODE

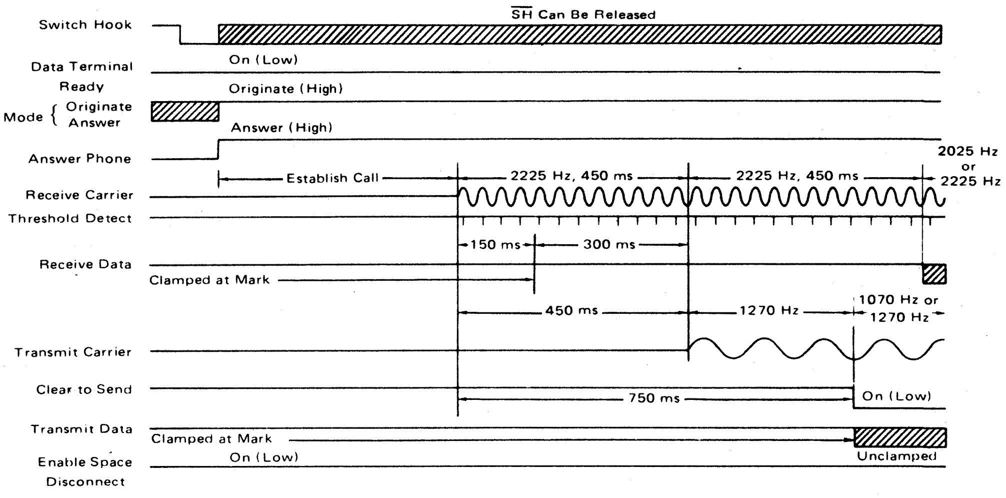
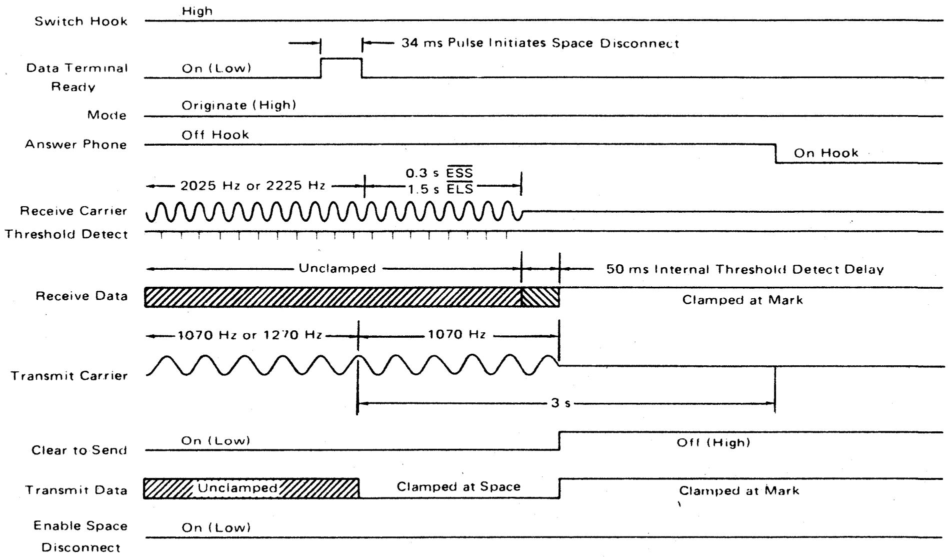


FIGURE 7 — INITIATE DISCONNECT



Receive Data Rate (Rx Rate)

The demodulator has been optimized for signal-to-noise performance at 300 bps and 600 bps. The Receive Data Rate input must be low for 0-600 bps and should be high for 0-300 bps.

Transmit Data (Tx Data)

Transmit Data is the binary information presented to the modem function for modulation with FSK techniques. A high level represents a Mark.

Data Terminal Ready ($\overline{\text{DTR}}$)

The Data Terminal Ready signal must be low before the modem function will be enabled. To initiate a disconnect, $\overline{\text{DTR}}$ is held high for 34 ms minimum. A disconnect will occur 3 s later.

Break Release ($\overline{\text{Brk R}}$)

After receiving a 150 ms space signal, the clamped high condition of the Receive Break output can be removed by holding Break Release low for at least 20 µs.

Transmit Break ($\overline{\text{Tx Brk}}$)

The Break command is used to signal the remote modem to stop sending data.

A Transmit Break (low) greater than 34 ms forces the modem to send a continuous space signal for 233 ms. Transmit Break must be initiated only after $\overline{\text{CTS}}$ has been established. This is a negative edge sense input. Prior to initiating $\overline{\text{Tx Brk}}$, this input must be held high for a minimum of 34 ms.

Enabled Space Disconnect ($\overline{\text{ESD}}$)

When $\overline{\text{ESD}}$ is strapped low and $\overline{\text{DTR}}$ is pulsed to initiate a disconnect, the modem transmits a space for either 3 s or until a loss of threshold is detected, whichever occurs first. If $\overline{\text{ESD}}$ is strapped high, data instead of a space is transmitted. A disconnect occurs at the end of 3 s.

Enable Short Space Disconnect ($\overline{\text{ESS}}$)

$\overline{\text{ESS}}$ is a strapping option which, when low, will automatically hang up the phone upon receipt of a continuous space for 0.3 s. *$\overline{\text{ESS}}$ and $\overline{\text{ELS}}$ must not be simultaneously strapped low.*

Enable Long Space Disconnect ($\overline{\text{ELS}}$)

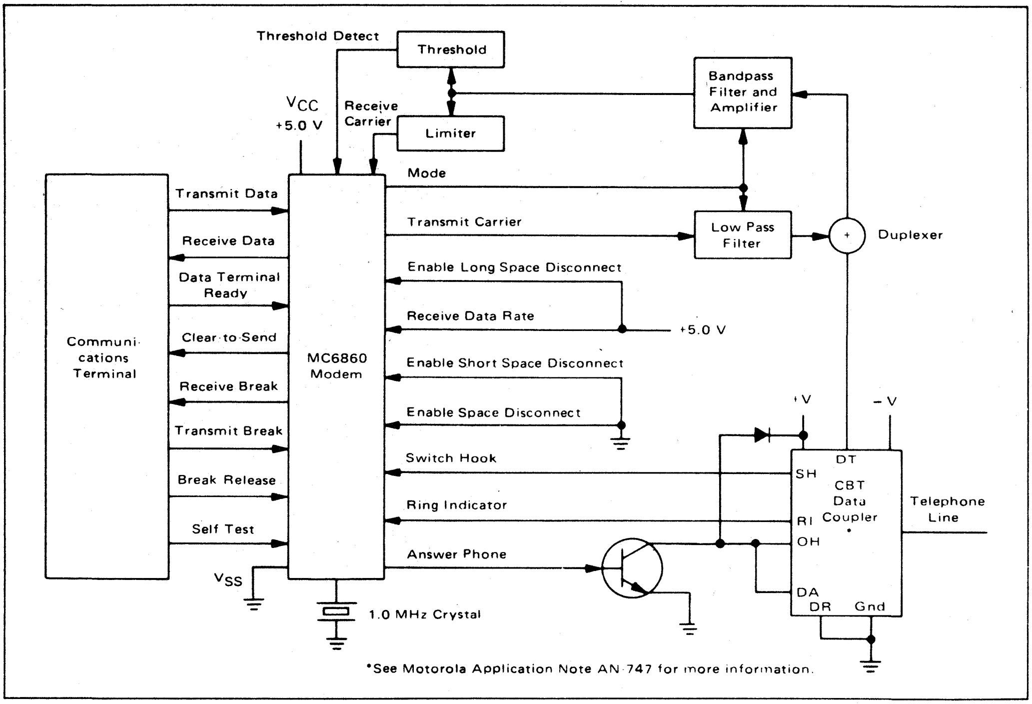
$\overline{\text{ELS}}$ is a strapping option which, when low, will automatically hang up the phone upon receipt of a continuous space for 1.5 s.

Crystal (Xtal)

A 1.0 MHz crystal with the following parameters is required to utilize the on-chip oscillator. A 1.0-MHz square wave can also be fed into this input to satisfy the clock requirement.

Mode:	Parallel
Frequency:	1.0 MHz ±0.1%
Series Resistance:	750 ohms max
Shunt Capacitance:	7.0 pF max
Temperature:	0-70°C
Test Level:	1.0 mW
Load Capacitance:	13 pF

FIGURE 8 — I/O INTERFACE CONNECTIONS FOR MC6860 (ORIGINATE/ANSWER MODEM)



*See Motorola Application Note AN-747 for more information.

When utilizing the 1.0 MHz crystal, external parasitic capacitance, including crystal shunt capacitance, must be ≤ 9 pF at the crystal input. Reliable crystal oscillator start-up requires that the V_{CC} power-on transition time be >15 milliseconds.

Test Clock (TST)

A test signal input is provided to decrease the test time of the chip. In normal operation this input *must be strapped low.*

Self Test ($\overline{ST}$)

When a low voltage level is placed on this input, the demodulator is switched to the modulator frequency and demodulates the transmitted FSK signal. Channel establishement, which occurred during the initial handshake, is not lost during self test. The Mode Control ouput changes state during Self Test, permitting the receive filters to pass the local Transmit Carrier.

$\overline{ST}$	$\overline{SH}$	$\overline{RI}$	Mode
H	⎺⎽⎺ *	H	H
H	H	L	L
L	⎺⎽⎺ *	H	L
L	H	L	H

*Note maximum $\overline{SH}$ low time in Table 1.

Answer Phone (An Ph)

Upon receipt of Ring Indicator or Switch Hook signal and Data Terminal Ready, the Answer Phone output goes high $[(\overline{SH}+\overline{RI})\bullet\overline{DTR}]$. This signal drives the base of a transistor which activates the Off Hook, and Data Transmission control lines in the data coupler. Upon call completion, the Answer Phone signal returns to a low level.

Mode

The Mode output indicates the Answer (low) or Originate (high) status of the modem. This output changes state when a Self Test command is applied.

Clear-To-Send ($\overline{CTS}$)

A low on the $\overline{CTS}$ output indicates the Transmit Data input has been unclamped from a steady Mark, thus allowing data transmission.

Receive Data (Rx Data)

The Receive Data output is the data resulting from demodulating the Receive Carrier. A Mark is a high level.

Receive Break (Rx Brk)

Upon receipt of a continuous 150 ms space, the modem automatically clamps the Receive Break output high. This output is also clamped high until Clear-to-Send is established.

Digital Carrier (FO)

A test signal output is provided to decrease the chip test time. The signal is a square wave at the transmit frequency.

Transmit Carrier (Tx Car)

The Transmit Carrier is a digitally-synthesized sine wave (Figure 9) derived from the 1.0 MHz crystal reference. The frequency characteristics are as follows:

Mode	Data	Transmit Frequency	Tolerance*
Originate	Mark	1270 Hz	−0.15 Hz
Originate	Space	1070 Hz	0.90 Hz
Answer	Mark	2225 Hz	−0.31 Hz
Answer	Space	2025 Hz	−0.71 Hz

*The reference frequency tolerance is not included.

The proper output frequency is transmitted within 3.0 μs following a data bit change with no more than 2.0 μs phase discontinuity. The typical output level is 0.35 V (RMS) into 100 k ohm load impedance.

The second harmonic is typically 32 dB below the fundamental (see Figure 10).

POWER-ON RESET

Power-on reset is provided on-chip to insure that when power is first applied the Answer Phone output is in the low (inactive) state. This holds the modem in the inactive or idle mode until a $\overline{SH}$ or $\overline{RI}$ signal has been applied. Once power has been applied, a momentary loss of power at a later time may not be of sufficient time to guarantee a chip reset through the power-on reset circuit.

To insure initial power-on reset action, the external parasitic capacitance on $\overline{RI}$ and $\overline{SH}$ should be <30 pF. Capacitance values >30 pF may require the use of an external pullup resistor to V_{CC} on these inputs in addition to the pullup devices already provided on chip.

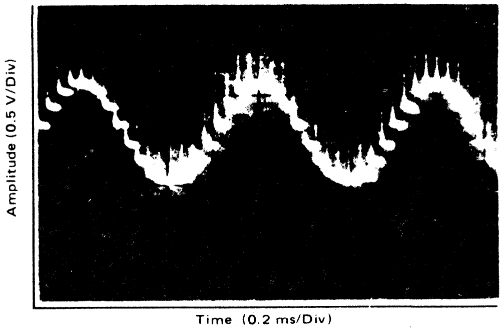


FIGURE 9 — TRANSMIT CARRIER SINE WAVE

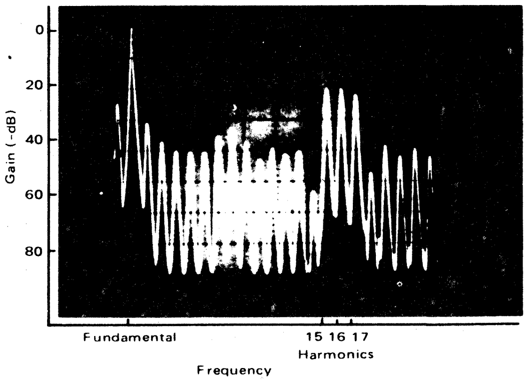


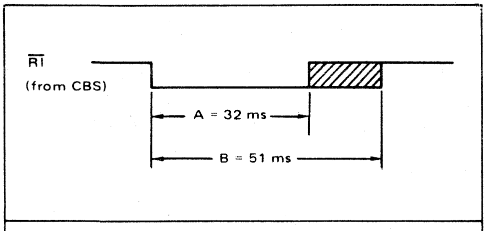
FIGURE 10 — TRANSMIT CARRIER FREQUENCY SPECTRUM

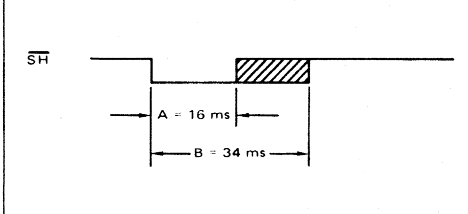
TABLE 1 – ASYNCHRONOUS INPUT PULSE WIDTH AND OUTPUT DELAY VARIATIONS
(Time delays specified do not include the 1-MHz reference tolerance.)

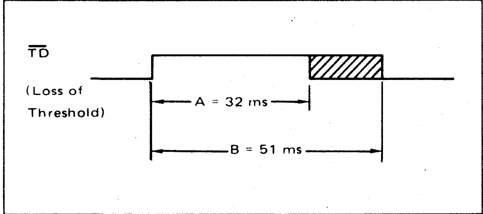
Due to the asynchronous nature of the input signals with respect to the circuit internal clock, a delay variation or input pulse width requirement will exist. Time delay A is the maximum time for which no response will occur. Time delay B is the minimum time required to guarantee an input response. Input signal widths in the cross-hatched region (i.e., greater than A but less than B) may or may not be recognized as valid.

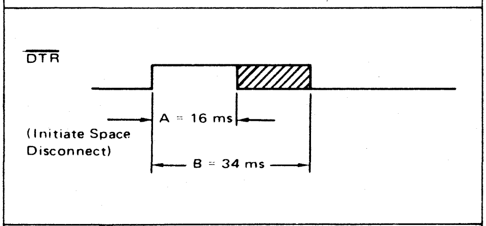
For output delays, time A is the minimum delay before an output will respond. Time B is the maximum delay for an output to respond. Output signal response may or may not occur in the cross-hatched region (i.e., greater than A but less than B).

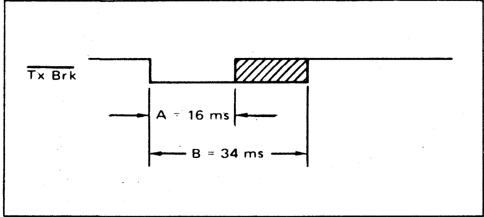
INPUT PULSES



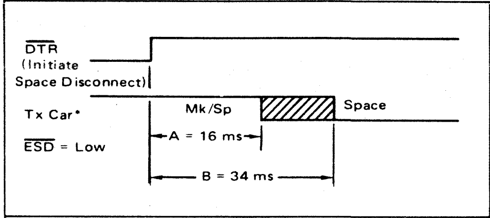


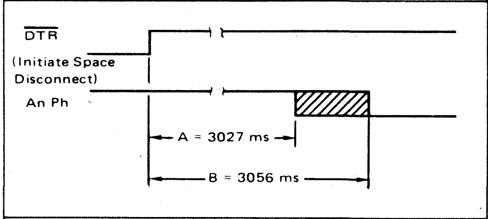


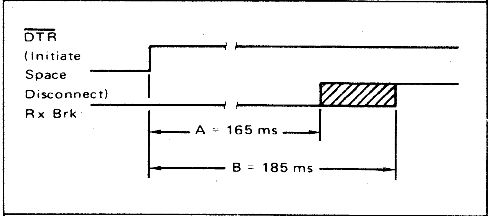


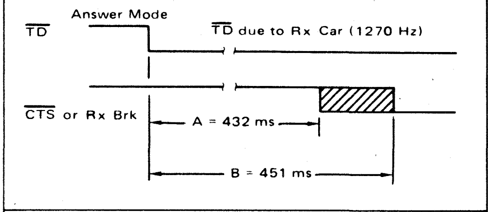


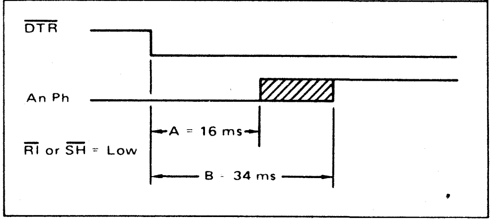
OUTPUT DELAYS







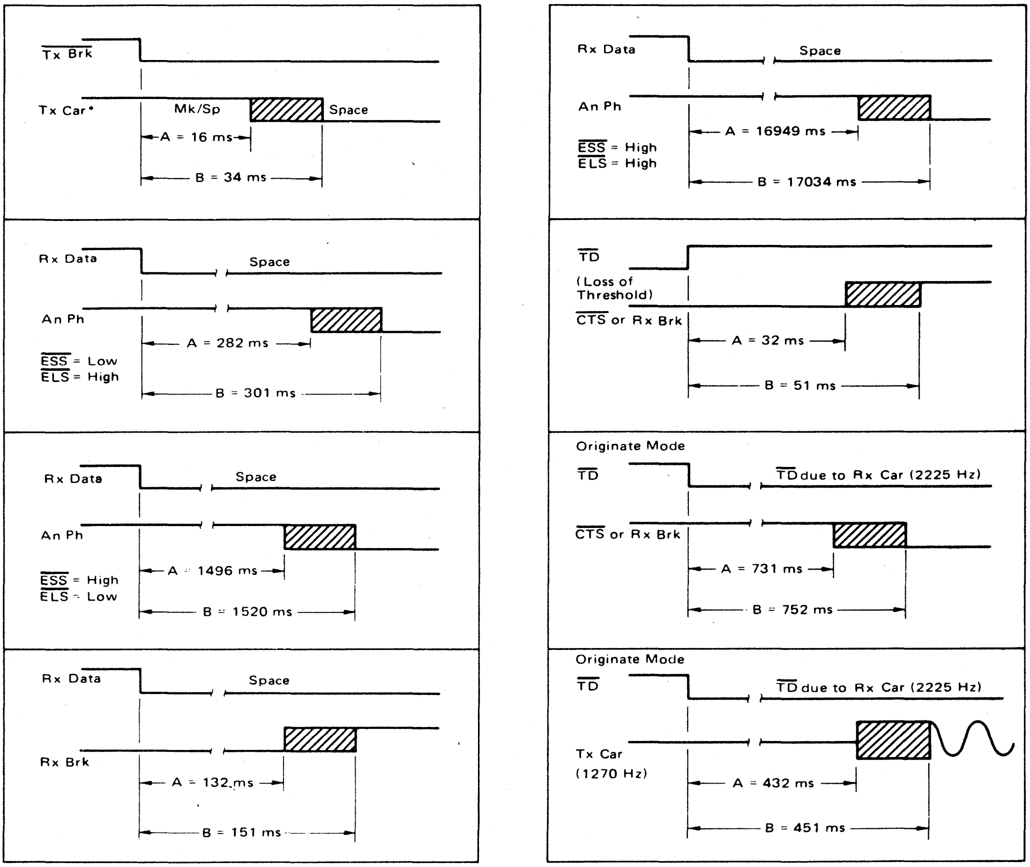




*Digital Representation.

(continued)

TABLE 1 – OUTPUT DELAY VARIATIONS (continued)



*Digital Representation

TABLE 2 – TRANSMIT BREAK AND DISCONNECT DELAYS

Function Description	Min	Max	Unit
$\overline{\text{Tx Brk}}$ (Space Duration)	232	235	ms
Space Disconnect (Space Duration) ($\overline{\text{DTR}}$ = High, $\overline{\text{ESD}}$ and $\overline{\text{TD}}$ = Low)	3010	3023	ms
Loss of Carrier Disconnect (Measured from positive edge of $\overline{\text{CTS}}$ to negative edge of An Ph, with $\overline{\text{RI}}$, $\overline{\text{SH}}$, and $\overline{\text{TD}}$ = High)	16965	17034	ms
Override Disconnect (Measured from positive edge of $\overline{\text{RI}}$ or $\overline{\text{SH}}$ to negative edge of An Ph, with $\overline{\text{TD}}$ = High)	16916	17101	ms

FIGURE 11 — FLOW DIAGRAM

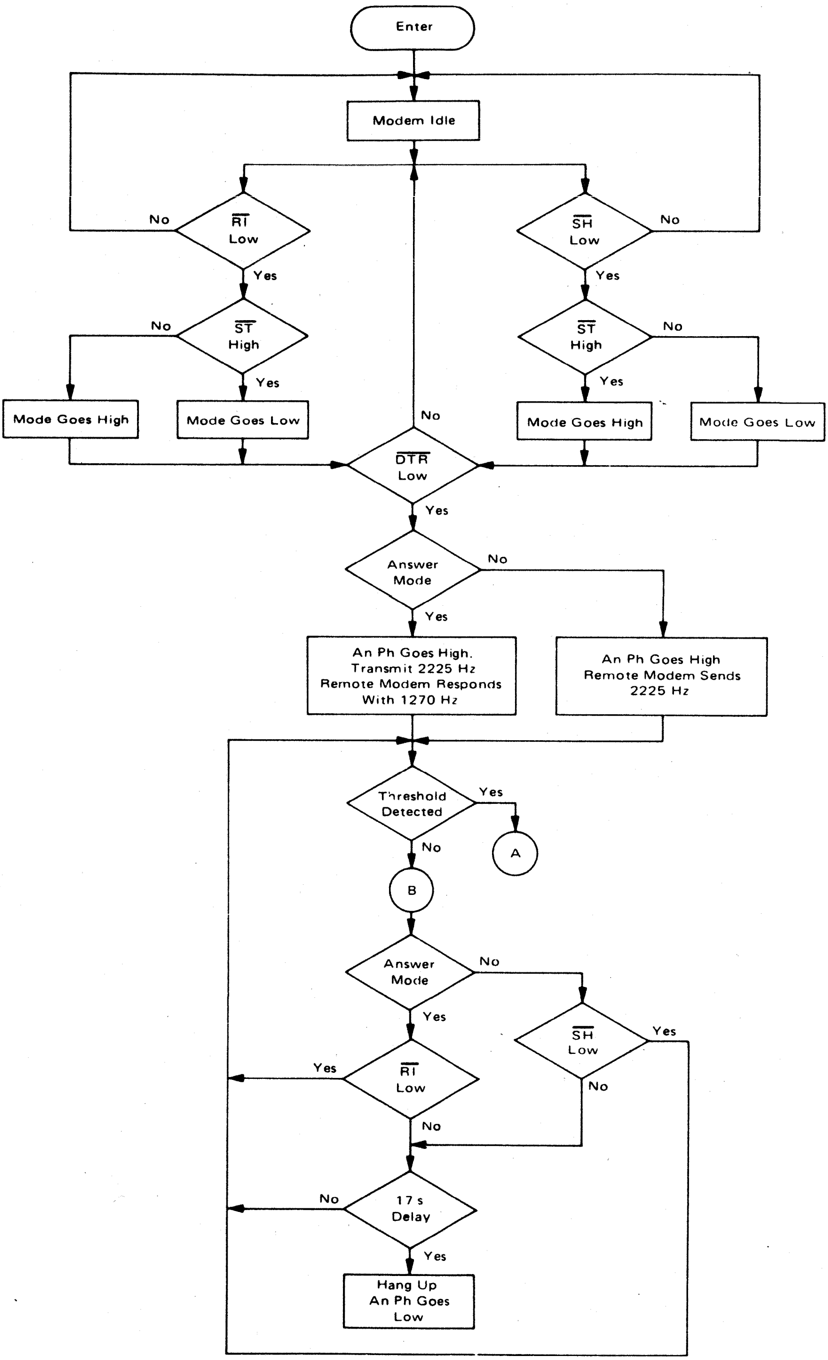
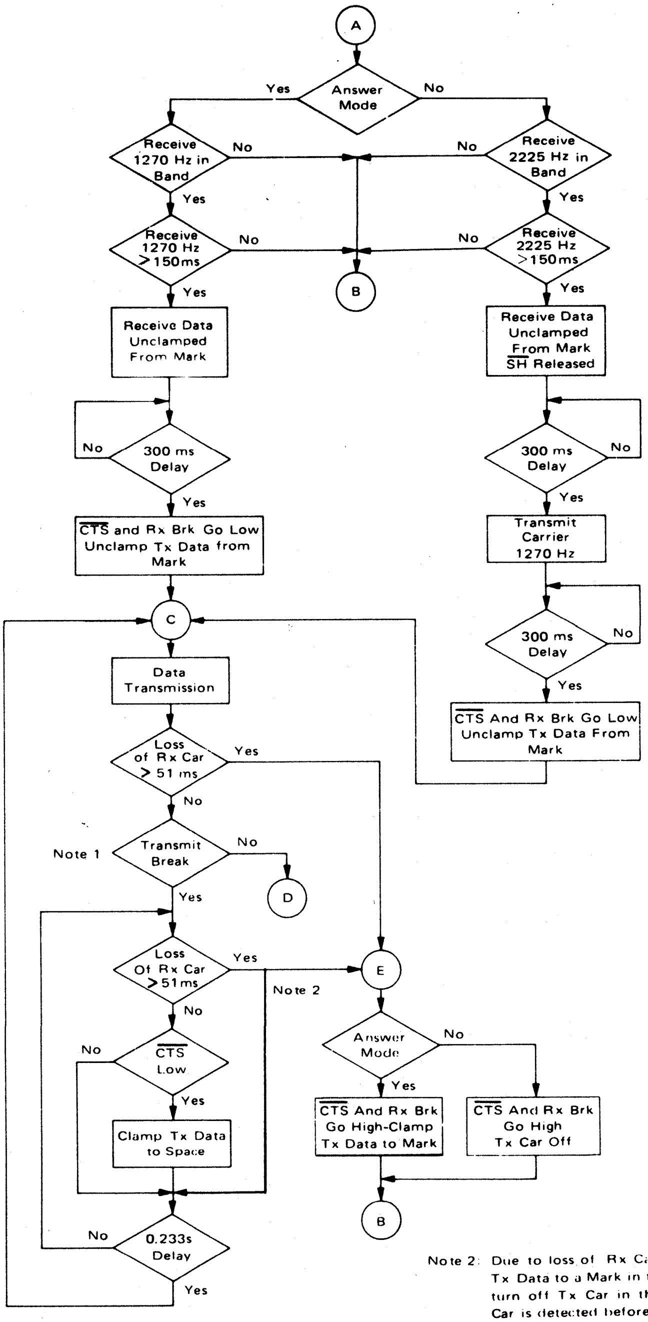


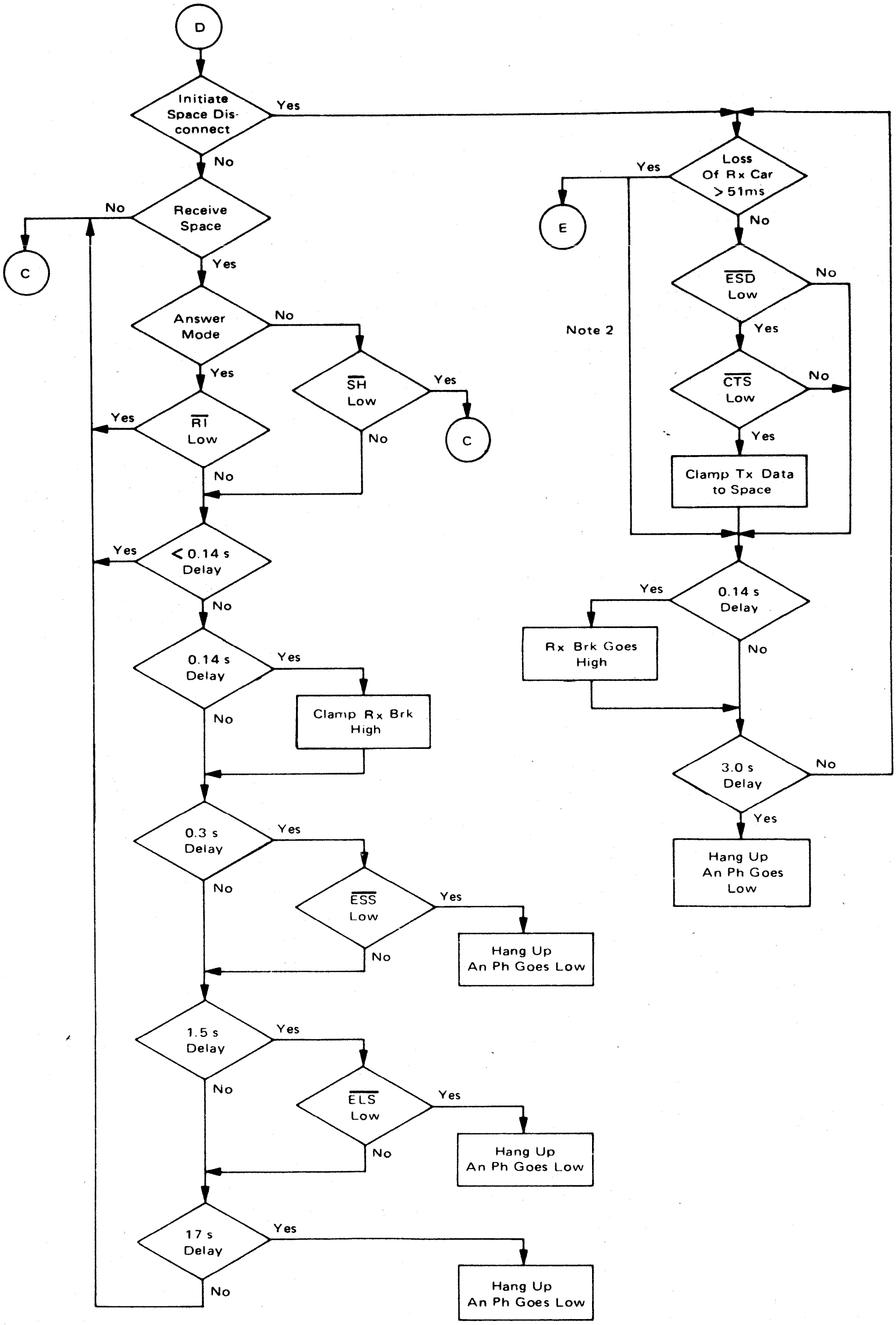
FIGURE 11 — FLOW DIAGRAM (CONTINUED)



Note 1: Transmit Break, Initiate Space Disconnect, and Receive Space are mutually exclusive events.

Note 2: Due to loss of Rx Car, the modem will clamp Tx Data to a Mark in the Answer Mode and will turn off Tx Car in the Originate Mode. If Rx Car is detected before completion of Tx Brk or Initiate Space Disconnect, normal operation of Tx Brk or Initiate Space Disconnect will continue until completion of their respective time delays.

FIGURE 11 — FLOW DIAGRAM (CONCLUDED)



MC6862

2400 bps DIGITAL MODULATOR

The MC6862 is a MOS subsystem designed to be integrated into a wide range of equipment utilizing serial data communication.

The modulator provides the necessary modulation and control functions to implement a serial data communication link over a voice grade channel, utilizing differential phase shift keying (DKSP) at bit rates of 1200 or 2400 bps. Phase options are provided for both the U.S. and international markets. The MC6862 can be implemented into a wide range of data handling systems, including stand-alone modems, data storage devices, remote data commuication terminals, and I/O interfaces for counters.

N-channel silicon-gate technology permits the MC6862 to operate using a single voltage supply and be fully TTL compatible.

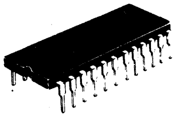
The modulator is compatible with the M6800 microcomputer family, and provides medium-speed data communications capability.

- Clear-to-Send Delay Options
- 511-Bit CCITT Test Pattern
- Terminal Interfaces are TTL Compatible
- Compatible Functions for 201B/C Data Sets
- CCITT and U.S. Phase Options
- 1200/2400 bps Operation
- Answer-Back Tone

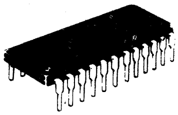
MOS

(N-CHANNEL, SILICON-GATE)

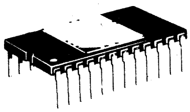
2400 bps
MODULATOR



S SUFFIX
CERDIP PACKAGE
CASE 623

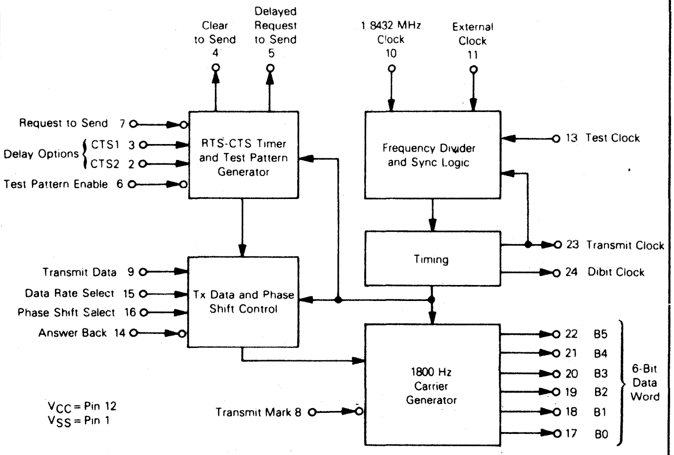


P SUFFIX
PLASTIC PACKAGE
CASE 709

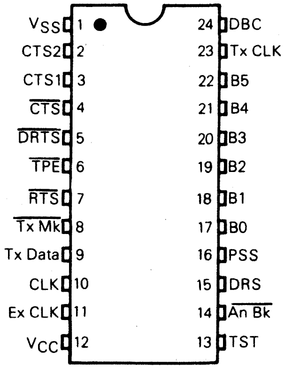


L SUFFIX
CERAMIC PACKAGE
CASE 716

BLOCK DIAGRAM



PIN ASSIGNMENT



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 MC6862 MC6862S, MC6862C	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 Ceramic Package Plastic Package Cerdip Package	θ_{JA}	 60 120 65	°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. 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:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

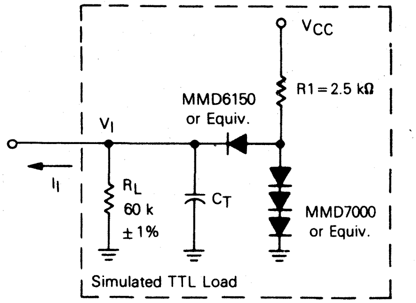
DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 ± 0.25 Vdc, V_{SS} = 0, $T_A = T_L$ to T_H, all outputs loaded as shown in Figure 1 unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	V_{IL}	V_{SS}	–	$V_{SS}+0.8$	V
Input Current ($V_{in}=V_{SS}$) CTS1, CTS2, PSS, DRS, $\overline{\text{An Bk}}$, and $\overline{\text{Tx MK}}$ $\overline{\text{RTS}}$ and $\overline{\text{TPE}}$	I_{in}	– –	– –	−0.2 −1.6	mA
Input Leakage Current (V_{in} = 5.25 V, $V_{CC}=V_{SS}$)	I_{IL}	–	–	2.5	µA
Output High Voltage (I_{OH} = −0.04 mA, Load A) (I_{OH} = 0.0 mA, Load B)	 V_{OH1} V_{OH2}	 $V_{SS}+2.4$ $V_{CC}-0.5$ V	 – –	 V_{CC} V_{CC}	V
Output Low Voltage (I_{OL} = 1.6 mA, Load A)	V_{OL}	V_{SS}	–	$V_{SS}+0.4$	V
Input Capacitance (f = 0.1 MHz, T_A = 25°C)	C_{in}	–	5.0	–	pF
Internal Power Dissipation (Measured at $T_A=T_L$) (All inputs at V_{SS} except Pin 13 = 57.6 kHz and ALL outputs open)	P_{INT}	–	210	315	mW
Input Transition Times, All Inputs Except 1.8432 MHz Input (From 10% to 90% points)	t_r, t_f	–	–	1.0*	µs
Input Transition Times, 1.8432 MHz Input (From 0.8 V to 2.0 V)	t_r, t_f	–	–	40	ns
Input Clock Duty Cycle, 1.8432 MHz Input (Measured at 1.5 V level)	D.C.	30	–	70	%
Tx Data Setup Time (Figure 2)	t_S	35	–	–	µs
Tx Data Hold Time (Figure 2)	t_H	35	–	–	µs
Output Transition Times	t_r, t_f	–	–	5.0	µs

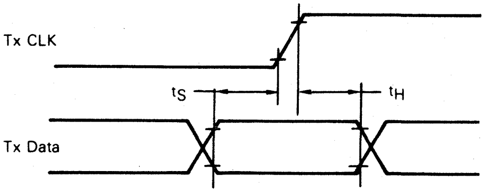
*Maximum Input Transition Times are ≤ 0.1 × Pulse Width or the specified maximum of 1.0 µs, whichever is smaller.

FIGURE 1 — OUTPUT TEST LOAD



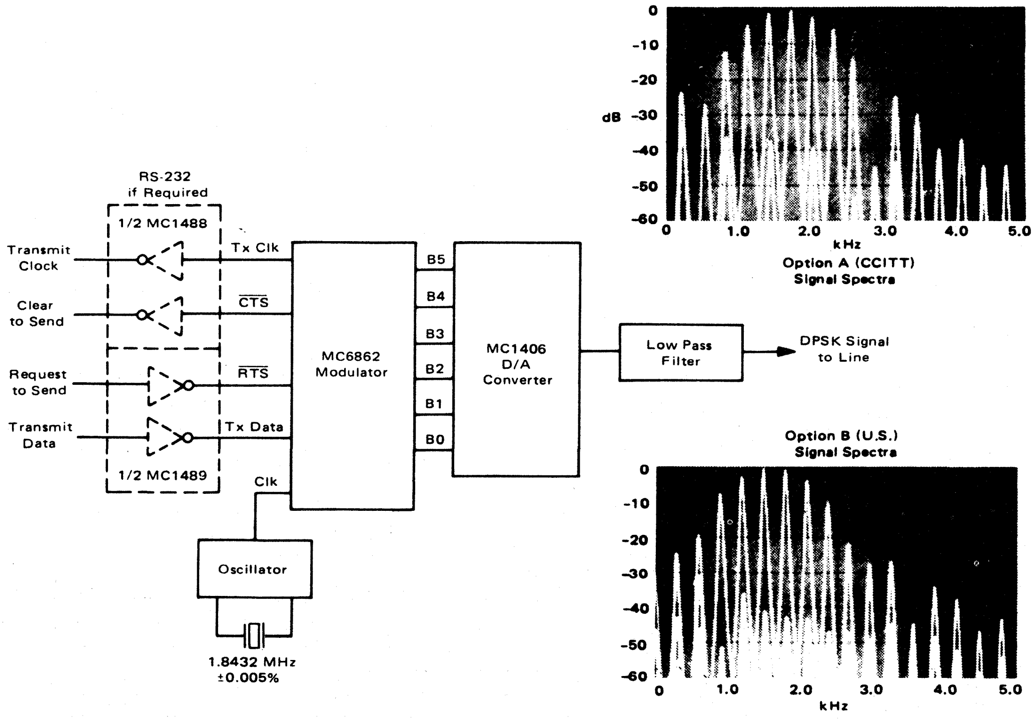
C_T = 20 pF = total parasitic capacitance, which includes probe, wiring, and load capacitances.

FIGURE 2 — TRANSMIT DATA SETUP AND HOLD TIME



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

FIGURE 3 — 2400 bps MODULATOR INTERFACE



Option A (CCITT) Signal Spectra

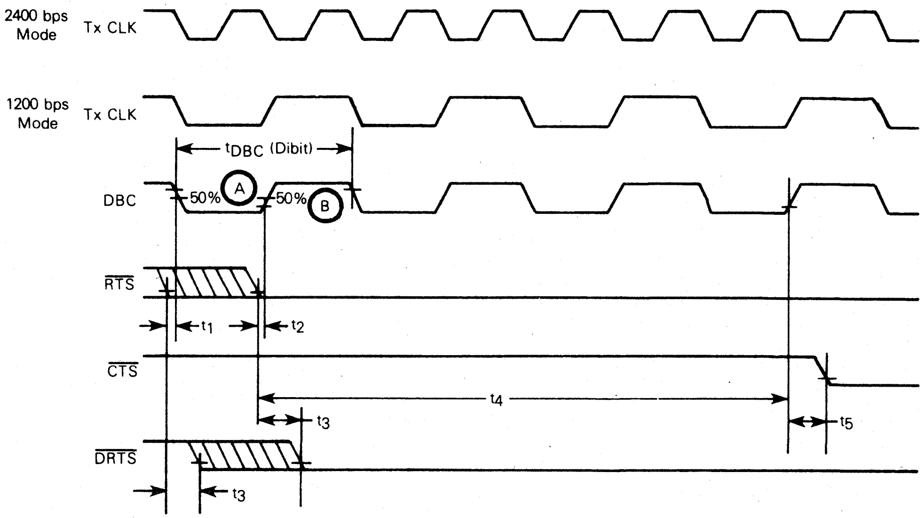
Option B (U.S.) Signal Spectra

DELAY TIMINGS (See Figures 4 and 5)

Characteristic	Symbol	Min	Typ	Max	Unit
$\overline{\text{RTS}}$ to DBC Delay	t_1	–	–	8	µs
DBC to $\overline{\text{RTS}}$ Delay	t_2	45	–	–	µs
$\overline{\text{RTS}}$-$\overline{\text{DRTS}}$ Delay	t_3	–	–	35	µs
$\overline{\text{RTS}}$-$\overline{\text{CTS}}$ Delay					
CTS1 = 0, CTS2 = 1	t_4*	0	–	35	µs
CTS1 = 1, CTS2 = 0		8.55	–	9.35	ms
CTS1 = 1, CTS2 = 1		24.9	–	26.4	ms
CTS1 = 0, CTS2 = 0		147.0	–	154.0	ms
$\overline{\text{CTS}}$-DBC Delay					
CTS1 = 1, CTS2 = 0	t_5	–	–	35	µs
CTS1 = 1, CTS2 = 1		–	–	35	
CTS1 = 0, CTS2 = 0		–	–	35	
$\overline{\text{RTS}}$ to $\overline{\text{CTS}}$ Low	t_6	–	–	1.60	ms
$\overline{\text{RTS}}$ Min Delay	t_7	–	–	1.67	ms
DBC to $\overline{\text{DRTS}}$ Delay	t_8	–	–	35	µs
DBC Cycle Time	t_{DBC}	833.28	833.33	833.37	µs

*The reference frequency tolerance is not included.

FIGURE 4 — $\overline{\text{RTS}}$-$\overline{\text{CTS}}$ AND $\overline{\text{RTS}}$-$\overline{\text{DRTS}}$ DELAYS



$\overline{\text{RTS}}$-$\overline{\text{CTS}}$ delay options are selected by the CTS1 and CTS2 inputs, and are stated as time delay interval t_4. An $\overline{\text{RTS}}$ input signal synchronized about point A will synchronize $\overline{\text{CTS}}$ with the positive transition of DBC (Dibit Clock). Delay t_4 is measured with respect to the negative transition of $\overline{\text{RTS}}$.

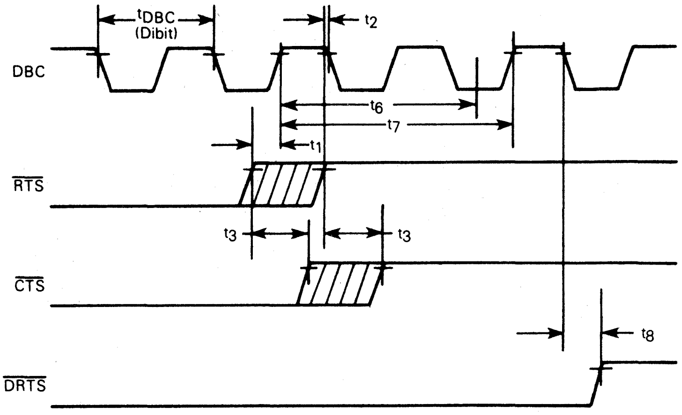
$\overline{\text{RTS}}$ signals synchronized with the positive transition of DBC (point B), will result in the same $\overline{\text{CTS}}$ delay (t_4). For this case the negative transition of $\overline{\text{CTS}}$ is synchronized with the negative transition of DBC with delay t_4 measured with respect to the negative transition of $\overline{\text{RTS}}$.

$\overline{\text{DRTS}}$ will go low within t_3 of the negative transition of $\overline{\text{RTS}}$. With the exception of the no-delay option, $\overline{\text{CTS}}$ will go low within t_5 of the positive transition of DBC, following the t_4 delay selected. This applies when $\overline{\text{RTS}}$ is synchronized to Point A as shown.

If $\overline{\text{RTS}}$ goes high and remains high ≥ 20 µs within time interval t_4, a reset of the internal $\overline{\text{RTS}}$-$\overline{\text{CTS}}$ timer function will occur. If $\overline{\text{RTS}}$ goes high for less than 20 µs, the circuit may or may not respond to this momentary loss of the $\overline{\text{RTS}}$ signal.

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 — LOSS OF $\overline{RTS}$ TO $\overline{DRTS}$ DELAY



A positive transition of $\overline{RTS}$ after $\overline{CTS}$ has become active can result in different functional characteristics of the $\overline{CTS}$ and $\overline{DRTS}$ output signals, depending on the time duration that $\overline{RTS}$ remains inactive.

Under all conditions, $\overline{CTS}$ will go high within t_3 following a positive transition of $\overline{RTS}$. If $\overline{RTS}$ goes high in the shaded region shown (i.e., synchronized to the positive transition of DBC) and remains high beyond the time interval defined as t_7, then $\overline{DRTS}$ will go high within t_8 of the next negative transition of DBC. If $\overline{RTS}$ were to go low after t_7, the $\overline{RTS}$-$\overline{CTS}$ delay times given in Figure 4 will result.

If $\overline{RTS}$ goes high in the shaded region shown, and then returns low within time interval t_6, the negative transition of $\overline{CTS}$ will follow within 35 μs, and $\overline{DRTS}$ will remain in the active or low state. Under these conditions, the normal $\overline{RTS}$-$\overline{CTS}$ delay times are not encountered when $\overline{RTS}$ is reactivated. If $\overline{RTS}$ goes low for less than 20 μs, the circuit may or may not respond

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.

DEVICE OPERATION

GENERAL

Figure 3 shows the modulator and its intra-connections. The data to be transmitted is presented in synchronous serial format to the modulator for conversion to DPSK signals used in transmission. The modulator output is digital; therefore, a D/A converter and a filter transform the signal to an analog form.

The control functions provide four different Clear-to-Send delay options. An Answer-Back tone is available for automatic answering applications. The modulator has a built-in 511-bit pseudorandom pattern generator for use in system diagnostic tests.

INPUT/OUTPUT FUNCTIONS

Request to Send ($\overline{RTS}$)

The $\overline{RTS}$ signal from the data terminal controls transmission from the modulator. A low level on $\overline{RTS}$ activates the modulator data output. A constant mark, for synchronization, is sent during the $\overline{RTS}$ to $\overline{CTS}$ delay interval. Termination of the transmission is accomplished by taking $\overline{RTS}$ high (see Figures 4 and 5).

Delayed Request to Send ($\overline{DRTS}$)

This output can be used to control transmission as specified by the Transmit Mark control input. $\overline{DRTS}$ follows the negative transition of $\overline{RTS}$, and goes negative within t_3 of the negative transition of $\overline{RTS}$ (Figure 4). The delay from a positive transition of $\overline{RTS}$ to a positive transition of $\overline{DRTS}$ is shown in Figure 5. The $\overline{DRTS}$ delay allows data within the modulator to be transmitted before transmission is inhibited.

Clear to Send ($\overline{CTS}$)

$\overline{CTS}$ follows $\overline{RTS}$ to both the logic 0 and logic 1 levels. The delay from a negative transition of $\overline{RTS}$ to a negative $\overline{CTS}$ transition is selectable by external strapping of CTS1 and CTS2. The delay from a positive transition of $\overline{RTS}$ to a positive $\overline{CTS}$ transition is less than t_4.

$\overline{CTS}$ will go low within t_5 after the positive transition of the Dibit Clock (see Figure 4) except when the non-delay option is selected. For the no-delay option, $\overline{CTS}$ follows $\overline{RTS}$ within t_5.

$\overline{RTS}$-$\overline{CTS}$ Delay Options (CTS1, CTS2)

The $\overline{RTS}$-$\overline{CTS}$ delays are selectable according to the following strapping options

$\overline{RTS}$-$\overline{CTS}$ Delay	CTS1	CTS2
0.0 + 0.035 ms, − 0.0 ms	0	1
8.55 to 9.35 ms	1	0
24.90 to 26.4 ms	1	1
147.0 to 154.0 ms	0	0

Transmit Mark (Tx Mk)

The Transmit Mark control allows the system designer to select whether the Delayed Request to Send activitates and deactivates the transmission on the modulator chip or off the chip in the output amplifier.

When Tx Mk is high, transmission is controlled on the modulator chip, and occurs from the chip only when $\overline{\text{DRTS}}$ or Answer Back is in the logic 0 state (see Figure 6).

When Tx Mk is low, transmission is controlled off the modulator chip. In this mode, the modulator chip transmits marks at all times except when data or an Answer-Back tone is being transmitted (see Figure 6).

Test Pattern Enable ($\overline{\text{TPE}}$)

A 511-bit test pattern generator is contained on the modulator chip. This pattern is in accord with CCITT specification V52.

The 511-bit test pattern is activated by applying a logic 0 to $\overline{\text{TPE}}$. A mark (logic 1) condition on the Transmit Data input with $\overline{\text{TPE}}$ activated (logic 0) causes the test pattern to appear at the data output. A space (logic 0) condition on Tx Data with $\overline{\text{TPE}}$ activated causes the test pattern data to appear inverted at the data output.

Although the Motorola 2400 bps modulator contains a CCITT 511 test pattern generator it does not incorporate the 511 data randomizer or scrambler.

Random data applied to Tx data with TPE activated causes the test pattern data to be scrambled (exclusive NORed) with the data, and the result appears at the data output.

The MC6863 demodulator does contain a built-in data descrambler, which is enabled by $\overline{\text{TPE}}$ input going active. To scramble data using the modulator, the circuit in Figure 7 must precede the TX Data input of the modulator. Tx Data is added to the scrambler output pattern. Then the data is delayed by a full data bit before being transmitted by the modem. This assures a proper Transmit Data/Transmit Clock phase relationship.

If the data scrambler is to be an optional feature, then the transmit data multiplexer would also have to be built. This is selected by the Test Pattern Enable signal or any other signal that is found suitable.

The scrambling of data in the data comm environment is not done in an attempt to encrypt information in the normal sense of the word. Rather, the purpose of the scrambling of data is to guarantee that with respect to the modem carrier, there is always random data on the line with little chance for a long string of ones or zeros to exist. This is particularly important if an adaptive equalizer is being incorporated at the demodulator. The adaptive equalizer will require reasonably evenly distributed data to optimize its statistical response to the incoming signal. The normally used code is the CCITT 511 sequence which is EXOR'd with data.

The test pattern generator can be enabled only when $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ are logic 0. If $\overline{\text{TPE}}$ is activated outside this time interval, the previously stated $\overline{\text{RTS}}$-$\overline{\text{CTS}}$ and $\overline{\text{RTS}}$-$\overline{\text{DRTS}}$ delays, shown in Figures 4 and 5, are not valid.

Data-Rate Select (DRS)

The modulator can transmit at either 2400 bps or 1200 bps. Both data rates utilize an 1800 Hz carrier signal and employ phase shifting at 1200 Hz. The 2400 bps rate is obtained by encoding two bits of data into each phase shift. The 2400 Hz rate is selected by applying a logic 1 to the Data-Rate Select lead. The 1200 Hz rate is selected by applying a logic 0 to DRS.

Phase-Shift Select (PSS)

Option A (CCITT) or Option B (U.S.) phase shift can be selected for 2400 bps operation. The input data format and phase shift relationship for these two options are as follows:

Data	PSS = 0 Option A*	PSS = 1 Option B
00	0°	+45°
01	+90°	+135°
11	+180°	+225°
10	+270°	+315°

*See example Figure 8.

FIGURE 6 — TRANSMIT MARK CONTROL

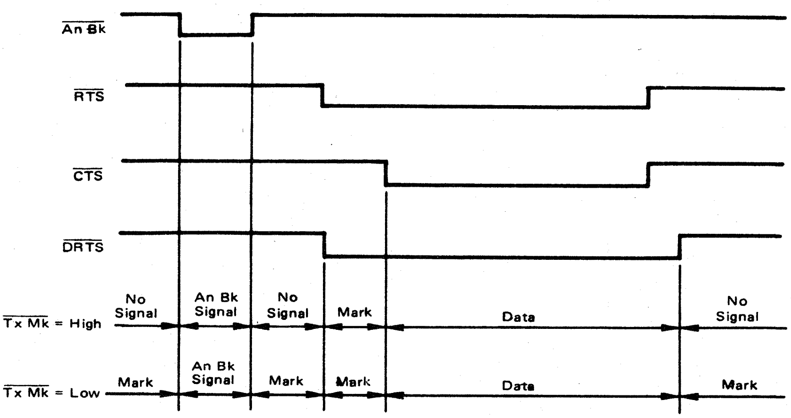


FIGURE 7 — MODULATOR CCITT 511 DATA SCRAMBLER

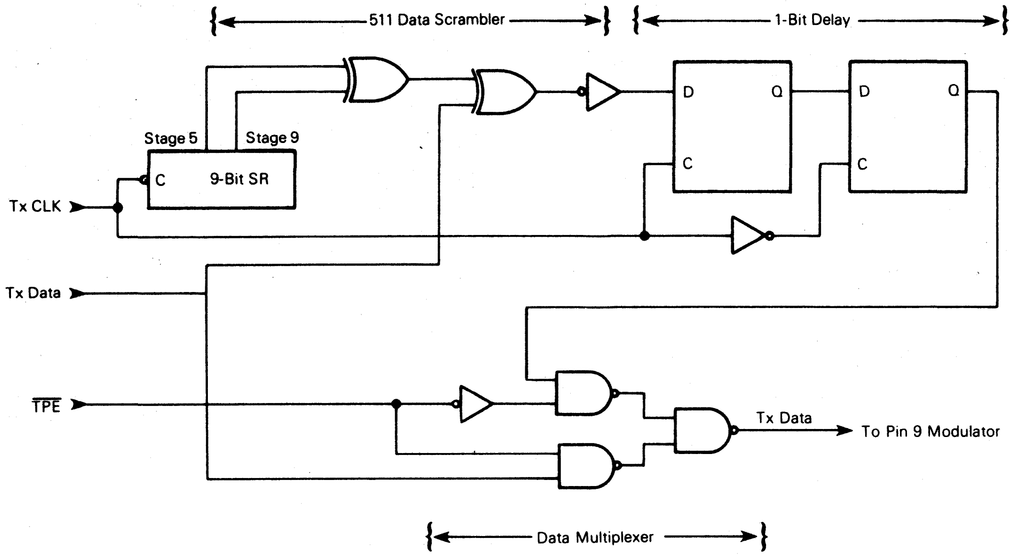
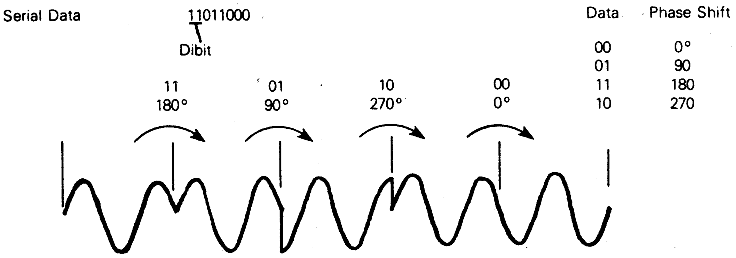


FIGURE 8 — EXAMPLE-CARRIER PHASE SHIFTS FOR OPTION A



For 1200 bps operation, Option C (CCITT) or Option D (U.S.) phase shift can be selected:

Data	PSS = 0 Option C	PSS = 1 Option D
0	+90°	+45°
1	+270°	+225°

Option C is selected by applying a logic 0 to the Phase Shift Select lead when the Data Rate Select lead is strapped for 1200 bps operation (logic 0). Option D is selected by applying a logic 1 to PSS with DRS at logic 0. The phase shifts shown are the difference in phase between the signal at the end of one dibit period and the new signal at the beginning of the next dibit.

Transmit Data (Tx Data)

Transmit Data is the serial binary information presented for DPSK modulation. A high level represents a mark. For timing, see Transmit Clock (Figure 4).

Transmit Clock (Tx CLK)

A 2400/1200 Hz Transmit Clock output is provided for the communication terminal. The Transmit Data signal is sampled on the positive transition of Transmit Clock. The Transmit Data to Transmit Clock setup and hold time requirements are shown in the Electrical Characteristics Table and in Figure 2.

Dibit Clock (DBC)

A 1200 Hz Dibit Clock identifies the modulation timing. This signal goes negative less than 100 μs prior to the start of dibit modulation.

External Clock (Ex CLK)

A 2400/1200 Hz clock signal applied to the External Clock lead causes Transmit Clock to be synchronized with Ex CLK. This input must have an accuracy within ±0.005%.

When no transitions occur on this input, the internal clock provides the 2400/1200 Hz transmit timing signal. Fast synchronization of Tx CLK to Ex CLK is not provided on the chip. *When Ex CLK is not used, it should be tied to either the logic 0 or logic 1 state.*

1.8432 MHz (CLK)

This input must be a square wave with rise and fall times of less than 40 ns and a 50 ±20% duty cycle. The clock accuracy must be written ±0.005%.

Answer Back ($\overline{\text{An Bk}}$)

A logic 0 level applied to Answer Back causes a 2025 Hz carrier to be generated on the modulator chip instead of a phase shifted 1800 Hz carrier. A logic 1 level applied to $\overline{\text{An Bk}}$ enables the modulator to generate the normal phase shifted 1800 Hz carrier signal, as shown in Figure 6. The time delay from a transition on $\overline{\text{An Bk}}$ to the appropriate signal at the modulator chip output is less than 2 ms.

Activation of $\overline{\text{An Bk}}$ (a logic 0) will disable all other operation modes including the $\overline{\text{Tx Mk}}$ function, and will reset $\overline{\text{CTS}}$ to an inactive state along with the $\overline{\text{RTX}}$-$\overline{\text{CTS}}$ internal timer. $\overline{\text{An Bk}}$ should therefore be activated only before initiating $\overline{\text{RTS}}$ or after loss of the $\overline{\text{DRTS}}$ output signal. The combination of a logic 0 on $\overline{\text{An Bk}}$ with a logic 0 on $\overline{\text{TPE}}$ is not used in normal system operation, and hence is used as a reset input during device test.

Digital Output (B0-B5)

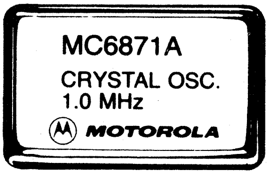
These outputs are designed to interface with a 6-bit digital-to-analog converter. The resultant signal out of the D/A is the differential phase shift keyed signal quantized at a 14.4 kHz rate. A low-pass filter can then be used to smooth the data transitions. B0 is the least-significant bit, and the positive level the active state.

Test Clock (TST)

A test signal input is provided to decrease test time of the chip. *In normal operation this input must be strapped low.*

MOTOROLA

MC6870, MC6871 series



actual size

Two-Phase Microprocessor Clocks

Designed to drive the Motorola MC6800 MPU

The *Functional Module* approach to data communications hardware design significantly decreases the time between the "idea" stage and the marketable product.

A fundamental building block in a modular microcomputer system is the 2-phase clock oscillator used to drive the microprocessor. Motorola is uniquely qualified to provide this building block because of expertise in the three relevant fields: oscillator design, quartz crystal technology, and thick film hybrid integrated circuit manufacturing.

This one-of-a-kind expertise has created several clocks designed to drive Motorola's MC6800 Microprocessor. This plug-in unit contains the crystal, the oscillator circuit, the NMOS and TTL drivers, and the waveshaping and interface circuitry; all the components necessary to provide the critical non-overlapping 2-phase waveforms used by the MC6800 MPU.

FEATURES

Clock Module—Each clock module requires only a single 5 volt power supply. The NMOS outputs can drive highly capacitive loads ranging from 80 pf to 160 pf and meet all MPU input waveshape and timing requirements.

Each TTL output signal leads the ϕ_2 NMOS so that additional system device delays can be accommodated. All TTL outputs are buffered so they can drive 5 TTL devices and maintain all output specifications.

Each module is crystal-controlled and is compensated for variations in temperature, voltage, and load. The standard frequency of each model is 1 MHz; however, other frequencies between 250 kHz and 2.5 MHz can be ordered.

Reliability—Decreased Component Count—Thick film hybrids offer a reliability advantage that comes primarily from reduced component count and therefore reduced interconnections. Further, the single hermetic seal on the hybrid package reduces the failure rate whereas in a discrete design a separate sealing process with an associated failure rate is needed for each component.

High Density Packaging—The hybrid MPU clock allows compact microcomputer design. It takes up only 1.34"x .840" space and has a seated height of .200".

Ruggedized Design—Maximum reliability at minimum cost is the result of combining three of Motorola's fields of experience: quartz crystal technology, clock oscillator design, and thick film hybrid integrated circuit manufacturing. Mass automated production techniques assure volume production. Gold plating of all crystals and Class 100 clean room processing testify that no short cuts are taken that might diminish reliability. Environmental testing proves the effectiveness of the rugged design for those applications in which shock and vibration are likely hazards.

Complete Process Control—Motorola is the only totally integrated manufacturer of quartz frequency control devices; full control of all processes from growing, sawing, lapping, and finishing quartz to combining it with other components into an electronic product—the MC6870A, MC6871A, and MC6871B MPU clocks.

Volume Production—Production facilities are oriented to mass automated production techniques. And, if required, capital for expansion is available to meet even greater requirements.

environmental specifications

Temperature Cycle: ±5 ppm max., 0 to 120°C, 3 cycles, 2 hrs. max. each, 25 ±2°C ref.

Shock: 1000G's 0.35 millisec, ½ sine wave, 3 shocks each plane

Vibration: 10-55 Hz, .060" D.A.; 55-2000Hz, 35 G's. Duration Time—12 Hours

Humidity: 85% Rel. Humidity, @ +85°C, 250 Hours

mechanical specifications

Gross Leak Test: All units 100% leak tested in de-ionized H_2O.

Hermetic Sealed Package: Mass spectrometer leak rate less than 2×10^{-8} atmos. cc/sec. of helium.

Seal Strengh: 20 lbs. max. force perpendicular to top and bottom.

Pin Material: Phosphor bronze, ¼ hard, Grade A .00003" thick gold flash finish.

Bend Test: Will withstand maximum bend of 90° reference to base for 1 bend.

Marking Ink: Epoxy, heat cured.

Solvent Resistance: Isopropyl Alcohol Tricholoroethane Freon TMC. No marking or seal destruction. Dipped 1 minute @ +25°C ±5°C in solvent.

Note: (1) Unit can be cleaned by only one type solvent listed.

Note: (2) Ultrasonic degreaser not to be used unless frequency and vibration of cleaner specified.

solderability specifications

Materials:

1.1 Solder: 60% tin and 40% lead.

1.2 Flux: The flux shall be 25 percent by weight of Grade WW rosin and 75 percent by weight of 99 percent isopropyl alcohol.

Procedure:

2.1 Solder Bath: The solder bath shall be maintained at 232 ±6°C.

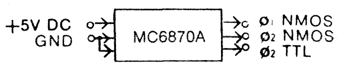
2.2 Solderability: Dip the terminals into the flux to the depth that is to be soldered or to a maximum depth of .025" from the body of the oscillator. Keep them in the flux for at least 5 seconds. Withdraw them from the flux. Dip them immediately into the molten solder to the same depth. Keep them in the molten solder for 2 to 5 seconds. Withdraw them and allow the solder to cool in air.

Requirements:

3.1 The terminals are considered solderable and acceptable for electrical connection purposes if 90 percent of the cold solder surface is uniform and free from breaks and pinholes. The other 10 percent of the cooled solder surface may show only pinholes, voids, or rough spots that are not concentrated in one area.

MC6870A

limited function microprocessor clock
250 kHz to 2.5 MHz



specifications

Rating	Symbol	Value	Unit
Supply Voltage	V_{cc}	5.00±5%	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Power Supply Drain (max.)	I_{pd}	100	mA

ELECTRICAL CHARACTERISTICS (V_{cc} = 5.0 ± 5%, V_{ss} = 0, T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Frequency					
Operating Frequency	f_c	.250		2.5	MHz
Frequency stability (inclusive of calibration tolerance at +25°C, operating temperature, input voltage change, load change, aging, shock and vibration)			±.01		%
NMOS Outputs at 1.0 MHz Operation**					
Pulse Width (meas. at V_{cc}= −.3V dc level)	$T\varnothing_1H$ $T\varnothing_2H$	430 450			ns ns
Logic Levels	V_{OLC} V_{OHC}	V_{ss}-.1 V_{cc}-.3	— —	V_{ss}+.3 V_{cc}+.1	Vdc Vdc
Rise and Fall Times	t_r t_f	5 5	12 12	50 50	ns ns
*Overshoot/Undershoot Logic "1" Logic "0"	V_{OS}	V_{cc}-.5 V_{ss}-.5		V_{cc}+.5 V_{ss}+.5	Vdc Vdc
Pulse duration of any overshoot or undershoot	T_{OS}			40	ns
Period @ 0.3V dc Level	t_{cyc}		1.00		us
Edge Timing @ V_{cc}=0.3V dc	Tx	940			ns
NMOS Relationship @ +0.5V dc Level	t_{d1} t_{d2}	0 0		8.0	us
TTL Outputs					
In ref. to $\varnothing_2$ NMOS @ 0.3V dc					
$\varnothing_2$ TTL @ +1.4V dc	T_A T_H	15 10	30 25	45 40	ns ns
Logic Levels	V_{OH} V_{OL}	2.4	3.2 .3	.4	Vdc Vdc
Rise and Fall Times .4V and 2.4V 2.4V and .4V	t_r t_f			15 15	ns ns
Logic "0" Sink (/Gate)	I_{OL}			−1.6	mA
Logic "1" Source (/Gate)	I_{OH}			+40	uA
Current Output Shorted	I_{SC}	−18		−57	mA
Load					
NMOS—Load Capacity $\varnothing_1$, $\varnothing_2$	C_{NMOS}	80	120	160	pf
TTL—No. of Loads				5	ttl
TTL—Load Capacity	C_{TTL}			50	pf

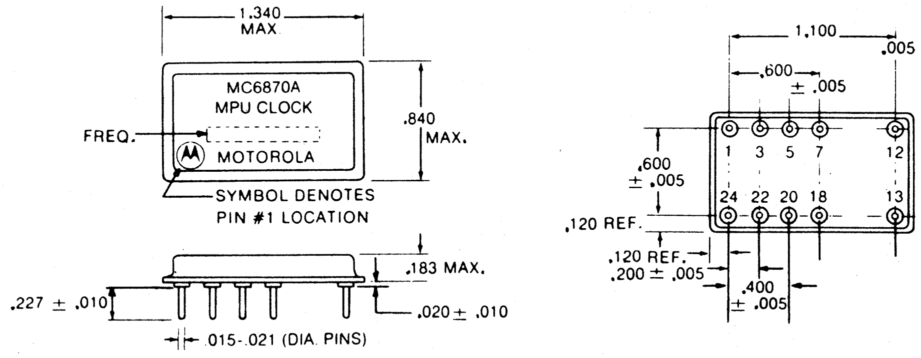
*Into specified test load

**Apply the following parameters for frequencies other than 1.0 MHz:
$T\phi_1H$=0.5 (P-140) ns
$T\phi_2H$=0.5 (P-100) ns
Tx=(P-60) ns
where P=desired period of operation in nanoseconds

PIN	CONNECTION
1	GND
3	NC
5	$\varnothing_2$ TTL
7	V_{cc} (+5VDC)
12	$\varnothing_2$ NMOS
13	$\varnothing_1$ NMOS
18	GND
20	NC
22	NC
24	NC

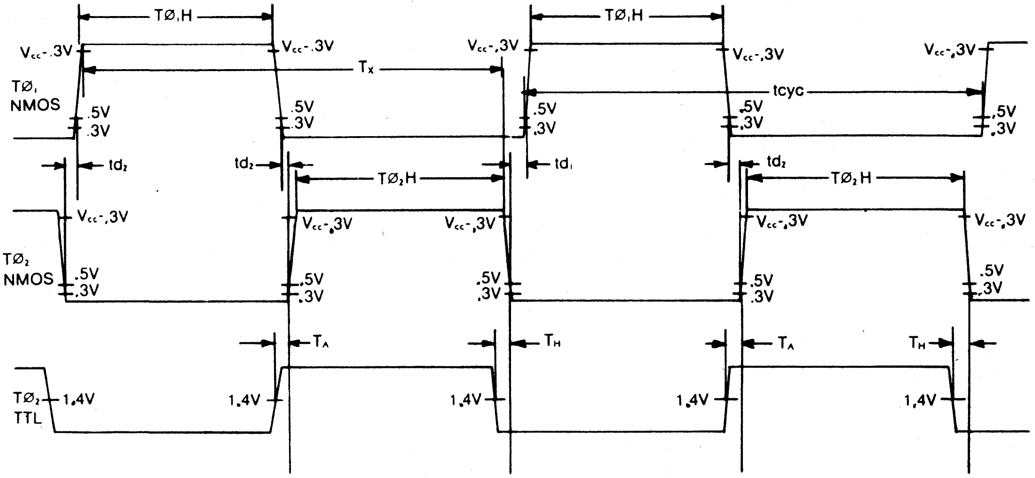
Note: All dimensions are in inches

DIMENSIONS

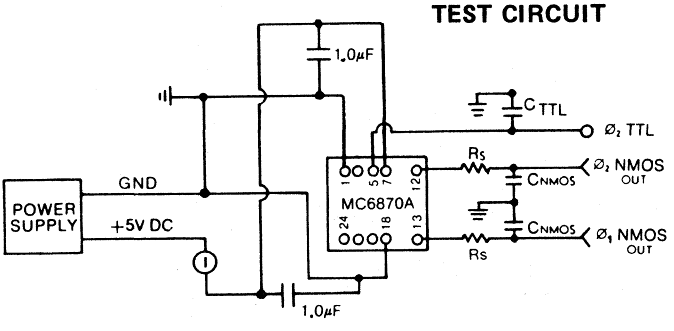


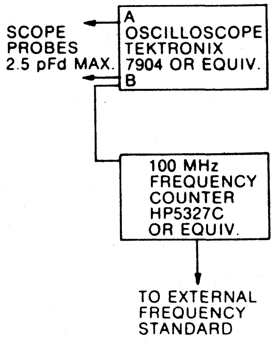
WAVEFORM TIMING

(ALL TIME IN NANOSECONDS)



TEST CIRCUIT





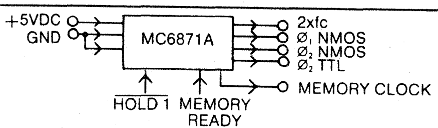
CTTL — MAX CAPACITY 50 pF.

CNMOS — 120 pF ± 40 pF IS THE SPECIFIED MAX. LOAD CAPACITANCE THAT SIMULATES THE MOTOROLA MC6800 MPU INPUT.

Rs—(22Ω) SIMULATES REAL PART OF MPU

MC6871A

full function microprocessor clock
850 kHz to 2.5 MHz



specifications

Rating	Symbol	Value	Unit
Supply Voltage	V_{cc}	5.00±5%	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Power Supply Drain (max.)	I_{pd}	100	mA

ELECTRICAL CHARACTERISTICS (V_{cc} = 5.0 ± 5%, V_{ss} = O,T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Frequency					
Operating Frequency	f_c	.850		2.5	MHz
Frequency stability (inclusive of calibration tolerance at +25°C, operating temperature, input voltage change, load change, aging, shock and vibration)			±.01		%
NMOS Outputs at 1.0 MHz Operation***					
Pulse Width (meas. at V_{cc}= −.3V dc level)	$T\varnothing_1H$ $T\varnothing_2H$	430 450			ns ns
Logic Levels	V_{OLC} V_{OHC}	V_{ss}-.1 V_{cc}-.3	— —	V_{ss}+.3 V_{cc}+.1	Vdc Vdc
Rise and Fall Times	t_r t_f	5 5	12 12	50 50	ns ns
*Overshoot/Undershoot Logic "1" Logic "0"	V_{OS}	V_{cc}-.5 V_{ss}-.5		V_{cc}+.5 V_{ss}+.5	Vdc Vdc
Pulse duration of any overshoot or undershoot	T_{OS}			40	ns
Period @ 0.3V dc Level	t_{cyc}		1.00		us
Edge Timing @ V_{cc}=0.3V dc	Tx	940			ns
NMOS Relationship @ +0.5V dc Level	t_{d1} t_{d2}	0 0		8.0	us
TTL Outputs					
In ref. to $\varnothing_2$ NMOS @ 0.3V dc					
$\varnothing_2$ TTL @ 1.4V dc	T_A T_H	15 10	30 25	45 40	ns ns
Memory Clock @ 1.4V dc	T_C T_J	30 20	50 40	70 60	ns ns
2xfc @ 1.4V dc	T_B	40	80	120	ns
Logic Levels	V_{OH} V_{OL}	2.4	3.2 .3	.4	Vdc Vdc
Rise and Fall Times .4V and 2.4V 2.4V and .4V	t_r t_f			15 15	ns ns
Logic "0" Sink (/Gate)	I_{OL}			−1.6	mA
Logic "1" Source (/Gate)	I_{OH}			+40	uA
Current Output Shorted	I_{SC}	−18		−57	mA
Load					
NMOS—Load Capacity $\varnothing_1$, $\varnothing_2$	C_{NMOS}	80	120	160	pf
TTL—No. of Loads				5	ttl
TTL—Load Capacity	C_{TTL}			50	pf
Logic Inputs ("0" Level Applies HOLD or MEMORY READY)**					
Holds $\varnothing_1$ NMOS 'High', $\varnothing_2$ NMOS 'Low', $\varnothing_2$ TTL 'Low'	$\overline{\text{HOLD 1}}$	−.2		+.4	Vdc
Holds $\varnothing_1$ NMOS 'Low', $\varnothing_2$ NMOS 'High', $\varnothing_2$ TTL 'High', and MEMORY CLOCK 'High'	MEMORY READY	−.2		+.4	Vdc

PIN	CONNECTION
1	GND
3	MEMORY CLOCK
5	$\varnothing_2$ TTL
7	V_{cc} (+5VDC)
12	$\varnothing_2$ NMOS
13	$\varnothing_1$ NMOS
18	GND
20	$\overline{\text{HOLD 1}}$
22	MEMORY READY
24	2xfc

Note All dimensions are in inches

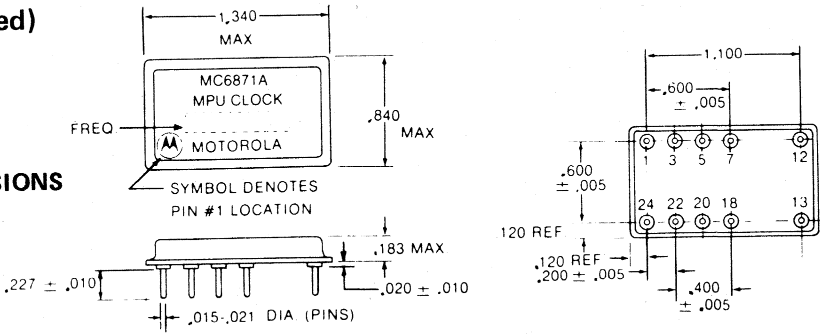
*Into specified test load

**Must be externally held at "1" level (2.4V min., 5.0V max.) if not used

***Apply the following parameters for frequencies other than 1 MHz:
$T\phi_1H$=0.5 (P-140) ns
$T\phi_2H$=0.5 (P-100) ns
Tx=(P-60) ns
where P=desired period of operation in nanoseconds

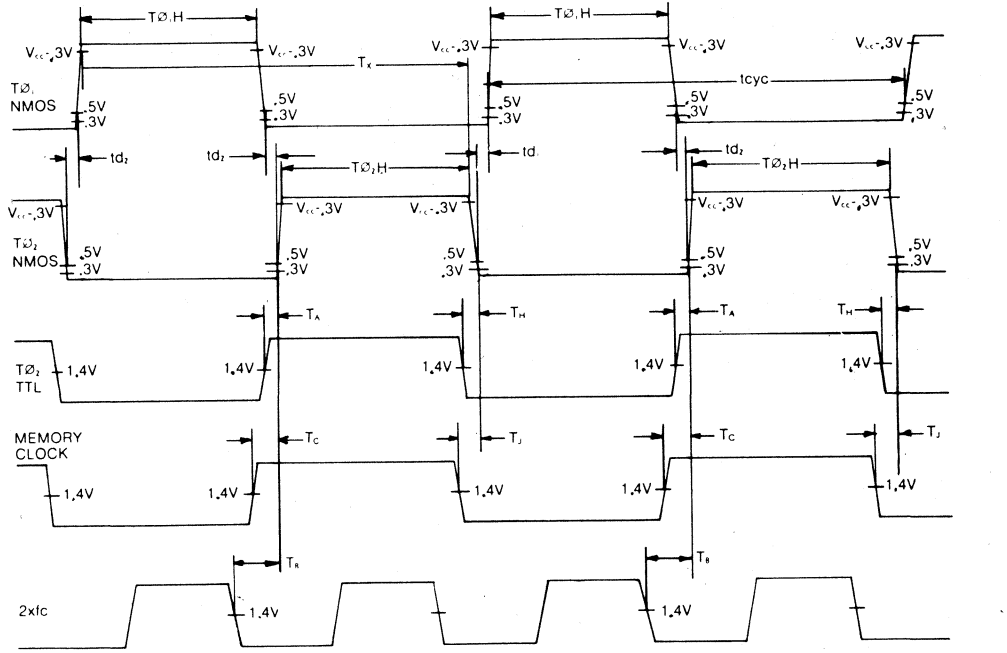
MC6871A (continued)

DIMENSIONS

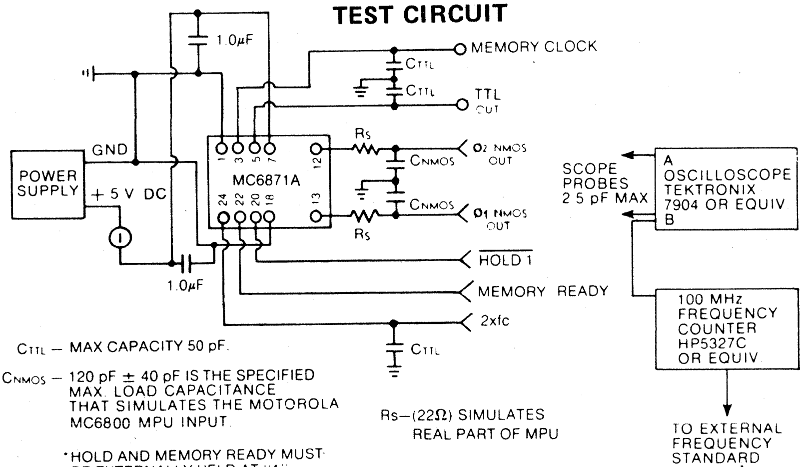


WAVEFORM TIMING

(ALL TIME IN NANOSECONDS)



TEST CIRCUIT



C_{TTL} — MAX CAPACITY 50 pF.

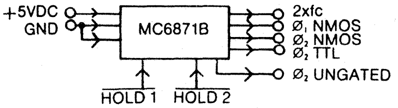
C_{NMOS} — 120 pF ± 40 pF IS THE SPECIFIED MAX. LOAD CAPACITANCE THAT SIMULATES THE MOTOROLA MC6800 MPU INPUT.

*HOLD AND MEMORY READY MUST BE EXTERNALLY HELD AT "1" LEVEL (2.4VDC MIN., 5.0VDC MAX.) WHEN NOT USED

Rs—(22Ω) SIMULATES REAL PART OF MPU

MC6871B

alternate function microprocessor clock
250 kHz to 2.5 MHz



specifications

Rating	Symbol	Value	Unit
Supply Voltage	V_{cc}	5.00±5%	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C
Power Supply Drain (max.)	I_{pd}	100	mA

ELECTRICAL CHARACTERISTICS (V_{cc} = 5.0 ± 5%, V_{ss} = O,T_A = 0° to 70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Frequency					
Operating Frequency Frequency stability (inclusive of calibration tolerance at +25°C, operating temperature, input voltage change, load change, aging, shock and vibration)	f_c	.250	 ±.01	2.5	MHz %
NMOS Outputs at 1.0 MHz Operation***					
Pulse Width (meas. at V_{cc} = −.3V dc level)	$TØ_1H$ $TØ_2H$	430 450			ns ns
Logic Levels	V_{OLC} V_{OHC}	V_{ss}-.1 V_{cc}-.3	— —	V_{ss}+.3 V_{cc}+.1	Vdc Vdc
Rise and Fall Times	t_r t_f	5 5	12 12	50 50	ns ns
*Overshoot/Undershoot Logic "1" Logic "0"	V_{OS}	V_{cc}-.5 V_{ss}-.5		V_{cc}+.5 V_{ss}+.5	Vdc Vdc
Pulse duration of any overshoot or undershoot	T_{OS}			40	ns
Period @ 0.3V dc Level	t_{cyc}		1.00		us
Edge Timing @ V_{cc}=0.3V dc	Tx	940			ns
NMOS Relationship @ +0.5V dc	t_{d1} t_{d2}	0 0		 8.0	 us
TTL Outputs					
In ref. to $Ø_2$ NMOS @ 0.3V dc					
$Ø_2$ TTL @ 1.4V dc	T_A T_H	15 10	30 25	45 40	ns ns
$Ø_2$ Ungated @ 1.4V dc	T_C T_J	30 20	50 40	70 60	ns ns
2xfc @ 1.4V dc	T_B	40	80	120	ns
Logic Levels	V_{OH} V_{OL}	2.4	3.2 .3	 .4	Vdc Vdc
Rise and Fall Times .4V and 2.4V 2.4V and .4V	t_r t_f			15 15	ns ns
Logic "0" Sink (/Gate)	I_{OL}			−1.6	mA
Logic "1" Source (/Gate)	I_{OH}			+40	uA
Current Output Shorted	I_{SC}	−18		−57	mA
Load					
NMOS—Load Capacity $Ø_1$, $Ø_2$	C_{NMOS}	80	120	160	pf
TTL—No. of Loads				5	ttl
TTL—Load Capacity	C_{TTL}			50	pf
Logic Inputs ("0" Level applies HOLD)**					
Holds $Ø_1$ NMOS 'High', $Ø_2$ NMOS 'Low', $Ø_2$ TTL 'Low'	$\overline{\text{HOLD 1}}$	−.2		+.4	Vdc
Holds $Ø_1$ NMOS 'Low', $Ø_2$ NMOS 'High', $Ø_2$ TTL 'High'	$\overline{\text{HOLD 2}}$	−.2		+.4	Vdc

*Into specified test load

**Must be externally held at "1" level (2.4V min., 5.0V max.) if not used

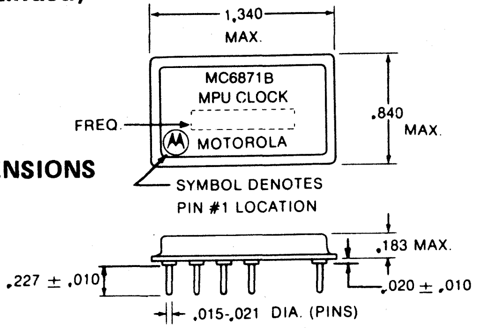
***Apply the following parameters for frequencies other than 1 MHz:
$T\phi_1H$=0.5 (P-140) ns
$T\phi_2H$=0.5 (P-100) ns
Tx=(P-60) ns
where P=desired period of operation in nanoseconds

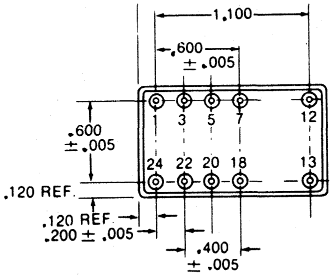
PIN	CONNECTION
1	GND
3	$Ø_2$ TTL UNGATED
5	$Ø_2$ TTL
7	V_{cc} (+5VDC)
12	$Ø_2$ NMOS
13	$Ø_1$ NMOS
18	GND
20	$\overline{\text{HOLD 1}}$
22	$\overline{\text{HOLD 2}}$
24	2xfc

Note: 4xfc available on request

Note All dimensions are in inches

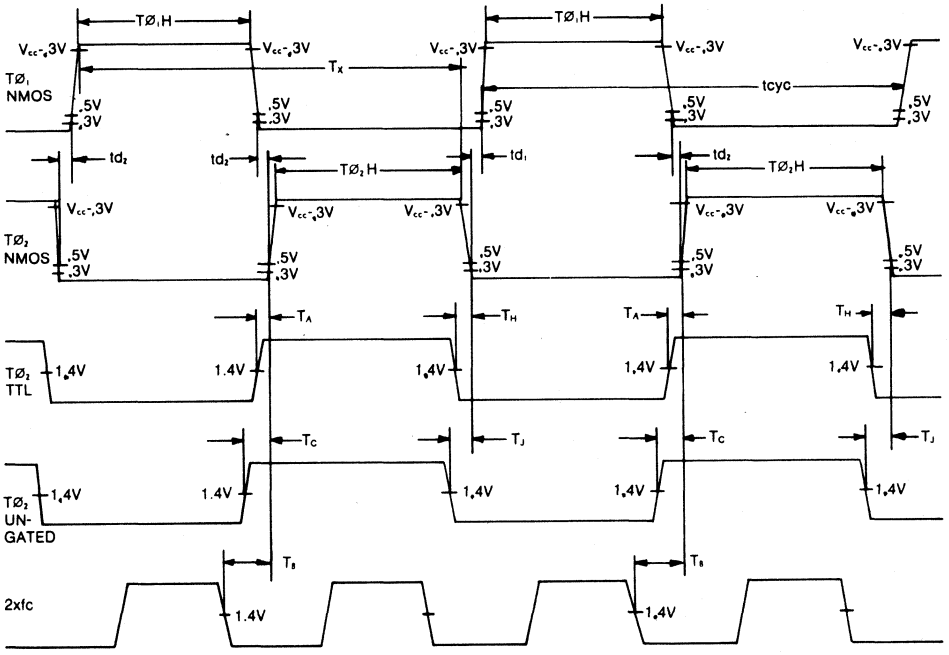
DIMENSIONS



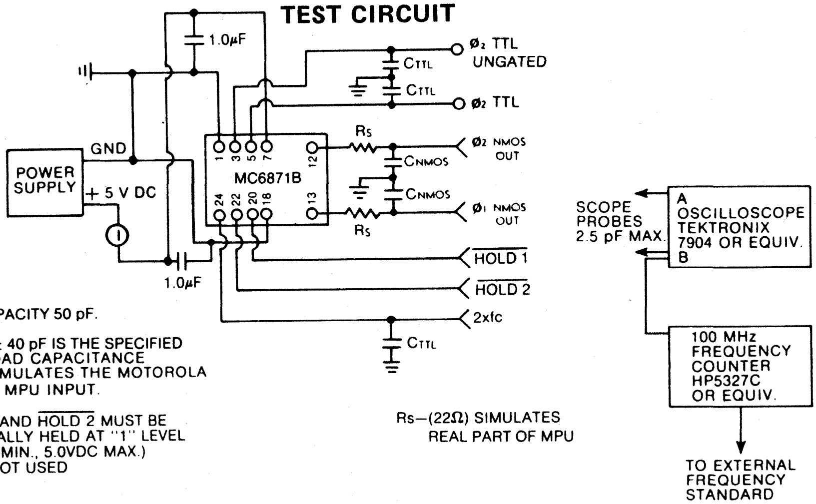


WAVEFORM TIMING

ALL TIME IN NANOSECONDS.



TEST CIRCUIT



C_{TTL} — MAX CAPACITY 50 pF.

C_{NMOS} — 120 pF ± 40 pF IS THE SPECIFIED MAX. LOAD CAPACITANCE THAT SIMULATES THE MOTOROLA MC6800 MPU INPUT.

$\overline{\text{HOLD 1}}$ AND $\overline{\text{HOLD 2}}$ MUST BE EXTERNALLY HELD AT "1" LEVEL (2.4VDC MIN., 5.0VDC MAX.) WHEN NOT USED

Rs—(22Ω) SIMULATES REAL PART OF MPU

MOTOROLA

MC6875
MC6875A

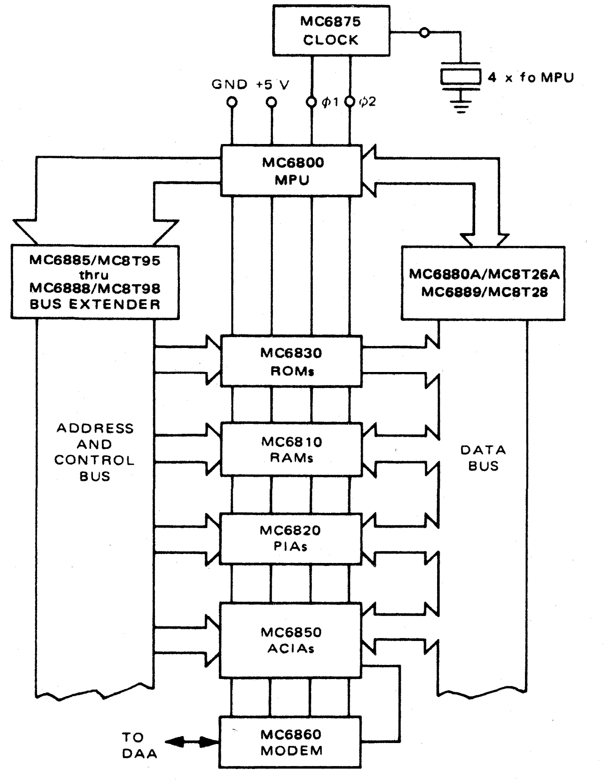
Specifications and Applications Information

M6800 CLOCK GENERATOR

Intended to supply the non-overlapping $\phi1$ and $\phi2$ 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.

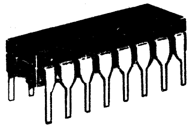
Schottky technology is employed for high speed and PNP-buffered inputs are employed for NMOS compatibility. A single +5 V power supply, and a crystal or RC network for frequency determination are required.

Typical MPU System with Bus Extenders



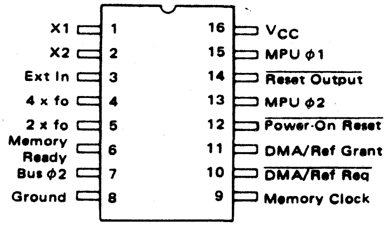
M6800 TWO-PHASE CLOCK GENERATOR/DRIVER

SCHOTTKY MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620

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°C.)

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	°C
Storage Temperature Range Ceramic Package	T_{stg}	-65 to +150	°C
Operating Junction Temperature Ceramic Package	T_J	 175	°C

NOTE:
MC6875AL requires the use of a heat sink. See note in Thermal Data.

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	°C

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_{CC} = 4.75 V, I_{OH} = −500 μA)	V_{OH}	2.4	–	–	V
$\overline{\text{Reset}}$ Output (V_{CC} = 4.75 V, V_{IH} = 3.3 V, $I_{OH\overline{R}}$ = −100 μA)	$V_{OH\overline{R}}$	2.4	–	–	V
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_{CC} = 4.75 V, I_{OL} = 16 mA)	V_{OL}	–	–	0.5	V
$\overline{\text{Reset}}$ Output (V_{CC} = 4.75 V, V_{IL} = 0.8 V, $I_{OL\overline{R}}$ = 3.2 mA)	$V_{OL\overline{R}}$	–	–	0.5	V
Input Voltage — High Logic State Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	V_{IH}	2.0	–	–	V
Input Voltage — Low Logic State Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs	V_{IL}	–	–	0.8	V
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_{IC}	–	–	-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 (V_{CC} = 4.75 V, V_{IH} = 5.0 V)	I_{IH}	–	–	25	μA
$\overline{\text{Power-On Reset}}$ (V_{CC} = 5.0 V, $V_{IH\overline{R}}$ = 5.0 V)	$I_{IH\overline{R}}$	–	–	50	μA
Input Current — Low Logic State Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs (V_{CC} = 5.25 V, V_{IL} = 0.5 V)	I_{IL}	–	–	-250	μA
$\overline{\text{Power-On Reset}}$ Input (V_{CC} = 5.25 V, V_{IL} = 0.5 V)	$I_{IL\overline{R}}$	–	–	-250	μA

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

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 $\phi 2$) (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 $\phi 1$) (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 $\overline{\text{Power-On Reset}}$ (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 $\phi 1$ high, $\phi 2$ 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 $\phi 1$ low, $\phi 2$ high state for slow memory interface.
- MPU $\phi 1$, MPU $\phi 2$ – Capable of driving the $\phi 1$ and $\phi 2$ inputs on two MC6800s.
- BUS $\phi 2$ – An output nominally in phase with MPU $\phi 2$ having MC8T26A type drive capability.
- MEMORY CLOCK – An output nominally in phase with MPU $\phi 2$ which free runs during a refresh request cycle.
- $\overline{\text{POWER-ON RESET}}$ – A Schmitt trigger input which controls $\overline{\text{Reset}}$. 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

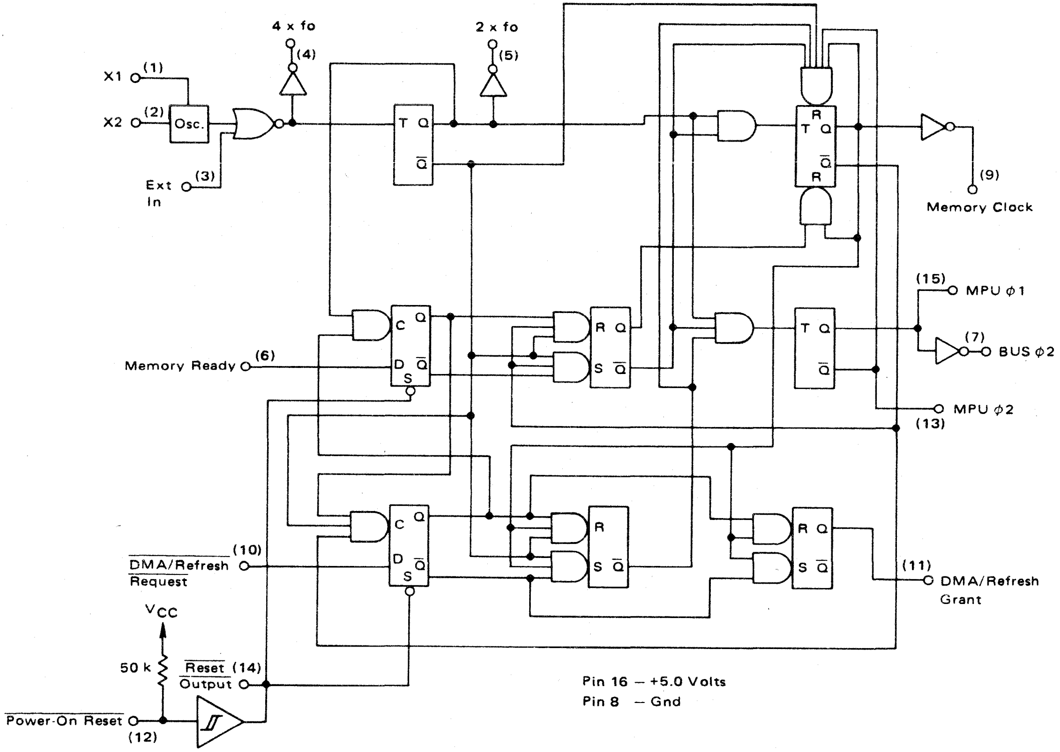


FIGURE 2 – TYPICAL HYSTERESIS CHARACTERISTIC OF $\overline{\text{RESET}}$ FUNCTION

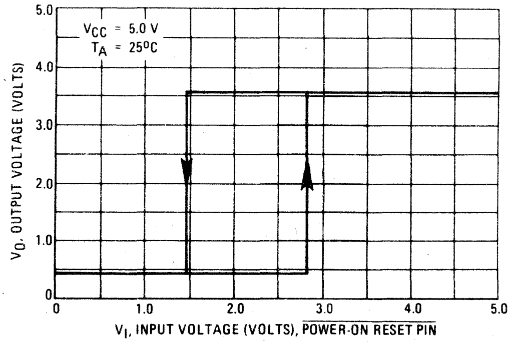
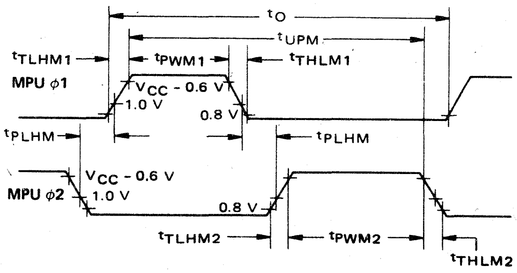


FIGURE 3 – TIMING DIAGRAM FOR MPU ϕ1 AND ϕ2



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

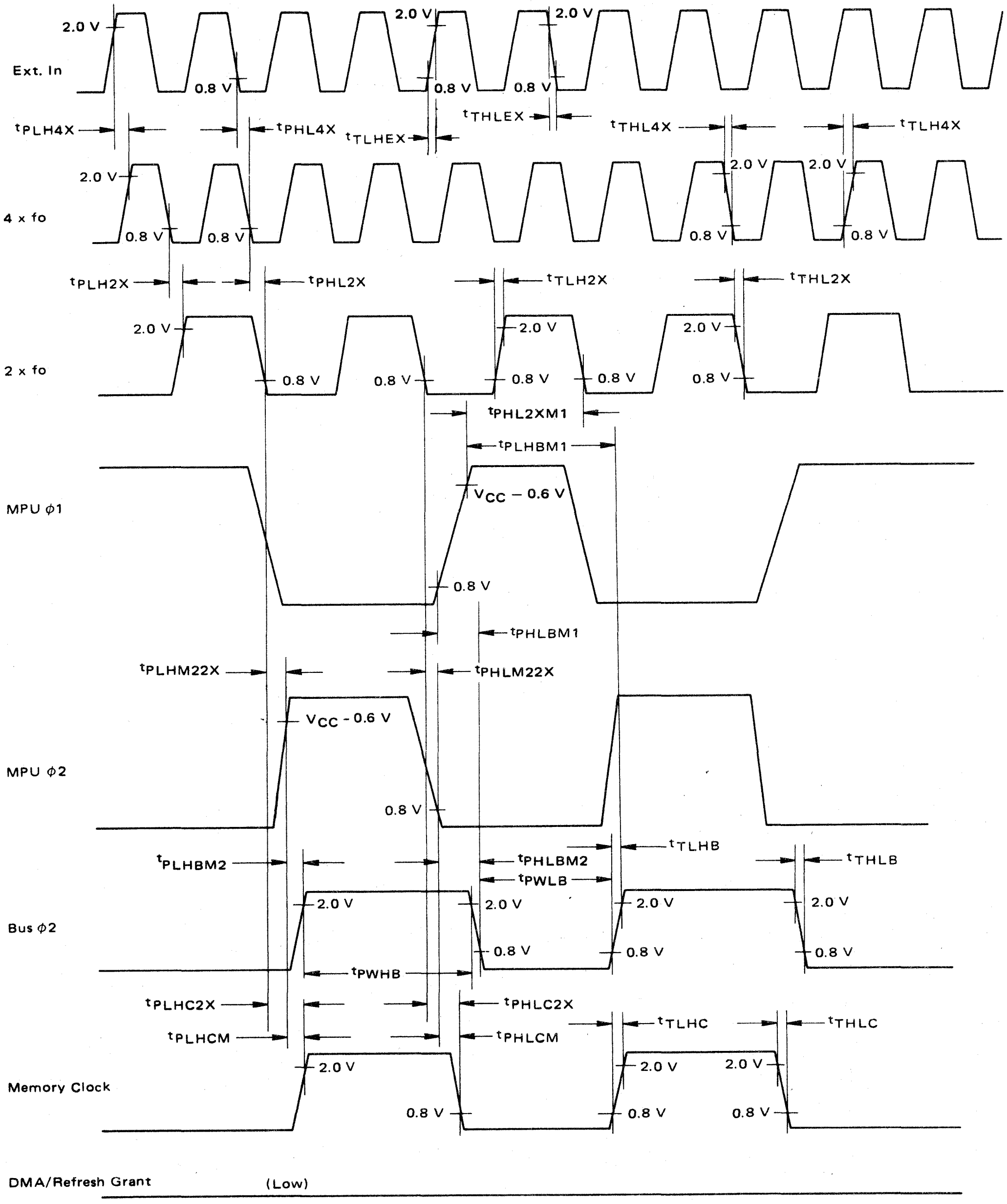


FIGURE 5 – TIMING DIAGRAM FOR MEMORY READY STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLMR} = t_{TLHMR} = 5.0$ ns

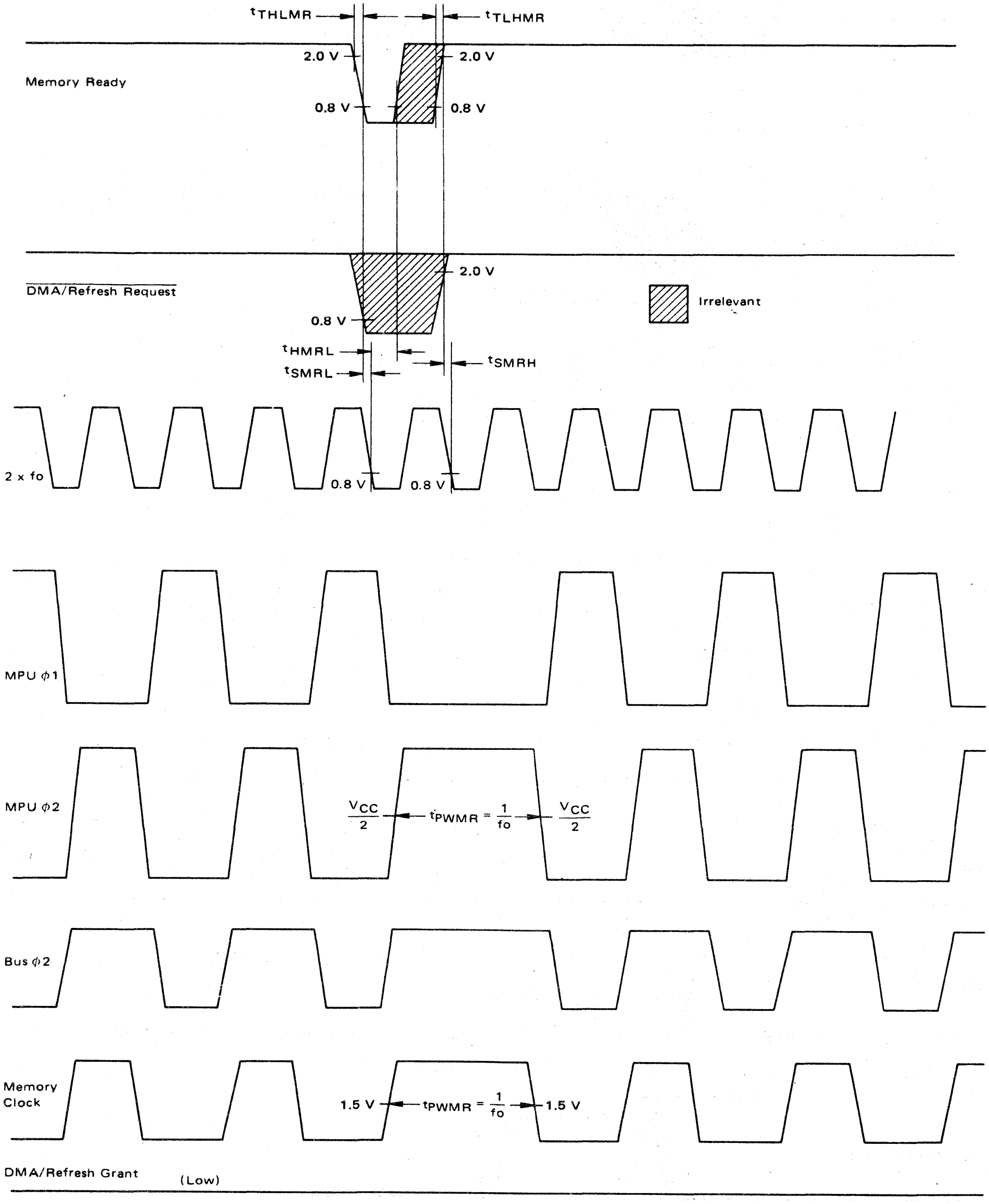


FIGURE 6 – TIMING DIAGRAM FOR $\overline{\text{DMA/REFRESH REQUEST}}$ STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLDR} = t_{TLHDR} = 5.0$ ns

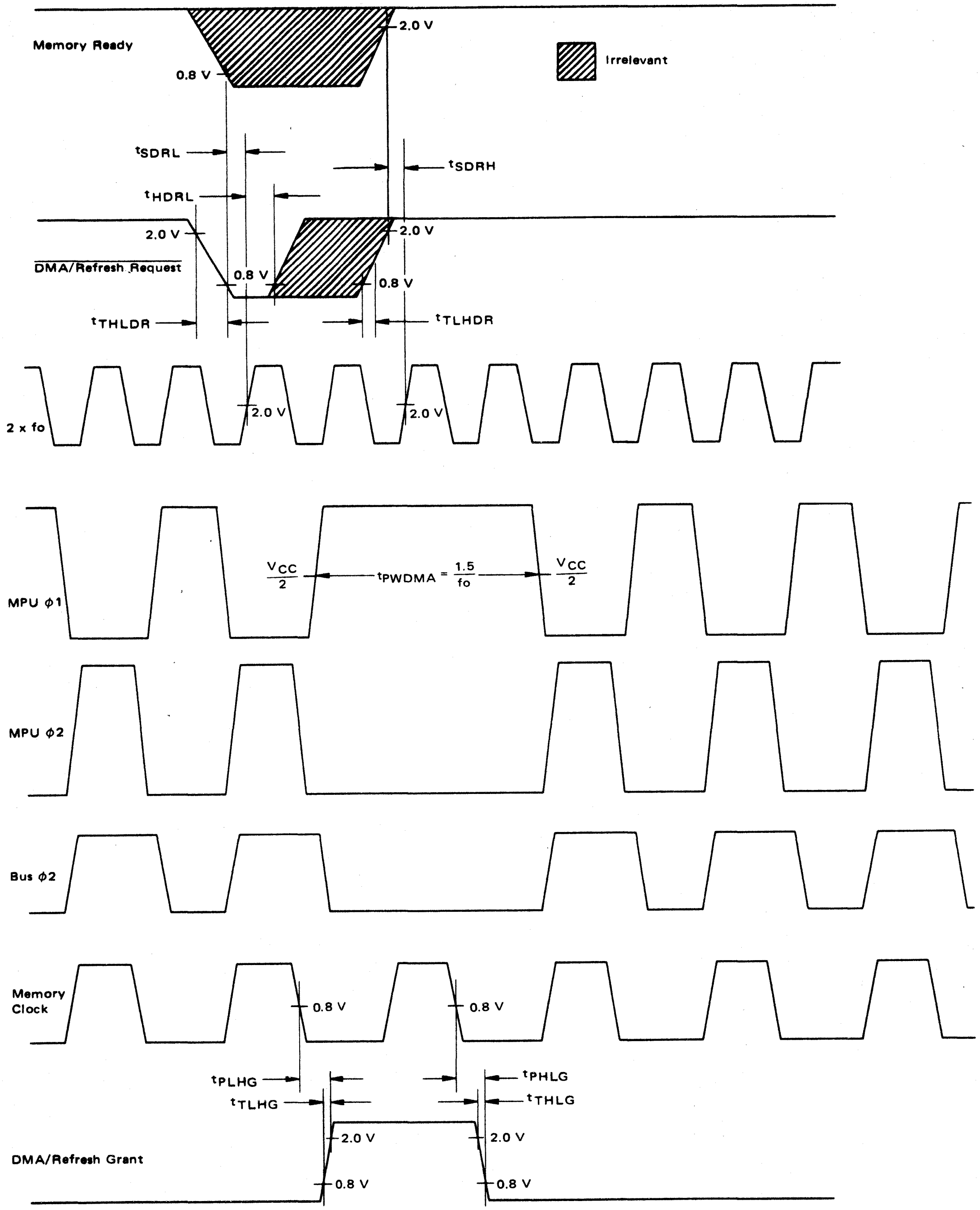


FIGURE 7 – $\overline{\text{POWER ON RESET}}$

Input Voltage: 0 to 5.0 V, f = 100 kHz – Pulse Width = 1.0 µs, $t_{TLH} = t_{THL} = 25$ ns

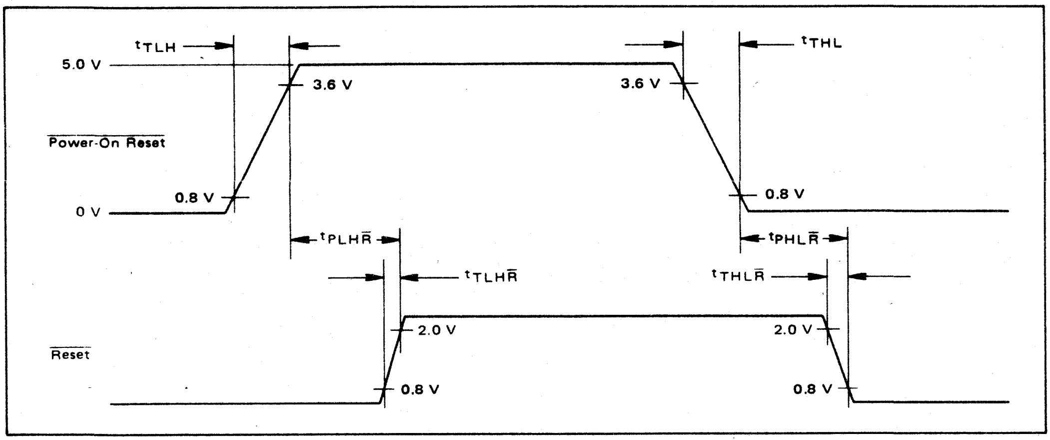
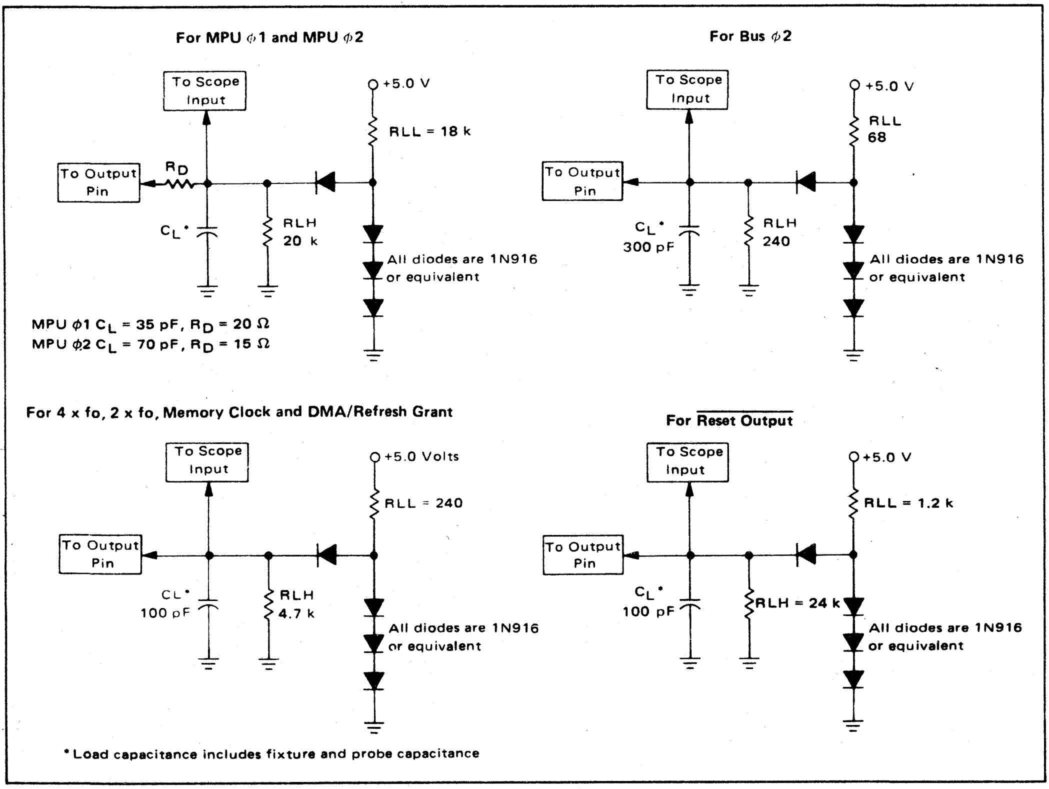


FIGURE 8 – LOAD CIRCUITS



* Load capacitance includes fixture and probe capacitance

APPLICATIONS INFORMATION

FIGURE 9 – TYPICAL RC FREQUENCY versus VOLTAGE

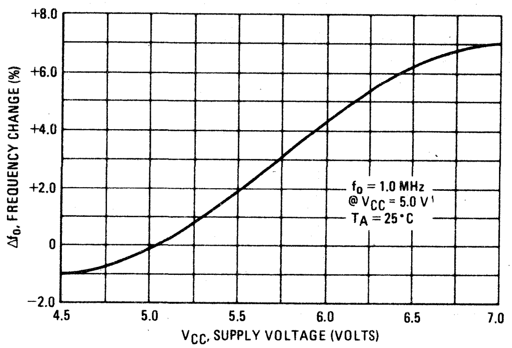


FIGURE 10 – TYPICAL RC FREQUENCY versus TEMPERATURE

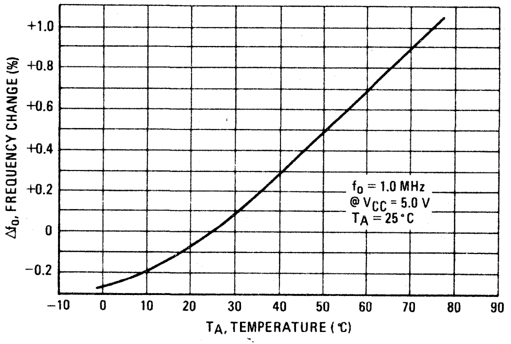
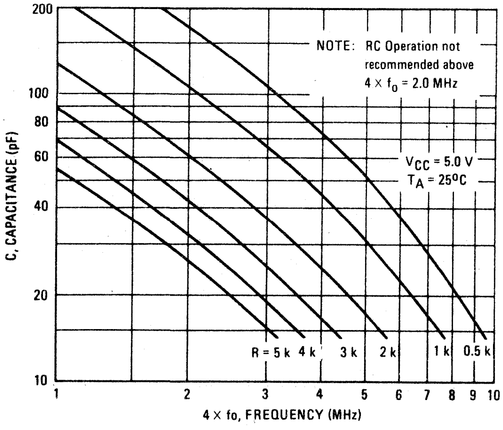


FIGURE 11 – TYPICAL FREQUENCY versus RESISTANCE FOR C VARIABLE



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.

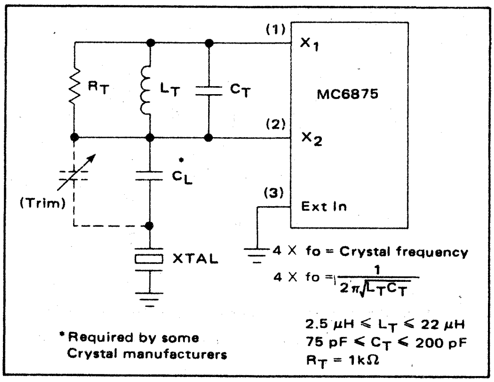
Unused inputs should be connected to V_{CC} or ground. $\overline{\text{Memory Ready}}$, $\overline{\text{DMA/Refresh Request}}$ and $\overline{\text{Power-On Reset}}$ should be connected to V_{CC} when not used. The External Input should be connected to ground when not used.

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



4 X fo = Crystal frequency

$4 \times fo = \frac{1}{2\pi\sqrt{L_T C_T}}$

$2.5\ \mu H \leq L_T \leq 22\ \mu H$

$75\ pF \leq C_T \leq 200\ pF$

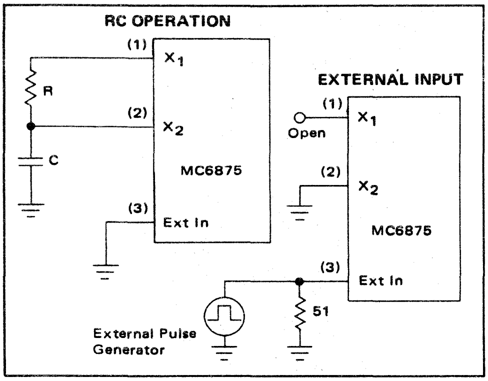
$R_T = 1k\Omega$

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 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 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 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 Power-On Reset input pin of the MC6875 is not suitable for use with a manual MPU reset switch if the 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 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 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 DMA/REFRESH REQUEST OR MEMORY READY INPUTS

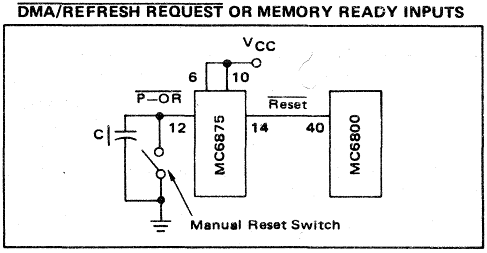
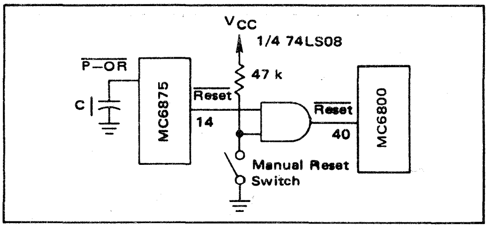


FIGURE 15 – MANUAL RESET FOR SYSTEMS USING DYNAMIC RAM OR SLOW STATIC RAM IN CONJUNCTION WITH MEMORY READY OR DMA/REFRESH REQUEST INPUTS

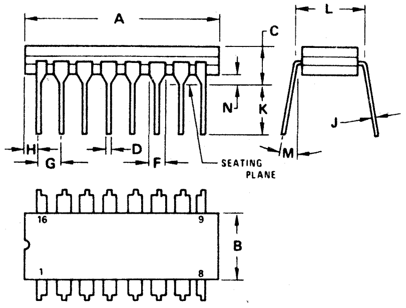


NOTE:
Operation of the MC6875AL over the full military temperature range (to maximum T_A) will result in excessive operating junction temperature.

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 $T_A \approx 95^{\circ}C$.

Contact AAVID Engineering, Inc.
30 Cook Court
Laconia, New Hampshire 03246
Tel. (603) 524-4443

OUTLINE DIMENSIONS



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

CASE 620-02

$R_{\theta JA}$ = 100°C/W (Typ)

$R_{\theta JC(peak)}$ = 15°C/W

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA}}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA}$ = Thermal Resistance Junction to Ambient

$R_{\theta JC}$ = Thermal Resistance Junction to Case

MOTOROLA

MC6880A
MC8T26A

This device may be ordered under either of the above type numbers.

QUAD THREE-STATE BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

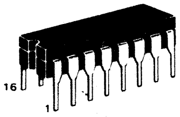
QUAD THREE-STATE BUS TRANSCEIVER

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.

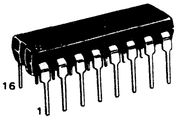
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier 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

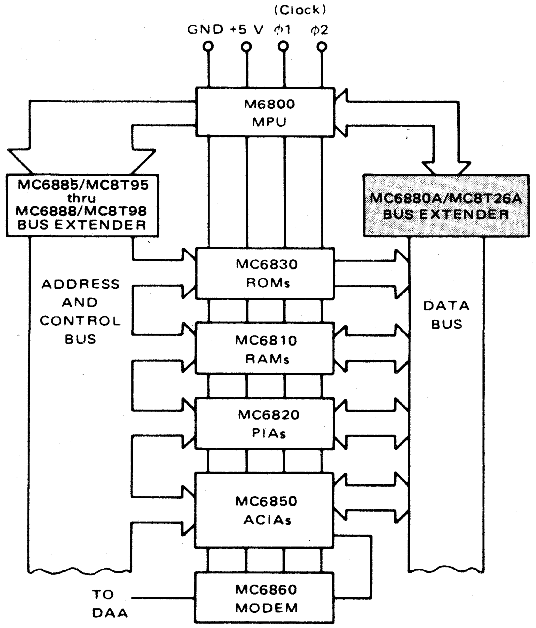


L SUFFIX
CERAMIC PACKAGE
CASE 620

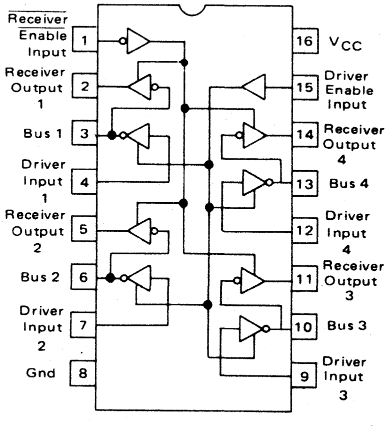


P SUFFIX
PLASTIC PACKAGE
CASE 648

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS – MC6880A MC8T26A



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	T_J		°C
Ceramic Package		175	
Plastic Package		150	
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (4.75 V $\leq V_{CC} \leq$ 5.25 V and 0°C $\leq T_A \leq$ 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	$I_{IL(D)\ DIS}$				
(Driver Input, $V_{IL(D)}$ = 0.4 V)		–	–	-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	
(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	I_{CC}	–	–	87	mA
(V_{CC} = 5.25 V)					

(1) Only one output may be short-circuited at a time.

SWITCHING CHARACTERISTICS (Unless otherwise noted, specifications apply at $T_A = 25^{\circ}C$ 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

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

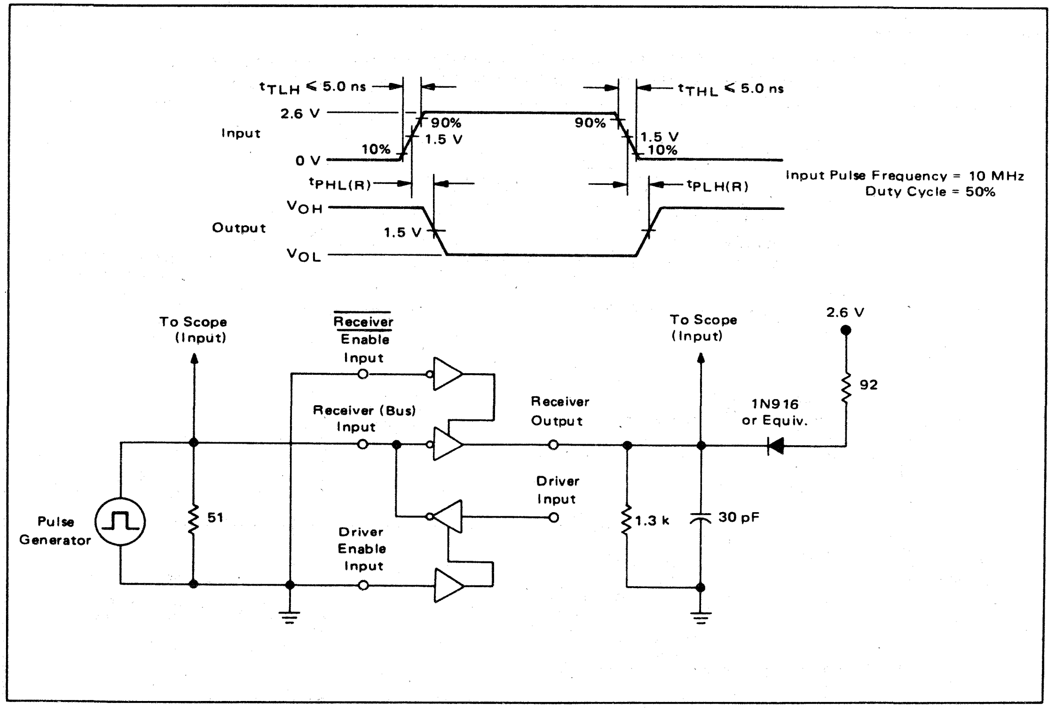


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

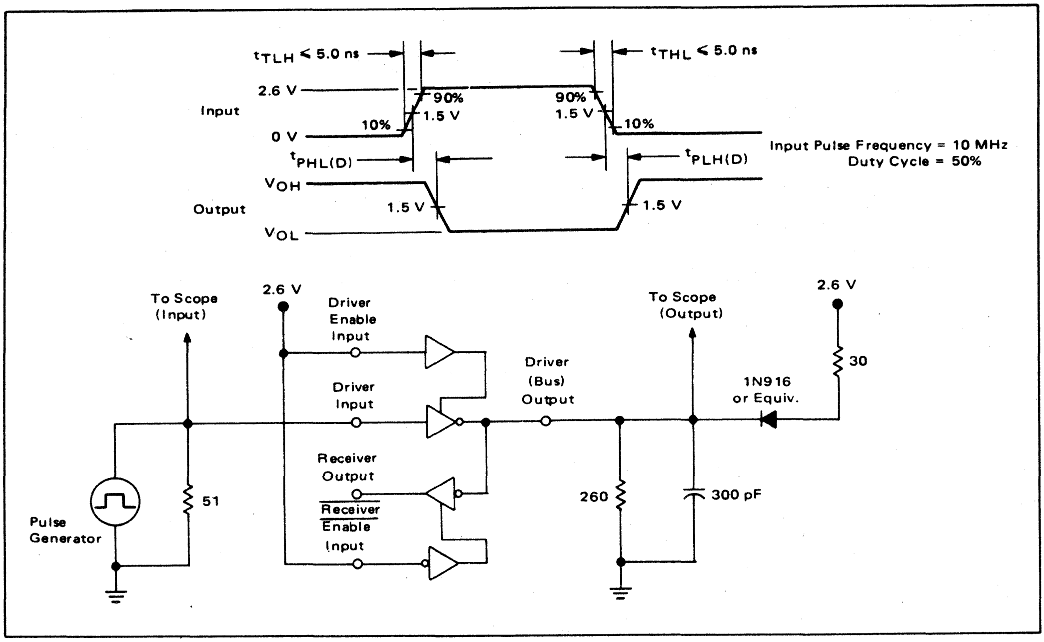


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

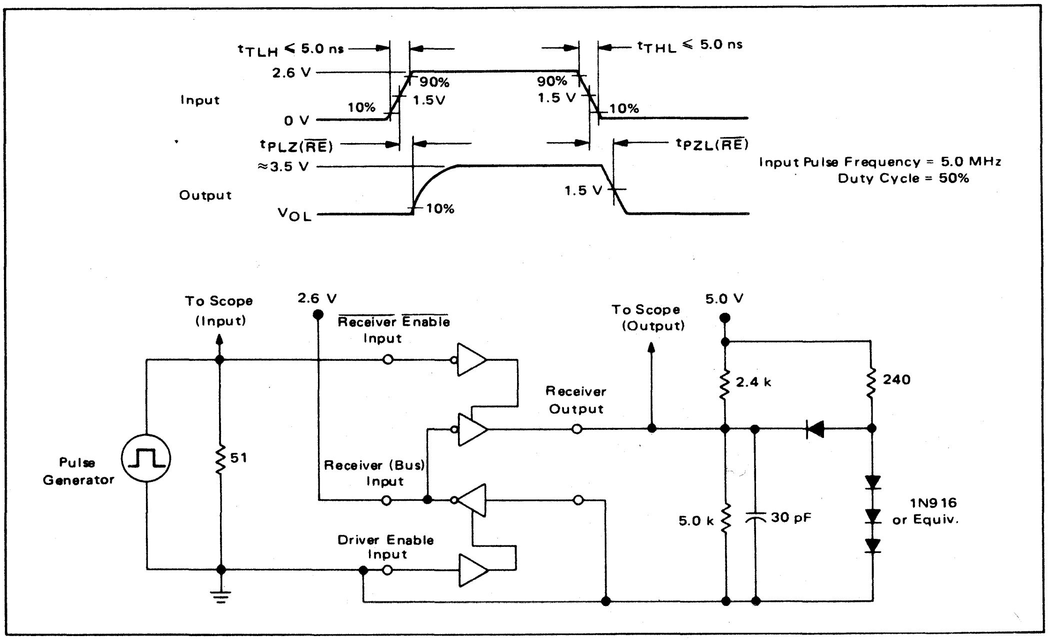


FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

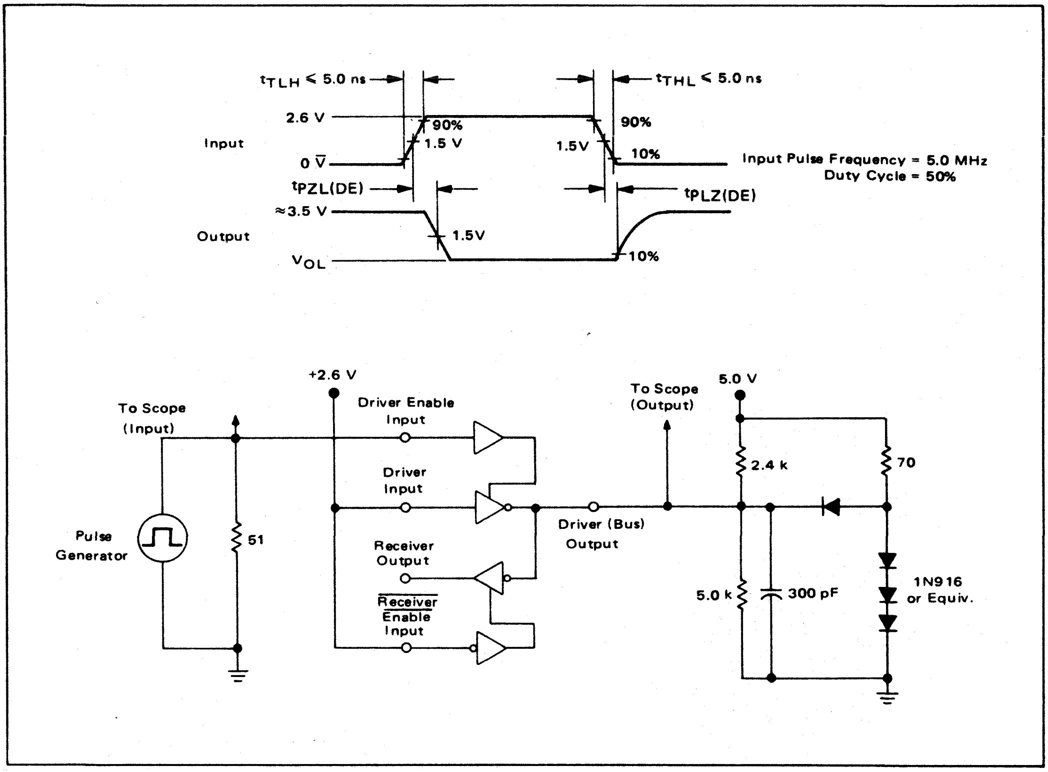
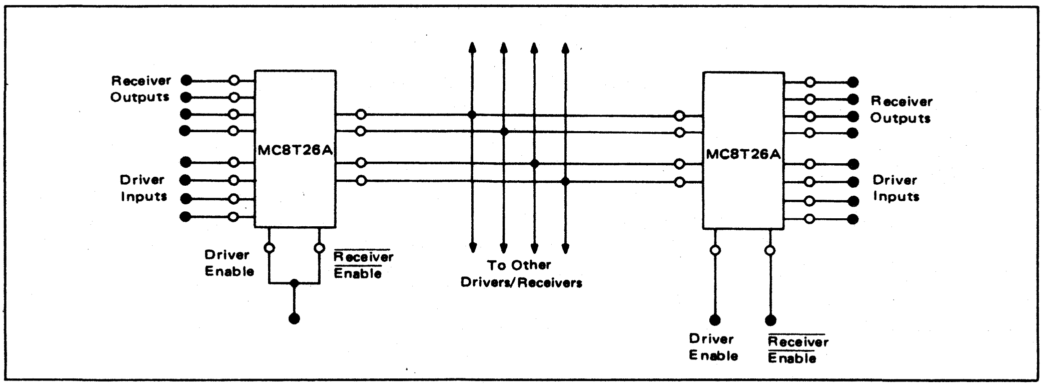
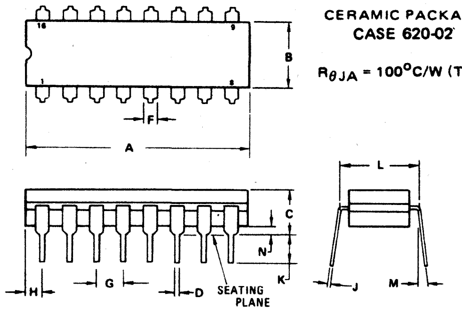


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS



L SUFFIX
CERAMIC PACKAGE
CASE 620-02

$R_{\theta JA}$ = 100°C/W (Typ)

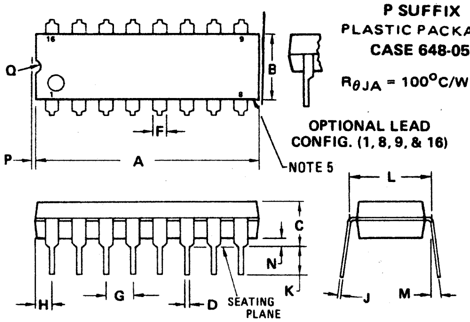


NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

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

P SUFFIX
PLASTIC PACKAGE
CASE 648-05

$R_{\theta JA}$ = 100°C/W (Typ)



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.

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

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA}(Typ)}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section
T_A = Maximum Desired Operating Ambient Temperature
$R_{\theta JA}(Typ)$ = Typical Thermal Resistance Junction to Ambient

MOTOROLA

SN74LS783
MC6883

Advance Information

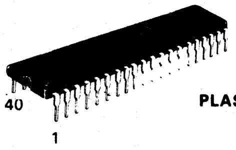
SYNCHRONOUS ADDRESS MULTIPLEXER

LOW POWER SCHOTTKY

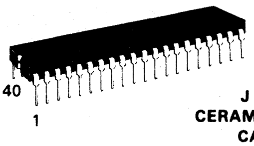
SYNCHRONOUS ADDRESS MULTIPLEXER

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

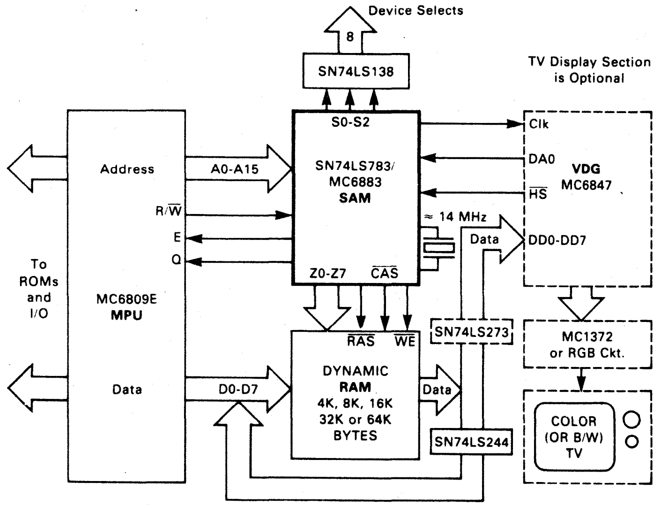


N SUFFIX
PLASTIC PACKAGE
CASE 711

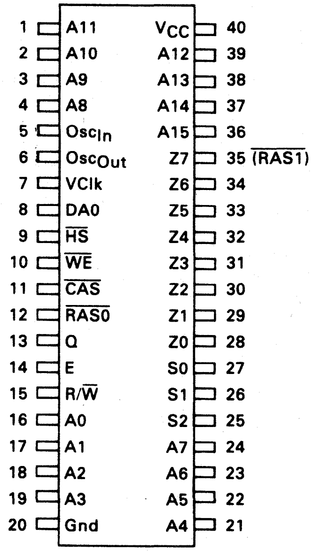


J SUFFIX
CERAMIC PACKAGE
CASE 734

SYSTEM BLOCK DIAGRAM



PIN ASSIGNMENT



Pin	Name	Name	Pin
1	A11	V_{CC}	40
2	A10	A12	39
3	A9	A13	38
4	A8	A14	37
5	Osc_{In}	A15	36
6	Osc_{Out}	Z7	35 ($\overline{\text{RAS1}}$)
7	VClk	Z6	34
8	DA0	Z5	33
9	$\overline{\text{HS}}$	Z4	32
10	$\overline{\text{WE}}$	Z3	31
11	$\overline{\text{CAS}}$	Z2	30
12	$\overline{\text{RAS0}}$	Z1	29
13	Q	Z0	28
14	E	S0	27
15	R/$\overline{\text{W}}$	S1	26
16	A0	S2	25
17	A1	A7	24
18	A2	A6	23
19	A3	A5	22
20	Gnd	A4	21

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

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	°C

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 (V_{CC} = Max, V_{in} = 5.25 V) VClk Input (V_{CC} = Max, V_{in} = 5.25 V) DA0 Input (V_{CC} = Max, V_{in} = 5.25 V) Osc_{Out} Input (V_{CC} = Max, V_{in} = 7.0 V) All Other Inputs Except Osc_{In}	I_I	 — — — —	 — — — —	 200 100 250 100	µA
Input Current High Logic State (V_{CC} = Max, V_{in} = 2.7 V) All Inputs Except VClk, Osc_{In}*	I_{IH}	 —	 —	 20	µA
Input Current — Low Logic State (V_{CC} = Max, V_{in} = 0.4 V) DA0 Input (V_{CC} = Max, V_{in} = 0.4 V) VClk Input (V_{CC} = Max, V_{in} = 0.4 V, Osc_{In} = Gnd) Osc_{Out} Input (V_{CC} = Max, V_{in} = 0.4 V) All Other Inputs Except Osc_{In}	I_{IL}	 — — — —	 — −30 — —	 −1.2 −60 −8 −.4	mA
Output Voltage — High Logic State (V_{CC} = Min, I_{OH} = −1.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$ (V_{CC} = Min, I_{OH} = −0.2 mA) E, Q (V_{CC} = Min, I_{OH} = −0.2 mA) All Other Outputs	 $V_{OH(C)}$ $V_{OH(E)}$ V_{OH}	 3.0 V_{CC} − 0.75 2.7	 — — —	 — — —	V
Output Voltage — Low Logic State (V_{CC} = Min, I_{OL} = 8.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$ (V_{CC} = Min, I_{OL} = 4.0 mA) E, Q Outputs (V_{CC} = Min, I_{OL} = 0.8 mA) VClk Output (V_{CC} = Min, I_{OL} = 4.0 mA) All Other Outputs	 $V_{OL(C)}$ $V_{OL(E)}$ $V_{OL(V)}$ V_{OL}	 — — — —	 — — — —	 0.5 0.5 0.6 0.5	V
Power Supply Current	I_{CC}	—	180	230	mA
Output Short-Circuit Current	I_{OS}	30	—	225	mA

*Including Osc_{Out} (when Osc_{In} is grounded).

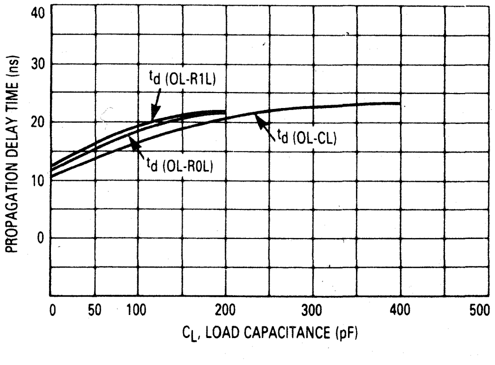
AC CHARACTERISTICS (4.75 V ≤ V_{CC} ≤ 5.25 V and 0 ≤ T_A ≤ 70°C, unless otherwise noted).

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

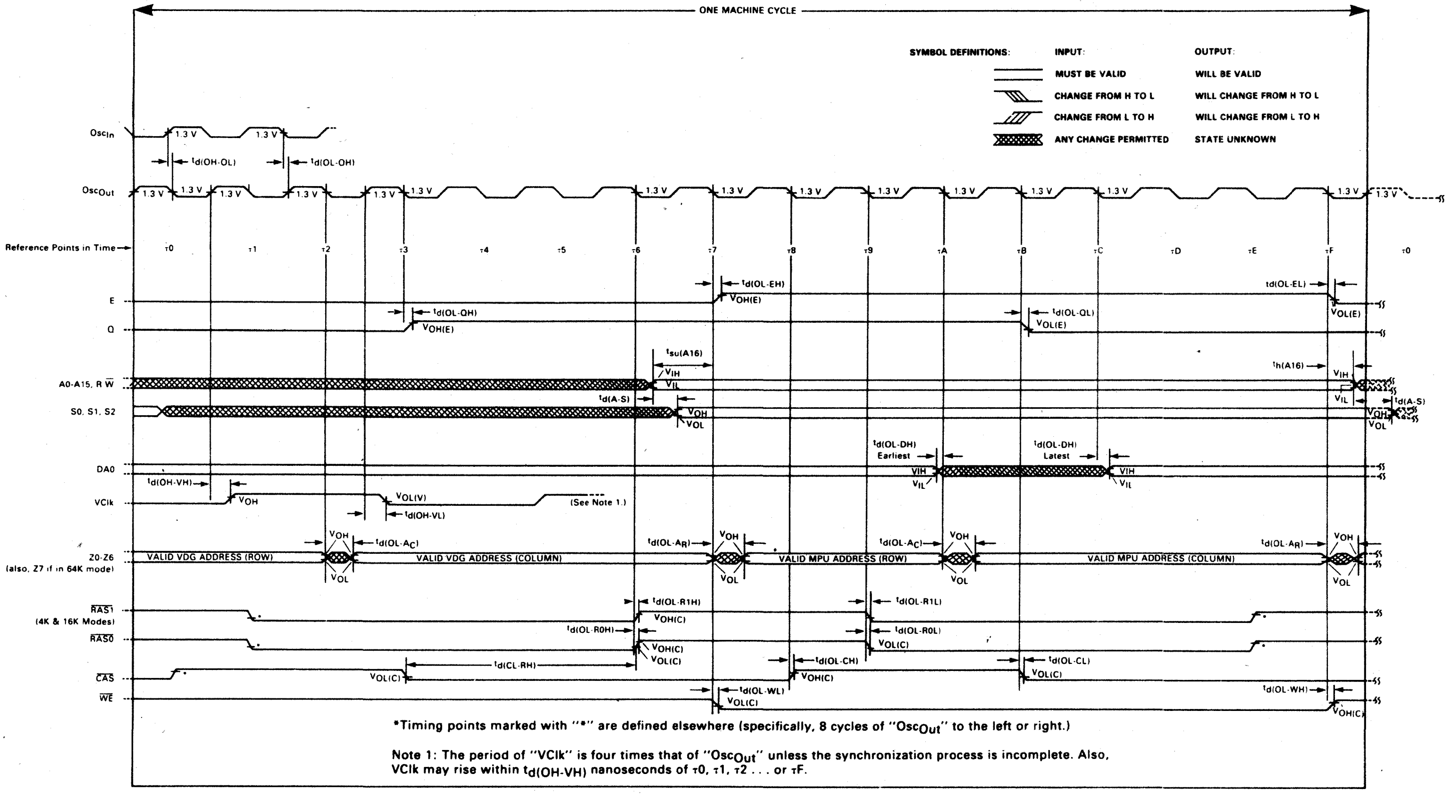
		Name	No.	Function
Input Pins	Power	V_{CC}	40	Apply + 5 volts ± 5%. SAM draws less than 230 mA.
		Gnd	20	Return Ground for +5 volts.
	MPU Address and Control	A15	36	Most Significant Bit.
		A14	37	MPU address bits A0-A15. These 16 signals come directly from the MPU and are used to directly address up to 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.
		A13	38	
		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.
		Osc_{In}	5	Apply 14.31818* MHz crystal and 2.5–30 pF trimmer to ground. See page 12.
	VDG Control	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.
Output Pins		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**.
		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.

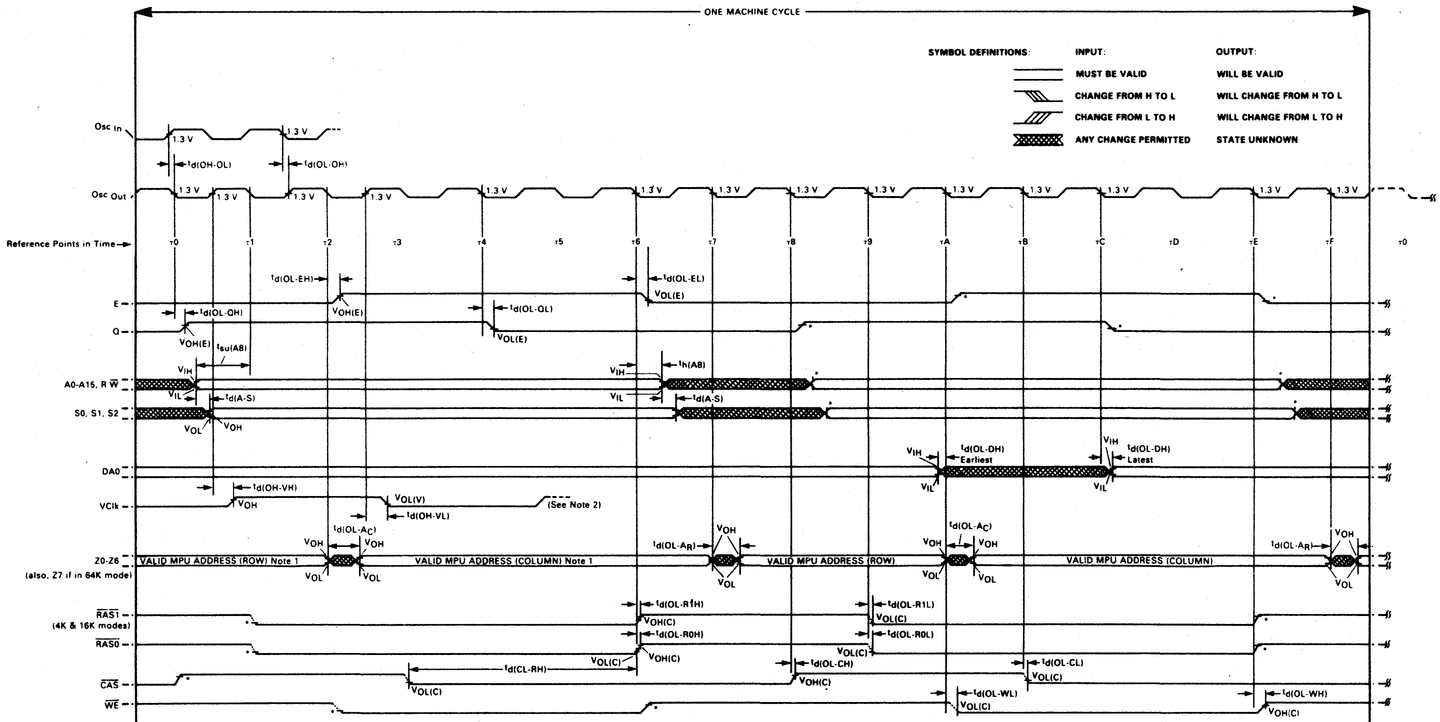
FIGURE 2 — TIMING WAVEFORMS for MPU RATE = SLOW



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 $t_{d(OH\text{-}VH)}$ nanoseconds of τ0, τ1, τ2 . . . or τF.

FIGURE 3— TIMING WAVEFORMS for MPU RATE = FAST

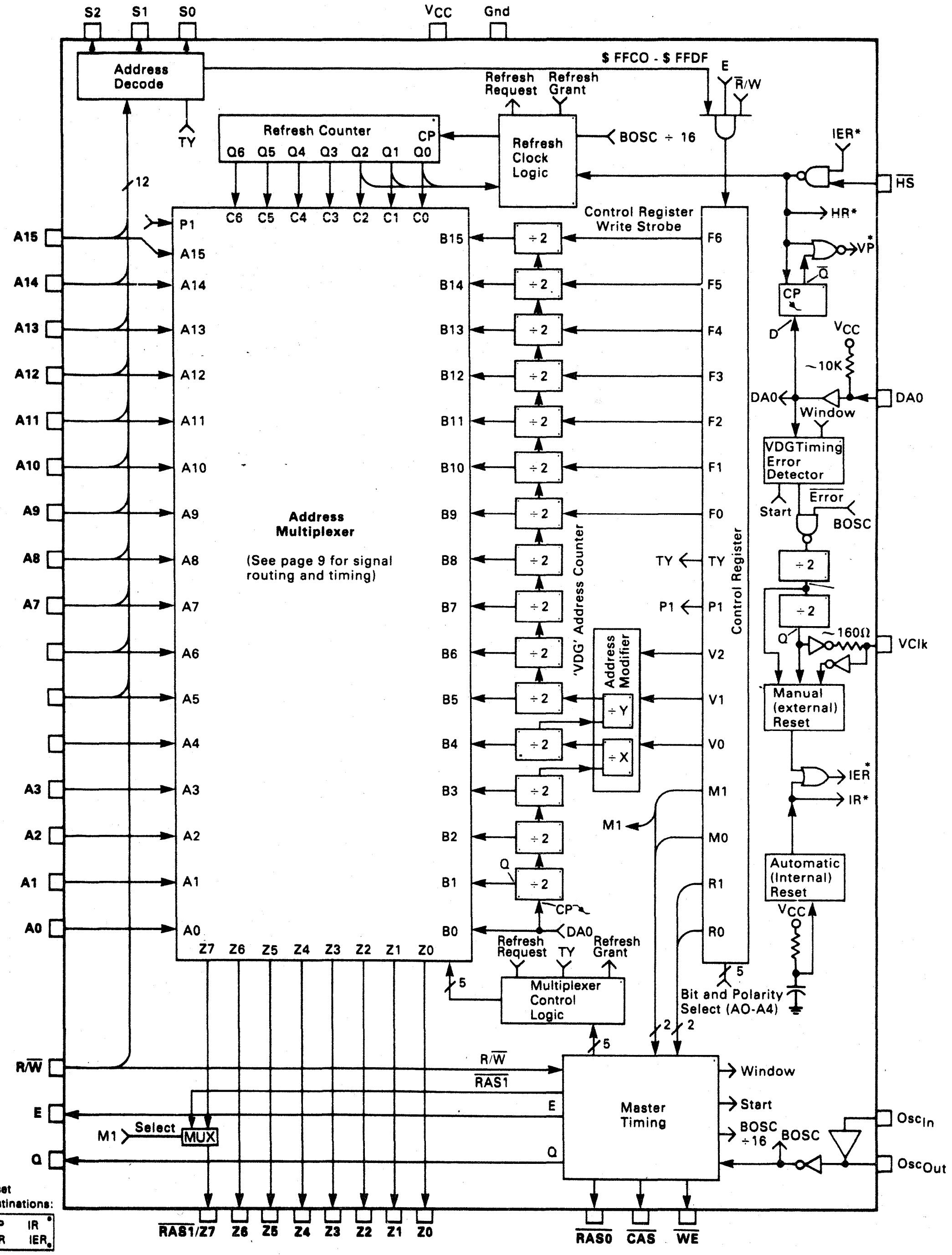


Timing points marked with "" are defined elsewhere (specifically, 8 cycles of "Osc_{Out}" to the left or the right.)

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 $t_{d(OH\text{-}VH)}$ nanoseconds of τ0, τ1, τ2, . . . or τF.

FIGURE 4 — SAM BLOCK DIAGRAM



*Reset Destinations:

VP	IR
HR	IER

See Text . . . Page 8.

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 $\overline{\text{HS}}$ 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 $\overline{\text{FS}}$ and $\overline{\text{MS}}$ 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 $\overline{\text{HS}}$ 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 $\overline{\text{HS}}$ 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 $\overline{\text{RAS1}}$ 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 $\overline{\text{R}}$/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 $\overline{\text{HS}}$ 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 $\overline{\text{HS}}$ 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 $FFC0 through $FFDE.) Note that the **data** on the MPU data bus is irrevelant.

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 $\overline{\text{R}}$/W, E and $FFC0 – $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 $\overline{\text{HS}}$ (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 Source	Row/Column	Signals Routed to Z0–Z7								Timing (Figure 2)
	M1	M0			Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0	
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, $\overline{\text{RAS0}}$ is active for addresses $0000 to $3FFF and $\overline{\text{RAS1}}$ 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.

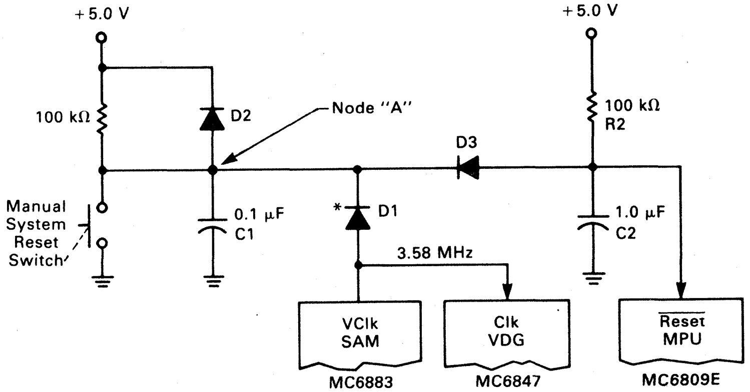
Now, the "E" clocks from all SAM's should be in-phase.

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.

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.

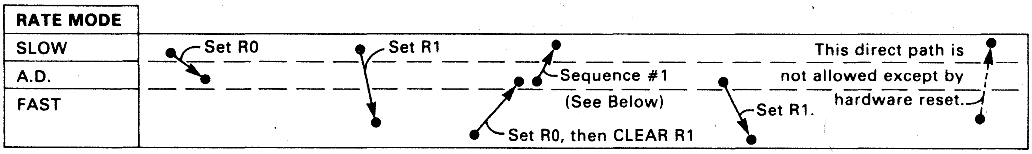
**See timing diagrams on page 5 and 6.

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
```

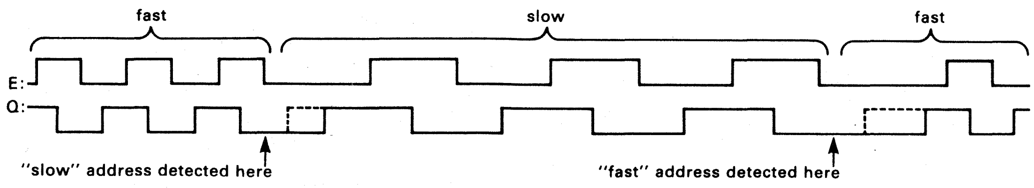
*Note: "TST" instruction affects MC6809E condition code register.

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



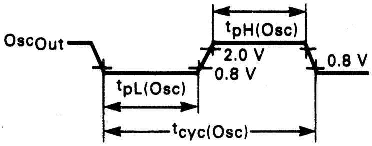
---

*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

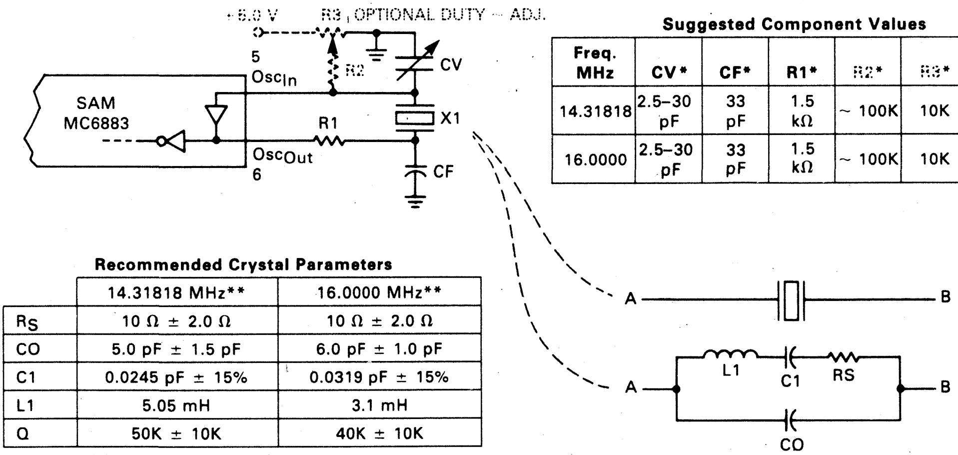
In Figure 4, an amplifier between $\mathrm{Osc}_{\mathrm{In}}$ and $\mathrm{Osc}_{\mathrm{Out}}$ provides the gain for oscillation (using a crystal as shown in Figure 9.) Alternately, Pin 5 ($\mathrm{Osc}_{\mathrm{In}}$) may be grounded while Pin 6 ($\mathrm{Osc}_{\mathrm{Out}}$) may be driven at low-power 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

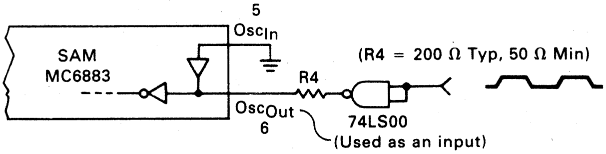
Freq. MHz	CV*	CF*	R1*	R2*	R3*	X1
14.31818	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*
16.0000	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*

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



---

*Optimum values depend on characteristics of the crystal (X1). For many applications, VClk must be 3.579545 MHz ± 50 Hz! Hence, $\mathrm{Osc}_{\mathrm{Out}}$ 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

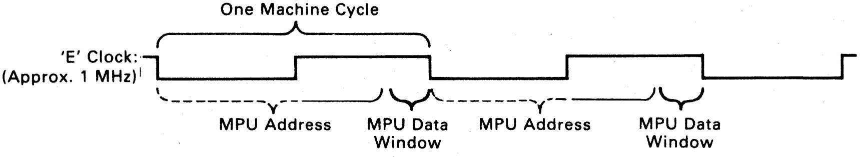
Video or No Video

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)

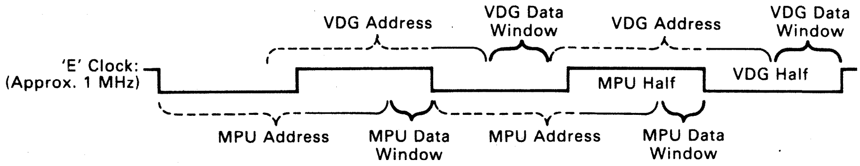
To minimize the number of RAM and interface chips, both the MPU and VDG share common dynamic RAM. Yet, the use of common RAM creates an apparent difficulty. That is, the MPU and VDG must both access the RAM without contention. This difficulty is overcome by taking advantage of the timing and architecture 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, $\overline{HS}$ 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 ($\overline{\mathrm{RAS}}$), column address select ($\overline{\mathrm{CAS}}$) and write enable ($\overline{\mathrm{WE}}$).

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 $\overline{\mathrm{RAS0}}$) the following combinations require no additional address logic.

FIGURE 13 — RAM CONFIGURATIONS

Address: MSB LSB	Chip Select:	
Z5Z4Z3Z2Z1Z0	$\overline{\mathrm{RAS0}}$	One or two banks of 4K x 8 (like MCM4027's)
Z5Z4Z3Z2Z1Z0	$\overline{\mathrm{RAS1}}$ (=Z7)	
Z6Z5Z4Z3Z2Z1Z0	$\overline{\mathrm{RAS0}}$	One or two banks of 16K x 8 (like MCM4116's)
Z6Z5Z4Z3Z2Z1Z0	$\overline{\mathrm{RAS1}}$ (=Z7)	
Z7Z6Z5Z4Z3Z2Z1Z0	$\overline{\mathrm{RAS0}}$	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.

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:

F6	F5	F4	F3	F2	F1	F0	0	0	0	0	0	0	0	0	0

Most Significant Bit

Least Significant Bit

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.)

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. When using Map Type "TY = 1", only the "Slow" MPU rate may be used. Future versions of the SAM may allow use of all rates.

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   0,X+
        20   02   BRA   SAM3
SAM2    A7   81   STA   0,X++
SAM3    5A        DEC   B
        26   F2   BNE   SAM1
        39        RTS
```

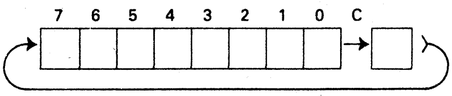
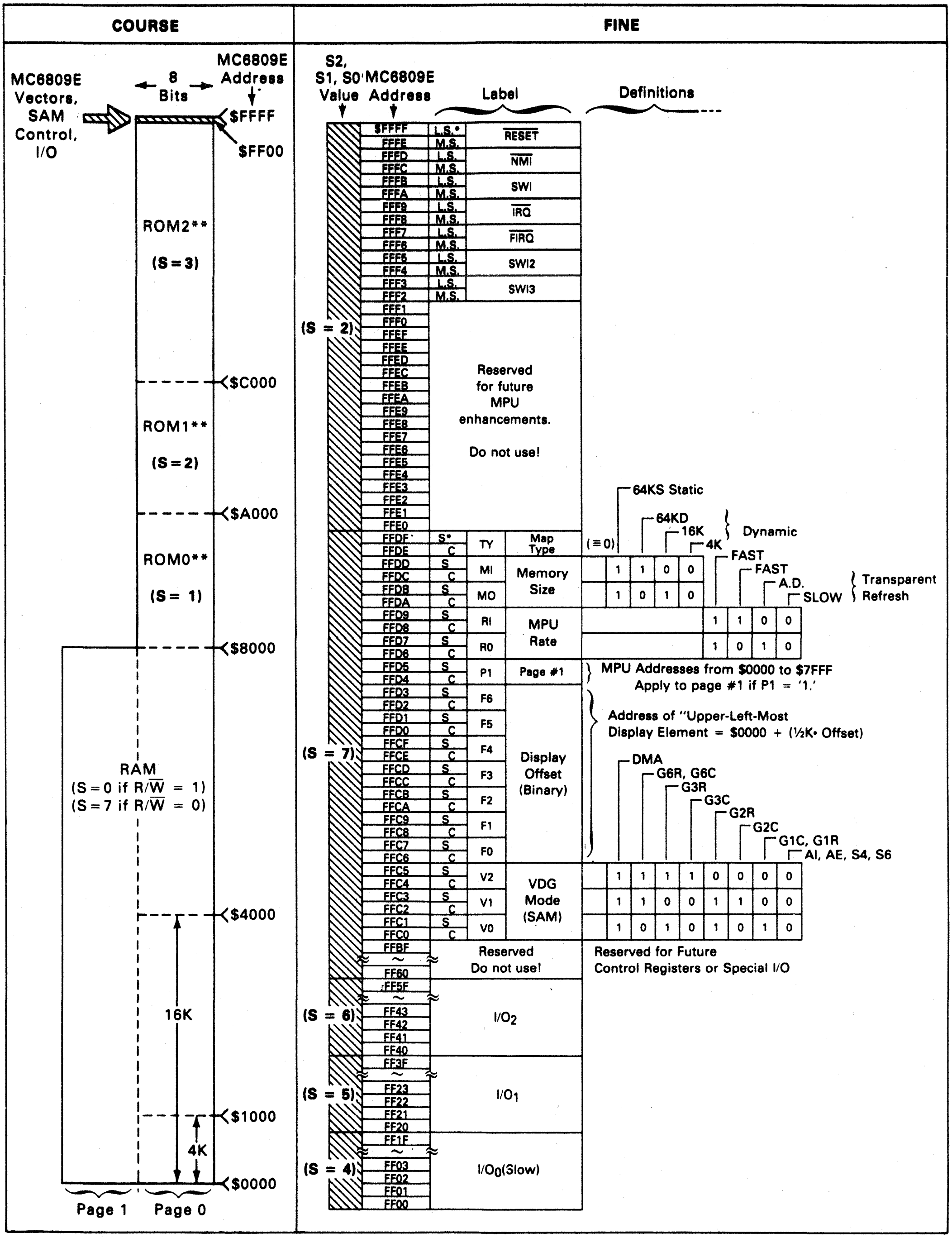


FIGURE 14 — MEMORY MAP (TYPE #0)



*Note:

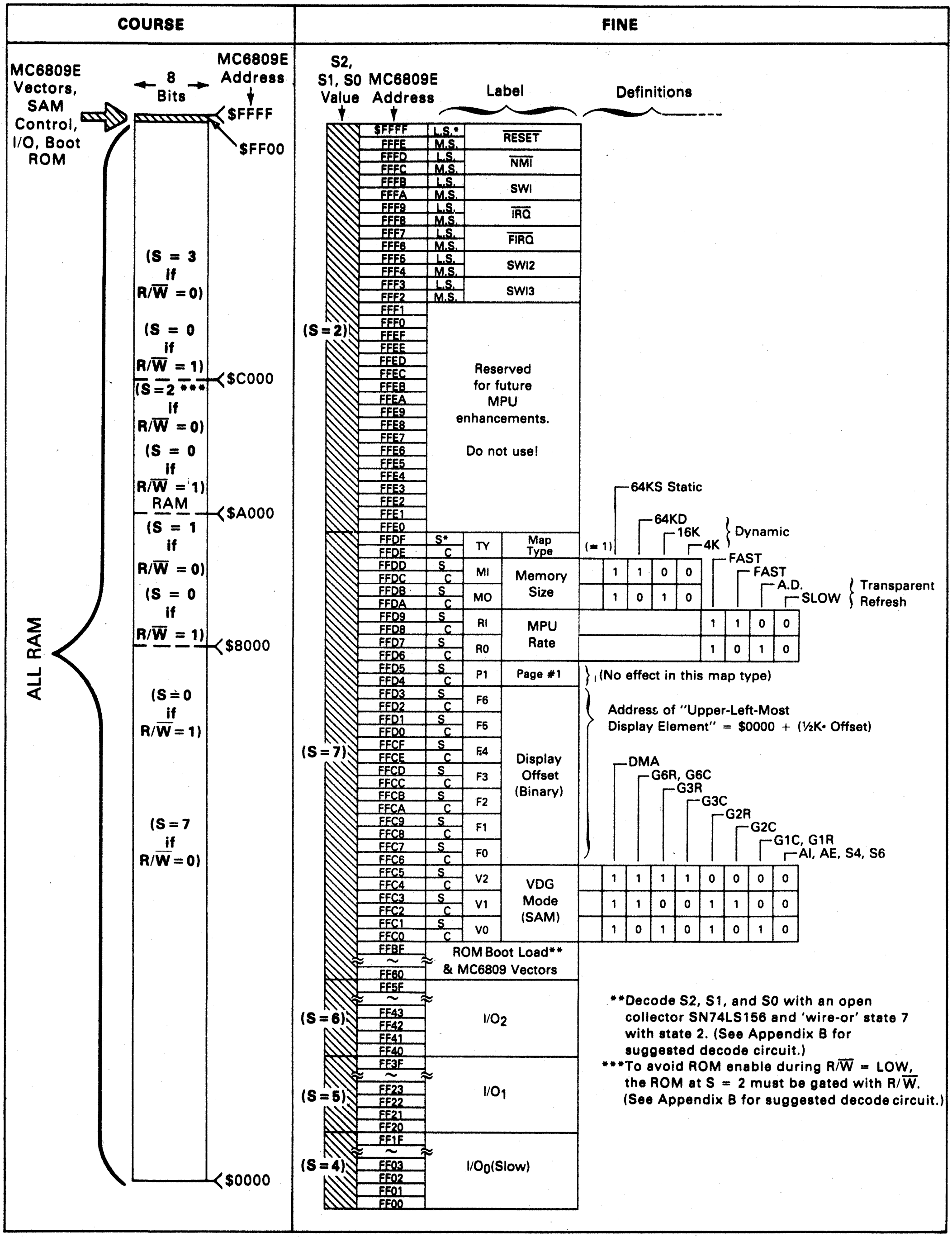
M.S. = Most Significant
L.S. = Least Significant

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)



**Decode S2, S1, and S0 with an open collector SN74LS156 and 'wire-or' state 7 with state 2. (See Appendix B for suggested decode circuit.)

***To avoid ROM enable during R/W = LOW, the ROM at S = 2 must be gated with R/W. (See Appendix B for suggested decode circuit.)

*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 $R/\overline{W}$ = 1 7 if $R/\overline{W}$ = 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 $R/\overline{W}$ = 1	RAM: 64K(– 256) addresses, shared by MPU and VDG. (If $R/\overline{W}$ = 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, 64H*64V, 64H*96V, 64H*192V. 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"

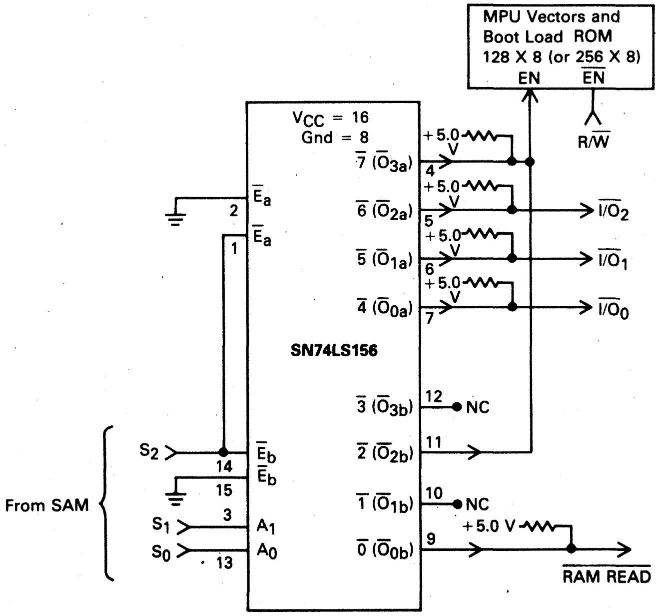
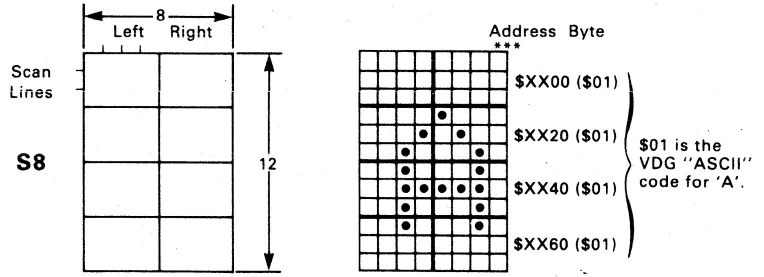


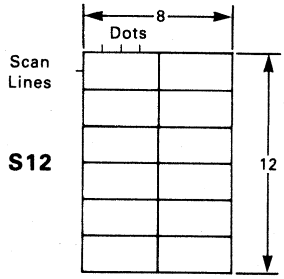
FIGURE 17 — DISPLAY MODES S8, S12, S24
Bit/Visible Dot Correlation

S8



- Alphanumeric Compatible

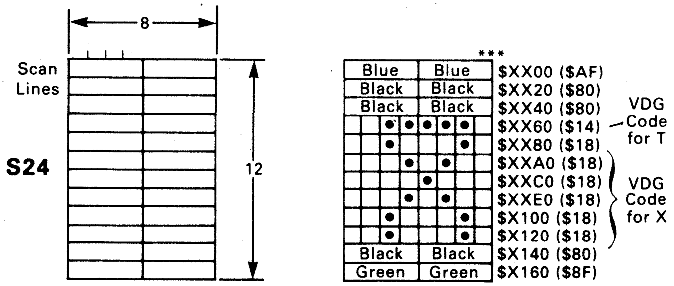
S12



Left	Right ***	
Red	Red	$XX00 ($BF)
Blue	Off	$XX20 ($AA)
Off	Green	$XX40 ($85)
Orange	Orange	$XX60 ($FF)
Off	Off	$XX80 ($80)
Yellow	Yellow	$XXA0 ($9F)

- Options: One of 8 colors for L or R or both. Off = Black

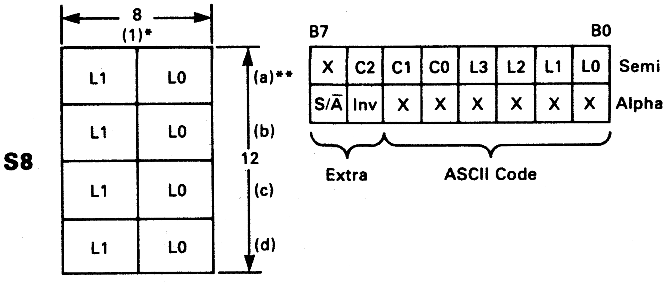
S24



- Underline, Overline
- Mix Character Dot Rows

*** Characters will always remain in standard VDG positions.

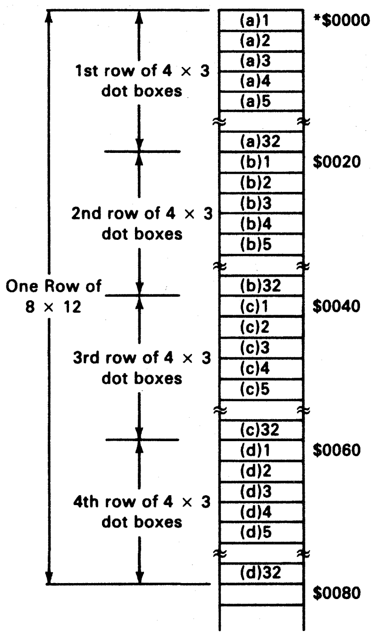
FIGURE 18 — S8 DISPLAY FORMAT EXAMPLES



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	

S8 Screen Memory Map



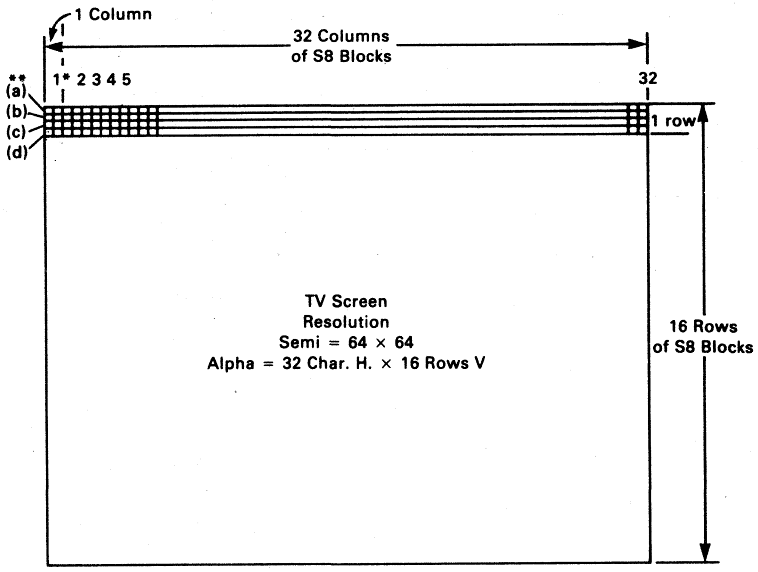
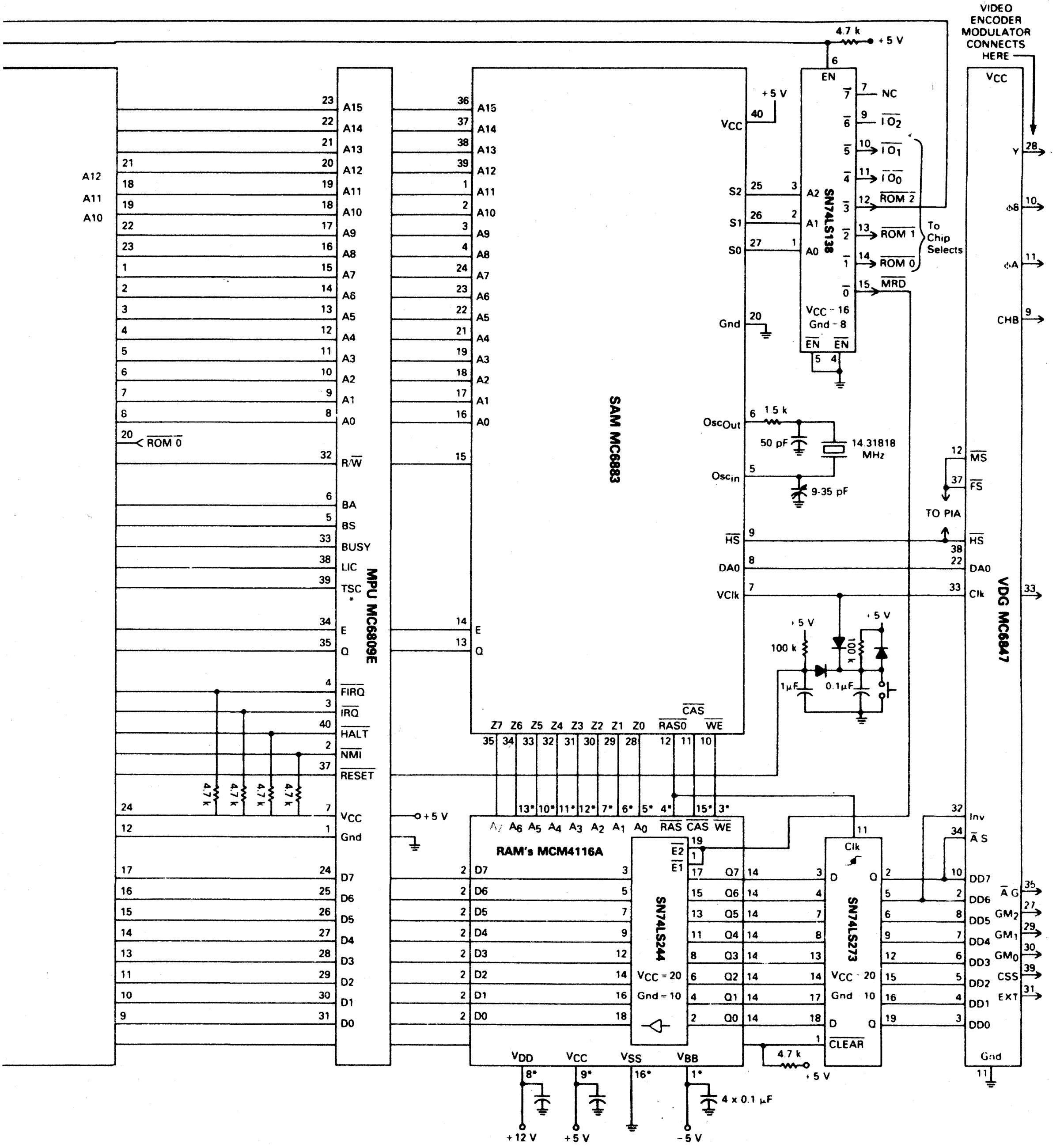


FIGURE 19 — EXAMPLE of MC6809

IC6883 and MC6847 COMPUTER



*This pin number on 8 different RAM chips is connected to this point.

24

FIGURE 20 — EQUIVALENT OF OSCILLATOR INPUT AND OUTPUT

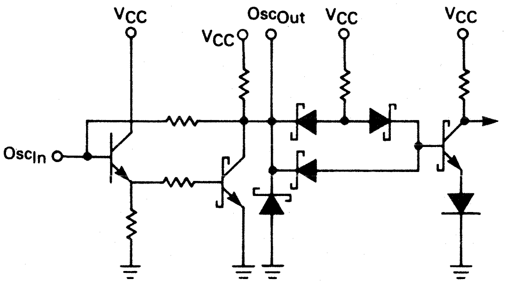


FIGURE 21 — DA0 INPUT

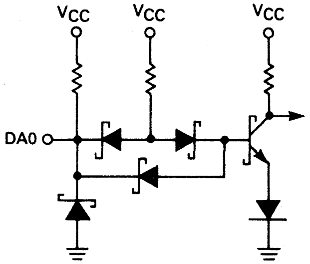


FIGURE 22 — VClk INPUT/OUTPUT

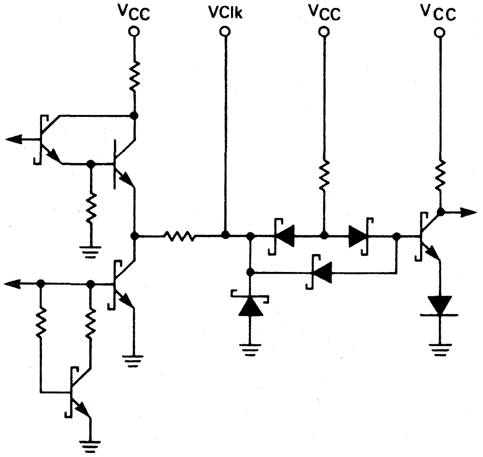


FIGURE 23 — E AND Q OUTPUTS

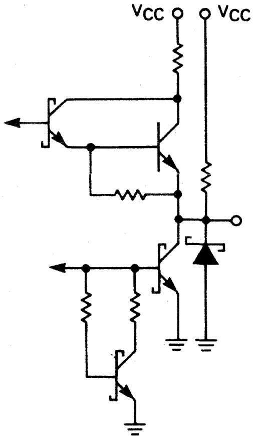


FIGURE 24 — TYPICAL INPUT

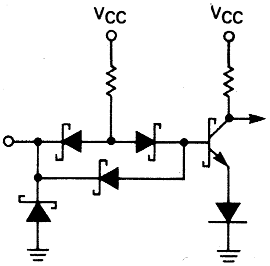
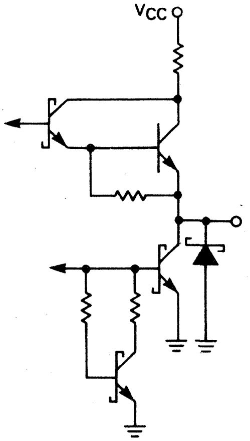


FIGURE 25 — TYPICAL OUTPUT



MOTOROLA

MC6889
MC8T28

This device may be ordered under either of the above type numbers.

NON-INVERTING BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

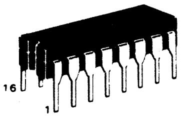
NON-INVERTING QUAD THREE-STATE BUS TRANSCEIVER

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.

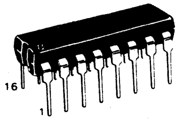
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier 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

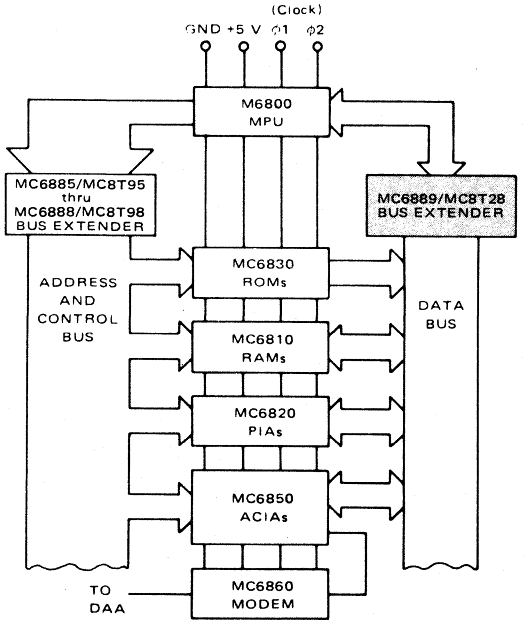


L SUFFIX
CERAMIC PACKAGE
CASE 620

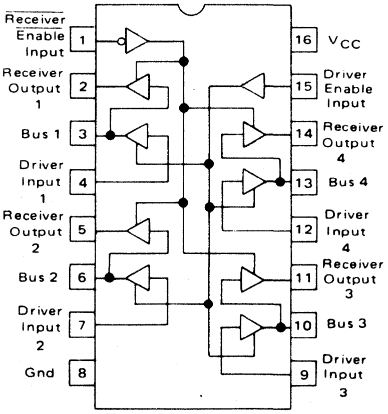


P SUFFIX
PLASTIC PACKAGE
CASE 648

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS – MC6889
MC8T28



ORDERING INFORMATION

Device	Alternate	Temperature Range	Package
MC6889L	MC8T28L	0 to +75°C	Ceramic DIP
MC6889P	MC8T28P	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 $\leq V_{CC} \leq$ 5.25 V and 0°C $\leq T_A \leq$ 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 $I_{IC(\overline{RE})}$ = +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}	–	–	110	mA

(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	

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

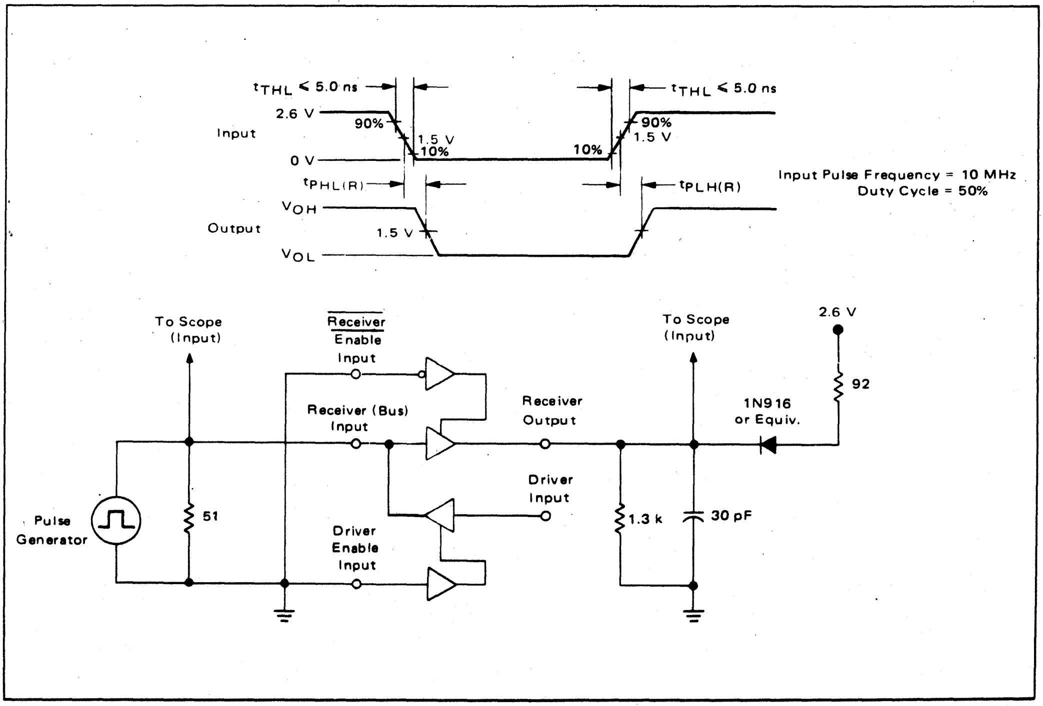


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

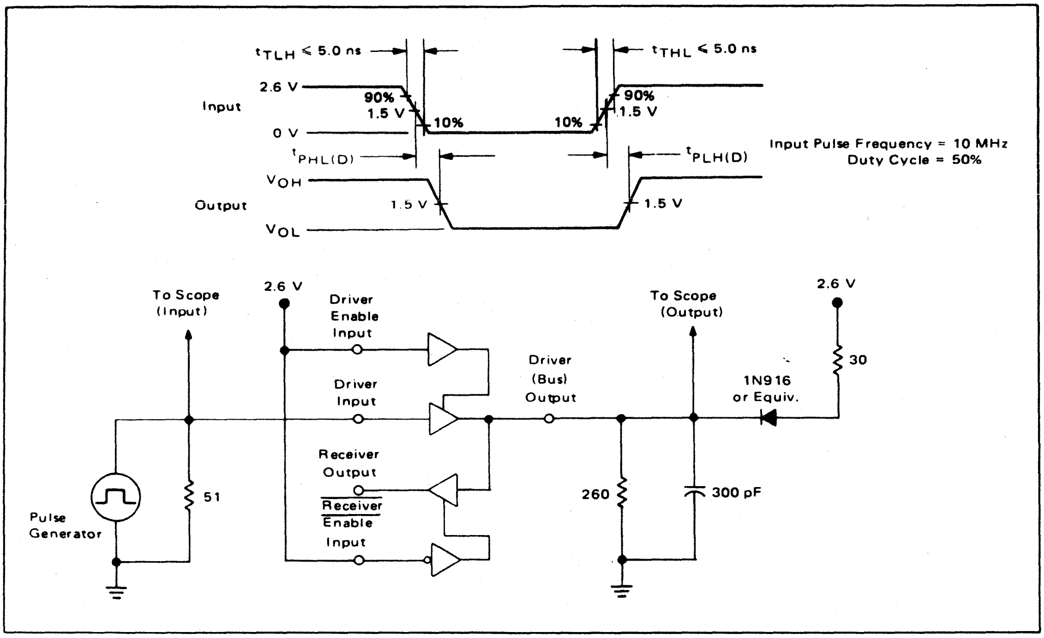


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

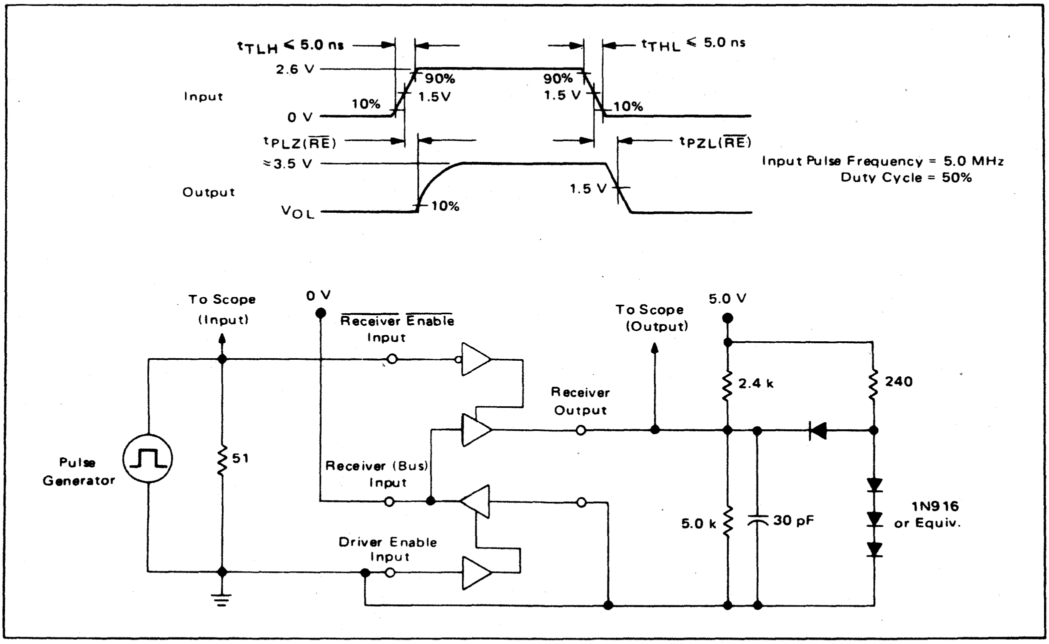


FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

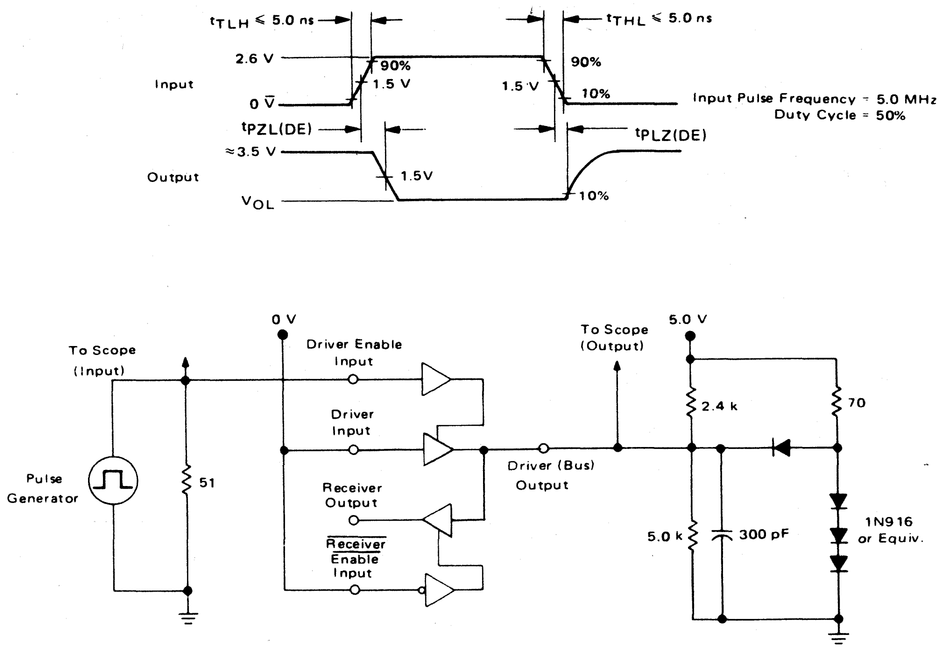
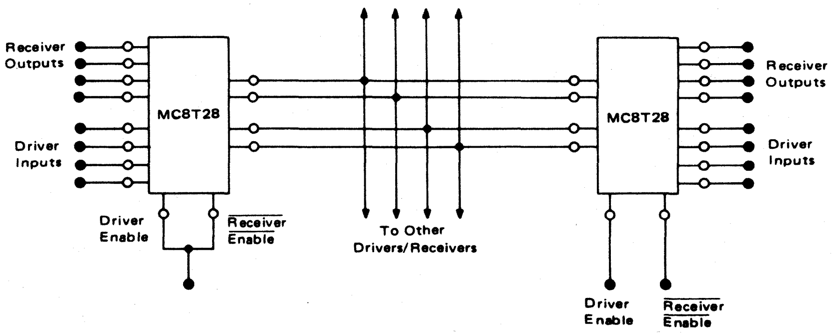
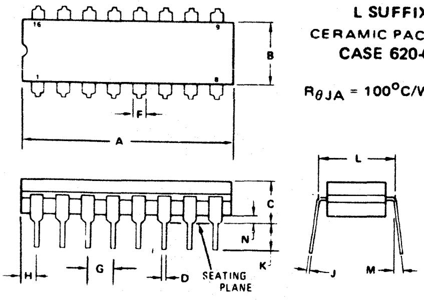


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS





L SUFFIX
CERAMIC PACKAGE
CASE 620-02

$R_{\theta JA}$ = 100°C/W (Typ)

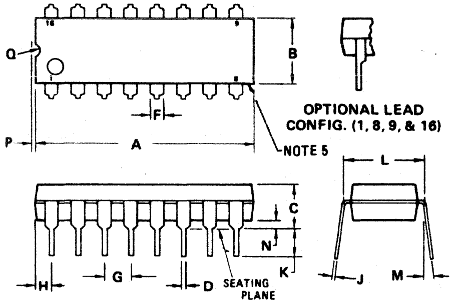
NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

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

CASE 620-02

P SUFFIX
PLASTIC PACKAGE
CASE 648-05

$R_{\theta JA}$ = 100°C/W (Typ)



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.

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

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA(Typ)}}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section
T_A = Maximum Desired Operating Ambient Temperature
$R_{\theta JA(Typ)}$ = Typical Thermal Resistance Junction to Ambient

MOTOROLA

MC14442

Advance Information

CMOS LSI

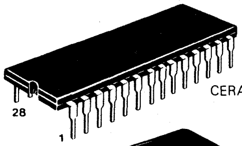
(LOW-POWER SILICON GATE
COMPLEMENTARY MOS)

MICROPROCESSOR-COMPATIBLE ANALOG-TO-DIGITAL CONVERTER

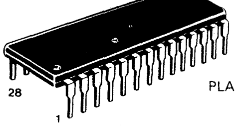
ANALOG-TO-DIGITAL CONVERTER (ADC)

The MC14442 ADC is a 28-pin bus-compatible 8-bit A/D converter with additional digital input capability. The device operates from a single 5 V supply and provides direct interface to the MPU data bus used with all Motorola M6800 family parts. It performs an 8-bit conversion in 32 machine cycles and allows up to 11 analog inputs. In addition, the part can accept up to 6 digital inputs. These inputs are designed to be either analog or digital inputs. All necessary logic for software configuration, channel selection, conversion control and bus interface is included.

- Direct Interface to M6800 Family MPUs
- Dynamic Successive Approximation A/D
- 32 μs Conversion at $f_E = 1.0$ MHz
- Ratiometric Conversion
- Completely Programmable
- Completely Software Compatible with the MC14444 ADC
- 5 Dedicated Analog Inputs
- 6 Inputs Usable for Either Analog or Digital Signals
- Completely TTL Compatible Inputs at Full Speed with Supply Voltage of 5 V ± 10%



L SUFFIX
CERAMIC PACKAGE
CASE 733

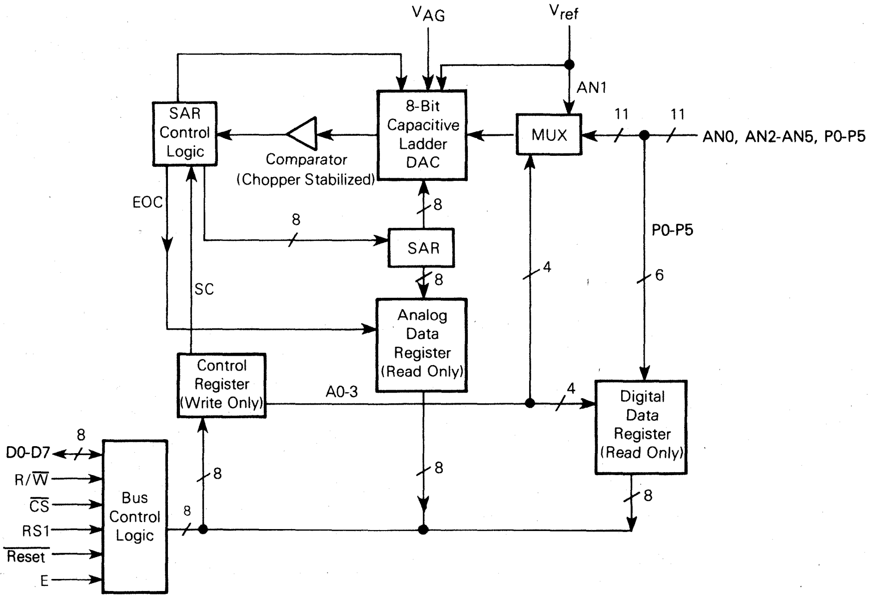


P SUFFIX
PLASTIC PACKAGE
CASE 710

ORDERING INFORMATION

MC14XXX
Suffix Denotes
L Ceramic Package
P Plastic Package

BLOCK DIAGRAM



ADI-941

ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.5 to +6.5	Vdc
Input Voltage	V_{in}	−0.5 to V_{DD} +0.5	Vdc
Input Current, Per Pin	I_{in}	±10	mA
Input Current, Per Package	I_{in}	±20	mA
Operating Temperature Range (Ambient)	T_A	−40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 V ±10%, V_{SS} = 0 V, T_A = T_L to T_H unless otherwise noted)

Characteristic	Symbol	Conditions	Min	Max	Unit
Bus Control Inputs (R/$\overline{W}$, Enable, $\overline{Reset}$, RS1, $\overline{CS}$)					
Input High Voltage	V_{IH}		2.0	–	Vdc
Input Low Voltage	V_{IL}		–	0.8	Vdc
Input Leakage Current	I_{in}	V_{in} = 0 to 5.5 Vdc	–	±1	μAdc
Data Bus (D0-D7)					
Input High Voltage	V_{IH}		2.0	–	Vdc
Input Low Voltage	V_{IL}		–	0.8	Vdc
Three-State (Off State) Input Leakage Current	I_{TSI}	V_{DD} = 5.5 Vdc, $V_{SS} \leq V_{in} \leq V_{DD}$	–	±10	μAdc
Ouput High Voltage	V_{OH}	I_{OH} = −1.6 mA	2.4	–	V
Output Low Voltage	V_{OL}	I_{OL} = 1.6 mA	–	0.4	V
Peripheral Inputs (P0-P5)					
Input High Voltage	V_{IH}		2.0	–	Vdc
Input Low Voltage	V_{IL}		–	0.8	Vdc
Input Leakage Current	I_{in}	V_{DD} = 5.5 Vdc, $V_{SS} \leq V_{in} \leq V_{DD}$	–	±1.0	μAdc
Current Requirements					
Supply Current	I_{DD}	V_{DD} = 5.5 Vdc	–	5	mAdc

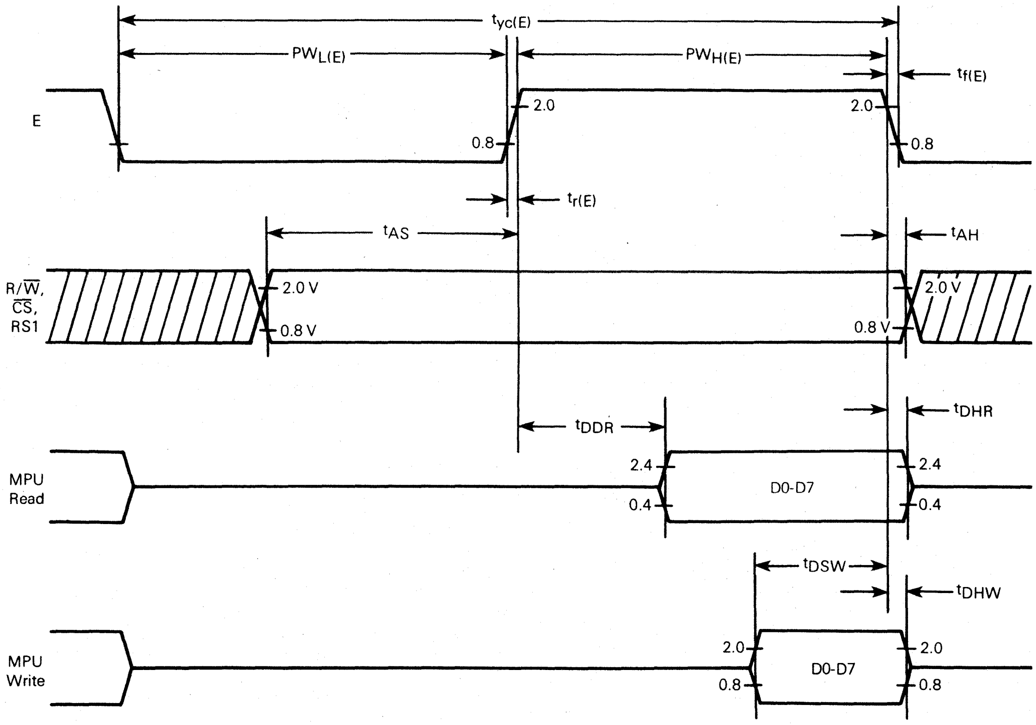
ANALOG CHARACTERISTICS

Characteristic	Description	Min	Max	Unit
Analog Multiplexer				
Leakage Current	Leakage current between all deselected analog inputs and any selected analog input with all analog input voltages between V_{SS} and V_{DD}	–	±400	nAdc
A/D Converter ($V_{SS}=0$ V, $V_{AG}=0$ V, 4.5 V $\le V_{ref} \le V_{DD} \le$ 5.5 V, $T_A=-40°$ to $+85°C$)				
Resolution	Number of bits resolved by the A/D	8	–	Bits
Nonlinearity	Maximum deviation from the best straight line through the A/D transfer characteristic	–	± ½	LSB
Zero Error	Difference between the output of an ideal and an actual A/D for zero input voltage	–	± ½	LSB
Full-Scale Error	Difference between the output of an ideal and an actual A/D for full-scale input voltage	–	± ½	LSB
Total Unadjusted Error	Maximum sum of Nonlinearity, Zero Error, and Full-Scale Error	–	± ½	LSB
Quantization Error	Uncertainty due to converter resolution	–	± ½	LSB
Absolute Accuracy	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code, all error sources included	–	± 1.0	LSB
Conversion Time	Total time to perform a single analog-to-digital conversion	–	32	E cycles
Sample Acquisition Time	Time required to sample the analog input	–	12	E cycles

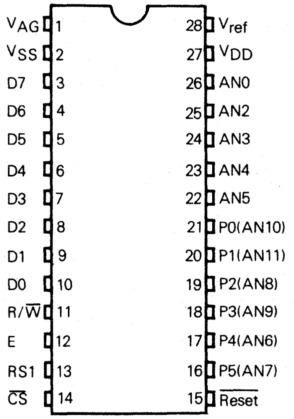
AC CHARACTERISTICS (See Figure 1)

Characteristic	Signal	Symbol	Min	Max	Unit
Enable Clock Cycle Time ($1/f_E$)	E	$t_{cyc(E)}$	943	–	ns
Enable Clock Pulse Width, High	E	$PW_{H(E)}$	440	–	ns
Enable Clock Pulse Width, Low	E	$PW_{L(E)}$	410	–	ns
Clock Rise Time	E	$t_{r(E)}$	–	25	ns
Clock Fall Time	E	$t_{f(E)}$	–	30	ns
Address Setup Time	RS1, $R/\overline{W}$, $\overline{CS}$	t_{AS}	145	–	ns
Data Delay (Read)	D0-D7	t_{DDR}	–	335	ns
Data Setup (Write)	D0-D7	t_{DSW}	185	–	ns
Address Hold Time	RS1, $R/\overline{W}$, $\overline{CS}$	t_{AH}	10	–	ns
Input Data Hold Time	D0-D7	t_{DHW}	10	–	ns
Output Data Hold Time	D0-D7	t_{DHR}	10	–	ns

FIGURE 1 — BUS TIMING



MC14442 PIN ASSIGNMENT



PIN FUNCTIONS

Pin No.	Pin Name	Function	Type
1	V_{AG}	A/D Converter Analog Ground	Input
2	V_{SS}	Digital Ground	Input
3	D7	Data Bus Bit 7 (MSB)	Input/Output
4	D6	Data Bus Bit 6	Input/Output
5	D5	Data Bus Bit 5	Input/Output
6	D4	Data Bus Bit 4	Input/Output
7	D3	Data Bus Bit 3	Input/Output
8	D2	Data Bus Bit 2	Input/Output
9	D1	Data Bus Bit 1	Input/Output
10	D0	Data Bus Bit 0 (LSB)	Input/Output
11	R/$\overline{W}$	Read/Write	Input
12	E	Enable Clock (φ2)	Input
13	RS1	Register Select	Input
14	$\overline{CS}$	Chip Select	Input
15	$\overline{Reset}$	Reset	Input
16	P5(AN7)	Digital Port or Analog Channel 7	Input
17	P4(AN6)	Digital Port or Analog Channel 6	Input
18	P3(AN9)	Digital Port or Analog Channel 9	Input
19	P2(AN8)	Digital Port or Analog Channel 8	Input
20	P1(AN11)	Digital Port or Analog Channel 11	Input
21	P0(AN10)	Digital Port or Analog Channel 10	Input
22	AN5	Analog Channel 5	Input
23	AN4	Analog Channel 4	Input
24	AN3	Analog Channel 3	Input
25	AN2	Analog Channel 2	Input
26	AN0	Analog Channel 0	Input
27	V_{DD}	Supply Voltage	Input
28	V_{ref}	A/D Converter Positive Reference Voltage	Input

MC14442 MPU INTERFACE SIGNALS

Bidirectional Data Bus (D0-D7) — The bidirectional data lines D0-D7 comprise the bus over which data is transferred in parallel to and from the MPU. The data bus output drivers are three-state devices that remain in the high-impedence state except during an MPU read of an ADC data register.

Enable Clock (E) — The enable clock provides two functions for the MC14442. First, it serves to synchronize data transfers into and out of the ADC. The timing of all other external signals is referenced to the leading or trailing edge of the enable clock. Secondly, the enable clock is used internally to derive the necessary SAR A/D conversion clocks. Because this conversion is a dynamic process, enable clock must be a continuous signal into the ADC during an A/D conversion.

Read/Write (R/$\overline{W}$) — The R/$\overline{W}$ signal is provided to the MC14442 to control the direction of data transfers to and from the MPU. A low state on this line is required to transfer data from the MPU to the ADC control register. A high state is required on R/$\overline{W}$ to transfer data out of either of the ADC data registers.

Reset ($\overline{Reset}$) — The reset line supplies the means of externally forcing the MC14442 into a known state. When a low is applied to the $\overline{Reset}$ pin, the start conversion bit of the control register is cleared. Analog channel 0 is automatically selected by the analog multiplexer. The A/D status bit is also cleared. Any A/D results present in the Analog Data register are not affected by a reset. Reset forces the data bus output drivers to the high-impedance state. The internal byte pointer (discussed in the following pages) is set to point to the most significant byte of any subsequently selected internal register. In order to attain an internally stable reset state, the $\overline{Reset}$ pin must be low for at least one complete enable clock cycle.

Chip Select ($\overline{CS}$) — Chip select is an active low input used by the MPU system to enable the ADC for data transfers. No data may be passed to or from the ADC through the data bus pins unless $\overline{CS}$ is in a low state. A selection of MPU address lines and the M6800 VMA signal or its equivalent should be utilized to provide chip select to the MC14442.

MC14442 ANALOG INPUTS AND DIGITAL INPUTS

(Refer to the ADC Block Diagram)

Dedicated Analog Channels (AN0, AN2-AN5) — These input pins serve as dedicated analog channels subject to A/D conversions. These channels are fed directly into the internal 12-to-1 analog multiplexer which feeds a single analog voltage to the A/D converter.

Shared Analog Channels (AN6-AN11) — These input pins are also connected to the analog multiplexer and may be used as analog channels for A/D conversion. However, these pins may also serve as digital input pins as described next.

Shared Digital Inputs (P0-P5) — P0-P5 comprise a 6-bit digital input port whose bits may also serve as analog channels. The state of these inputs may be read at any time from the ADC digital data register. The function of these pins is not programmed, but instead is simply assigned by the system designer on a pin-by-pin basis.

> **CAUTION:** Digital values read from the P0-P5 bit locations do not guarantee the presence of true digital input levels on these pins. P0-P5 pass through a TTL-compatible input buffer and into the digital data register. These buffers are designed with enough hysteresis to prevent internal oscillations if an analog voltage between 0.8 and 2 V is present on one or more of these six pins.

MC14442 SUPPLY VOLTAGE PINS

Positive Supply Voltage (V_{DD}) — V_{DD} is used internally to supply power to all digital logic and to the chopper stabilized comparator. Because the output buffers connected to this supply must drive capacitive loads, ac noise on this supply line is unavoidable internally. Analog circuits using this supply within the MC14442 were designed with high V_{DD} supply rejection; however, it is recommended that a filtering capacitance be used externally between V_{DD} and V_{SS} to filter noise caused by transient current spikes.

Ground Supply Voltage (V_{SS}) — V_{SS} should be tied to system digital ground or the negative terminal of the V_{DD} power source. Again, the output buffers cause internal noise on this supply, so analog circuits were designed with high V_{SS} rejection.

Positive A/D Reference Voltage (V_{ref}) — This is the voltage used internally to provide references to the analog comparator and the digital-to-analog converter used by the SAR A/D. The analog-to-digital conversion result will be ratiometric to $V_{ref} - V_{AG}$ (full scale). Hence V_{ref} should be a very noise-free supply. Ideally V_{ref} should be single-point connected to the voltage supply driving the system's transducers. V_{ref} may be connected to V_{DD}, but degradation of absolute A/D accuracy may result due to switching noise on V_{DD}.

A/D Ground Reference Voltage (V_{AG}) — This supply is the ground reference for the internal DAC and several reference voltages supplied to the comparator. It should also be noise-free to guarantee A/D accuracy. Absolute accuracy may be degraded if V_{AG} is wired to V_{SS} at the ADC package unless V_{SS} has been sufficiently filtered to remove switching noise. Ideally V_{AG} should be single-point grounded to the system analog ground supply.

MC14442 INTERNAL REGISTERS

The MC14442 ADC has three 16-bit internal registers. Each register is divided into two 8-bit bytes: a most significant (MS) byte (bits 8-15) and a least significant (LS) byte (bits 0-7). Each of these bytes may not be addressed externally, but instead are normally addressed by a single 16-bit instruction such as the M6800 LDX instruction. An internal byte pointer selects the appropriate register byte during the two E cycles of a normal 16-bit access. In keeping with the M6800 X register format, the pointer points first to the MS byte of any selected register. After the E cycle in which the MS byte is accessed, the pointer will switch to the LS byte and remain there for as long as chip select is low. The pointer moves back to the MS byte on the falling edge of E after the first complete E cycle in which the ADC is not selected. (See Figure 2a for more detail.) The MS byte of any register may also be accessed by a simple 8-bit instruction as shown in Figure 2b. However, the LS byte of all registers may be accessed only by 16-bit instructions as described above. By connecting the ADC register select (RS1) to the MPU address line A1, the three registers may be accessed sequentially by 16-bit operations.

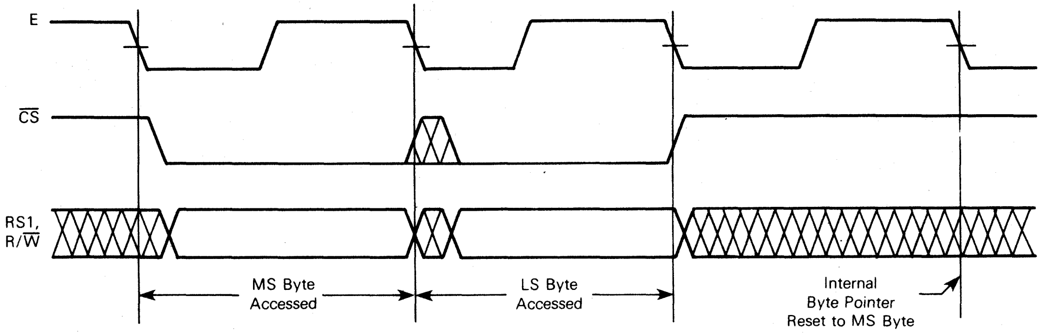
> **CAUTION:** RS1 should **not** be connected to address line A0 and the addressing of the ADC should be such that RS1 does not change states during a 16-bit access.

INTERNAL REGISTER ADDRESSING

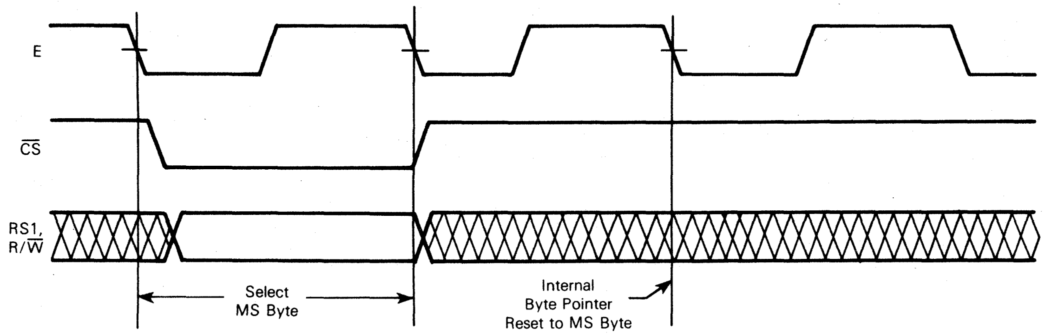
Addressing Signals				
$\overline{\text{Reset}}$	$\overline{\text{CS}}$	$R/\overline{W}$	RS1	ADC Response
0	X	X	X	Reset
1	0	0	0	No Response
1	0	0	1	MPU Write to Control Register
1	0	1	0	MPU Read from Analog Data Register
1	0	1	1	MPU Read from Digital Data Register
1	1	X	X	Chip Deselected (No Response)

FIGURE 2 — ADC ACCESS TIMING

a — Typical 16-Bit ADC Access



b — Typical 8-Bit ADC Access



MC14442 CONTROL REGISTER
(Write Only)

15							8	7							0
X (MSB)	X	X	X	X	X	X	SC (LSB)	X (MSB)	X	X	X	A3	A2	A1	A0 (LSB)

←Most Significant Byte→←Least Significant Byte→

←8-Bit Write→

←16-Bit Write→

Analog Multiplexer Address (A0-A3) – These four address bits are decoded by the analog multiplexer and used to select the appropriate analog channel as shown below.

Hexadecimal Address (A3 = MSB)	Select
0	AN0
1	V_{ref}
2-5	AN2-AN5
6-B	AN6-AN11
C-F	Undefined

Start A/D Conversion (SC) – When the SC bit is set to a logical 1, an A/D conversion on the specified analog channel will begin immediately after the completion of the control register write.

Unused Bits (X) – Bits 4-7 and 9-15 of the ADC Control Register are not used internally.

NOTE: A 16-bit control register write is required to change the analog multiplexer address. However, 8-bit writes to the MC14442 can be used to initiate an A/D conversion if the analog MUX is already selecting the desired channel. This is useful when repeated conversions on a particular analog channel are necessary.

MC14442 ANALOG DATA REGISTER
(Read Only)

15							8	7							0
EOC (MSB)	0	0	0	0	0	0	0 (LSB)	R7 (MSB)	R6	R5	R4	R3	R2	R1	R0 (LSB)

←Most Significant Byte→←Least Significant Byte→

←8-Bit Read→

←16-Bit Read→

A/D Result (R0-R7) – The LS byte of the analog data register contains the result of the A/D conversion. R7 is the MSB, and the converter follows the standard convention of assigning a code of $FF to a full-scale analog voltage. There are no special overflow or underflow indications.

A/D Status (EOC) – The A/D status bit is set whenever a conversion is successfully completed by the ADC. The status bit is cleared by either an 8-bit or a 16-bit MPU write to the ADC control register. The remainder of the bits in the MS byte of the analog data register are always set to a logical 0 to simplify MPU interrogation of the ADC status. For example, a single M6800 TST instruction can be used to determine the status of the A/D conversion.

MC14442 DIGITAL DATA REGISTER
(Read Only)

15							8	7							0
P5	P4	P3	P2	P1	P0	A3	A2	A1	A0	0	0	0	0	0	0

←Most Significant Byte→←Least Significant Byte→

←8-Bit Read→

←16-Bit Read→

Logical Zero (0) – These bits are always read as logical zero.

Analog Multiplexer Address (A0-A3) – The number of the analog channel presently addressed is given by these bits.

Shared Digital Port (P0-P5) – The voltage present on these pins is interpreted as a digital signal and the corresponding states are read from these bits.

WARNING: A digital value will be given for each pin even if some or all of the pins are being used as analog inputs.

ANALOG SUBSYSTEM
(See Block Diagram)

General Description

The analog subsystem of the MC14442 is composed of a 12-channel analog multiplexer, an 8-bit capacitive DAC (digital-to-analog converter), a chopper-stabilized comparator, a successive approximation register, and the necessary control logic to generate a successive approximation routine.

The analog multiplexer selects one of twelve channels and directs it to the input of the capacitive DAC. A fully-capacitive DAC is utilized because of the excellent matching characteristics of thin-oxide capacitors in the silicon-gate CMOS process. The DAC actually serves several functions. During the sample phase, the analog input voltage is applied to the DAC which acts as a sample-and-hold circuit. During the conversion phase, the capacitor array serves as a digital-to-analog converter. The comparator is the heart of the ADC; it compares the unknown analog input to the output of the DAC, which is driven by a conventional successive-approximation register. The chopper-stabilized comparator was designed for low offset voltage characteristics as well as V_{DD} and V_{SS} power supply rejection.

Device Operation

An A/D conversion is initiated by writing a logical 1 into the SC bit of the ADC control register. The MC14442 allows 2 enable clock cycles for the write into the control register even if only one byte is written. In this case, the second E cycle does not affect any internal registers. During the next 12½ enable cycles following a write command, the analog multiplexer channel is selected and the analog input voltage is stored on the sample and hold DAC. It is recommended that an input source impedance of 10 KΩ or less be used to allow complete charging of the capacitive DAC.

During cycle 13 the A/D is disconnected from the multiplexer output and the successive approximation A/D routine begins. Since the analog input voltage is being held on an internal capacitor for the entire conversion period, it is required that the enable clock run continuously until the A/D conversion is completed. The new 8-bit result is latched into the analog data register on the rising edge of cycle 32. At this point the end of conversion bit (EOC) is set in the analog data register MS byte. (See Figure 3, A/D Timing Sequence.)

NOTE: The digital data register or the analog data register may be read even if an A/D conversion is in progress. If the analog data register is read during an A/D conversion, valid results from the previous conversion are obtained. However, the EOC bit will be clear (logic 0) if an A/D conversion is in progress.

FIGURE 3 — A/D TIMING SEQUENCE

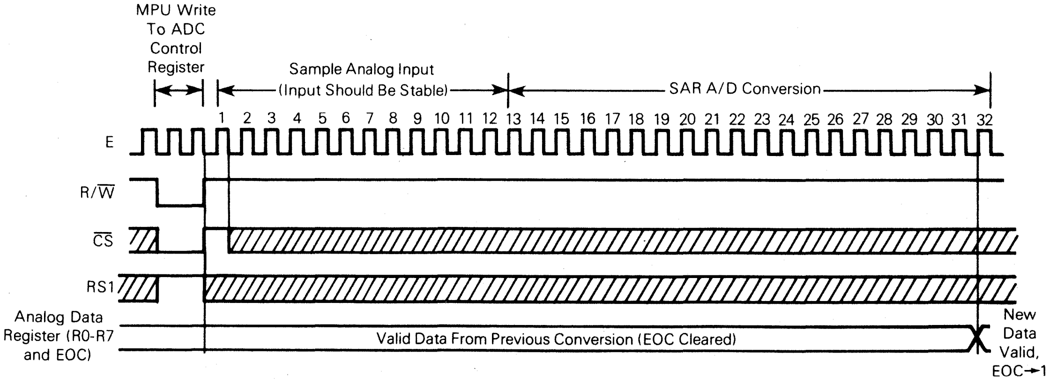
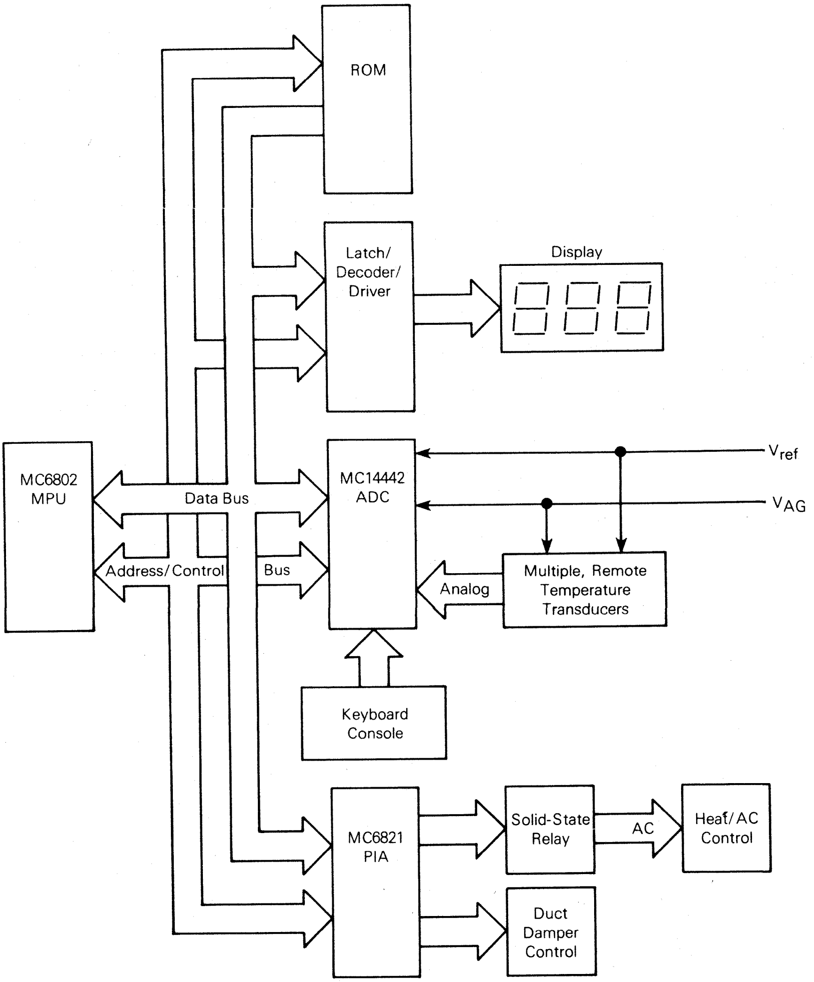


FIGURE 4 — TYPICAL MC14442 APPLICATION IN A CLIMATE CONTROLLER



MOTOROLA

MC68008

Product Preview

THE MC68008 REDUCED DATA BUS MC68000

Mainframe architecture is now available for 8-bit systems. The new MC68008 member of the M68000 Family allows the design of cost effective systems using 8-bit data buses while providing the benefits of a 32-bit microprocessor architecture. The performance of the MC68008 is greater than any 8-bit microprocessor and superior to several 16-bit microprocessors.

The MC68008 is completely code compatible with the MC68000. This means that programs developed for the MC68000 will run on the MC68008 and vice versa. This applies equally to either source code or object code.

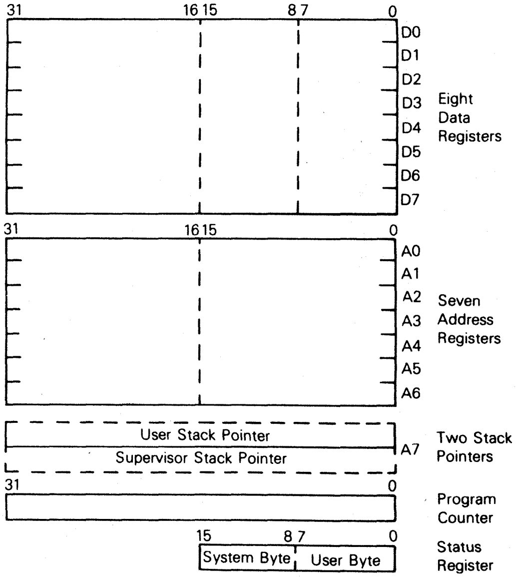
A system implementation based on an 8-bit data bus reduces system cost in comparison to 16-bit systems due to a more effective use of components. Byte-wide memories and peripherals can be used much more effectively. In addition, the non-multiplexed address and data buses eliminate the need for external demultiplexers, thus further simplifying the system.

The general purpose architecture of the MC68008 can easily satisfy market requirements for many diverse applications. The MC68008 provides the user with seventeen 32-bit registers, 56 basic instruction types, allows an efficient hardware system implementation, extensive exception processing capabilities and a 1 megabyte linear address space. The MC68008 will first be introduced in a 48 pin DIP.

REGISTER STRUCTURE

As shown in Figure 1, the programmer's model is identical to that of the MC68000, with seventeen 32-bit registers, a 32-bit program counter, and a 16-bit status register. The first eight registers (D0-D7) are used as data registers for bit, byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0-A6) and the system stack pointer (A7) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. All of the seventeen registers may be used as index registers.

FIGURE 1 — PROGRAMMING MODEL

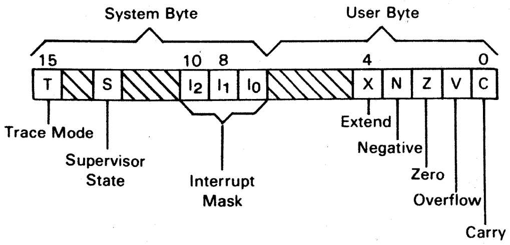


The system stack is used by many instructions. The 14 addressing modes allow the creation of user stack and queues. The system stack pointer is either the supervisor stack pointer (SSP) or the user stack pointer (USP), depending on the state of the S bit in the status register. If the S bit is set, indicating that the processor is in the supervisor state, then the SSP is the active system stack pointer and the USP is not used. If the S bit is low, indicating that the processor is in the user state, then the USP is the active system stack pointer and the SSP is protected from user modification.

The large non-segmented linear address space (1 megabyte) of the MC68008 allows large modular programs to be developed and executed efficiently. A large linear address space allows program segment sizes to be determined by the application rather than forcing the designer to adopt an arbitrary segment size without regard to his individual requirements.

The status register, shown in Figure 2, may be considered as two bytes: the user byte and the system byte. The user byte contains five bits defining the overflow (V), zero (Z), negative (N), carrry (C), and extend (X) condition codes. The system byte contains five bits. Three bits are used to define the current interrupt priority; any interrupt level higher than the current mask level will be recognized. (Note that the MC68008 supports three interrupt levels—2, 5, and 7 and that level 7 interrupts are non-maskable—that is, level 7 interrupts are always processed.) Two additional bits indicate whether the processor is in the trace (T) mode and/or in the supervisor (S) state. Ample space remains in the status register for future extensions of the M68000 Family.

FIGURE 2 — STATUS REGISTER



INSTRUCTION SET

The MC68008 is completely code compatible with the MC68000. This means that programs developed for the MC68000 will run on the MC68008 and vice versa. This applies equally to either source code or object code.

The instruction set was designed to minimize the number of mnemonics remembered by the programmer. To further reduce the programmer's burden, the addressing modes are orthogonal.

The instruction set, shown in Table 1, forms a set of programming tools that include all processor functions to perform data movement, integer arithmetic, logical operations, shift and rotate operations, bit manipulation, BCD operations, and both program and system control. Some additional instructions are variations or subsets of these and appear in Table 2.

TABLE 1 — INSTRUCTION SET

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
B$_{CC}$	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DB$_{CC}$	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
S$_{CC}$	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

TABLE 2 — VARIATIONS OF INSTRUCTION TYPES

Instruction Type	Variation	Description
ADD	**ADD**	Add
	ADDA	Add Address
	ADDQ	Add Quick
	ADDI	Add Immediate
	ADDX	Add with Extend
AND	**AND**	Logical And
	ANDI	And Immediate
	ANDI to CCR	And Immediate to Condition Code
	ANDI to SR	And Immediate to Status Register
CMP	**CMP**	Compare
	CMPA	Compare Address
	CMPM	Compare Memory
	CMPI	Compare Immediate
EOR	**EOR**	Exclusive Or
	EORI	Exclusive Or Immediate
	EORI to CCR	Exclusive Or Immediate to Condition Code
	EORI to SR	Exclusive Or Immediate to Status Register

Instruction Type	Variation	Description
MOVE	**MOVE**	Move
	MOVEA	Move Address
	MOVEQ	Move Quick
	MOVE from SR	Move from Status Register
	MOVE to SR	Move to Status Register
	MOVE to CCR	Move to Condition Codes
	MOVE USP	Move User Stack Pointer
NEG	**NEG**	Negate
	NEGX	Negate with Extend
OR	**OR**	Logical Or
	ORI	Or Immediate
	ORI to CCR	Or Immediate to Condition Code
	ORI to SR	Or Immediate to Status Register
SUB	**SUB**	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBQ	Subtract Quick
	SUBX	Subtract with Extend

Each instruction, with few exceptions, operates on bytes, words, and long words. The instructions vary from one to five words in length. The length of the instruction and the operation to be performed are specified by the first word of the instruction. The first word is called the operation word. The remaining words specify the operands. These words are either immediate operands or extensions to the effective address mode specified in the operation word.

Most instructions can use any of the 14 addressing modes listed in Table 3. These addressing modes consist of six basic types: register direct, absolute, program counter relative, register indirect, immediate, and implied. The register indirect addressing modes also have the capability to perform postincrementing, predecrementing, offsetting, and indexing. The program counter relative mode may be used in combination with indexing and offsetting for writing relocatable programs.

TABLE 3 — ADDRESSING MODES

Register Direct Data Register Direct Address Register Direct	**Register Indirect** Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset
Absolute Data Absolute Short Absolute Long	**Immediate Data** Immediate Quick Immediate
Program Counter Relative with Offset Relative with Index and Offset	**Implied** Implied Register

The combination of instruction types, data types, and addressing modes provides over 1000 useful instructions. These instructions include signed and unsigned multiply and divide, quick arithmetic operations, BCD arithmetic, and expanded operations (through traps).

HARDWARE IMPLEMENTATION

The MC68008 is an embodiment of several innovative concepts. Communication with memory (note that I/O is memory mapped) occurs via a high-performance non-multiplexed asynchronous address and data bus. The proprietary microcode of the MC68008 simplifies the design and provides expandability of the processor. Exception processing is extensively used to preserve the integrity of the machine.

Figure 3 shows the functional grouping of the MC68008 signal lines.

FIGURE 3 — INPUT AND OUTPUT SIGNALS

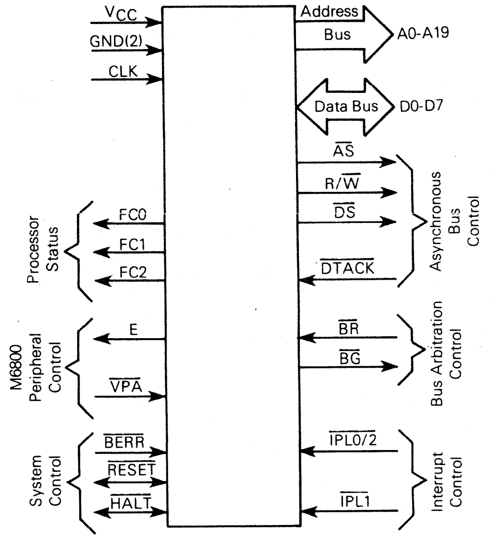
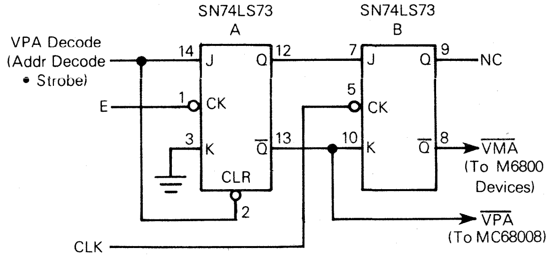


FIGURE 4 — $\overline{\text{VMA}}$ GENERATION



The asynchronous bus structure allows flexibility in choosing the access time required for the application, thus optimizing both performance and cost. A non-multiplexed bus was chosen for the 30% throughput increase as opposed to a multiplexed address and data bus.

The M6800 peripheral control signals are used to interface synchronous M6800 Family peripherals to the asynchronous MC68008. Enable (E) is generated by the MC68008 with a period of ten MC68008 clock periods (six clocks low, four clocks high). Valid peripheral address ($\overline{\text{VPA}}$) is an input to the MC68008 which is generated by the decoder used to generate chip selects for the M6800 peripherals. After $\overline{\text{VPA}}$ is asserted, the MC68008 synchronizes its bus cycle to E. $\overline{\text{VMA}}$ may be generated by the circuit shown in Figure 4 and should be used to ensure that the M6800 peripheral chip select meets the required timing. Depending on when the $\overline{\text{VPA}}$ decode is valid with respect to E, the synchronous bus cycles may last from 12 to 22 clock periods.

The processor status outputs indicate the state (user or supervisor) and the type of access currently underway (program or data). The processor also uses the status outputs to indicate execution of an interrupt acknowledge cycle.

Bus arbitration control is accomplished using a two line interface formed by bus request ($\overline{\text{BR}}$) and bus grant ($\overline{\text{BG}}$). In a system with multiple bus masters (processors) or DMA devices, these signals determine which device will be the bus master. The MC68452 bus arbitration module member of the M68000 Family can be used to extend arbitration support to eight bus masters.

The two interrupt control inputs indicate the encoded priority level of the device requesting the interrupt. Level seven is the highest level interrupt and is not maskable, while level zero indicates that no interrupt is pending. The MC68008 supports three interrupt levels—2, 5 and 7— and is a direct subset of the MC68000.

The three system control lines, $\overline{\text{RESET}}$, $\overline{\text{HALT}}$, and $\overline{\text{BERR}}$, are used to reset the processor and/or the system, halt the processor, or indicate that a bus error has occurred. A bus cycle may be rerun if the original error can be corrected. The $\overline{\text{HALT}}$ line may also be driven by the processor to indicate a catastrophic system failure, in which case the processor halts to preserve system integrity.

EXCEPTION PROCESSING

Exception processing allows the MC68008 to handle interrupts, address errors, unimplemented instructions, and other commonly encountered exceptions while maintaining absolute system integrity.

The exception processing state is associated with interrupts, trap instructions, tracing, and other exceptional conditions. An exception may be generated internally by an instruction or by an unusual condition occurring during program execution. Exception processing provides an efficient context switch to enable the processor to handle unusual conditions without degrading system integrity.

SUMMARY

The MC68008 is the simplest way to insure high performance in an environment using 8-bit data paths and preserve a growth path to a full 32-bit implementation. This growth path is easily achievable due to compatibility of both software and hardware.

MOTOROLA

MC68488
(1.0 MHz)

MC68A488
(1.5 MHz)

MC68B488
(2.0 MHz)

MOS

(N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

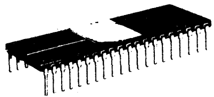
GENERAL PURPOSE INTERFACE ADAPTER

GENERAL PURPOSE INTERFACE ADAPTER

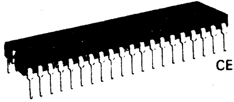
The MC68488 GPIA provides the means to interface between the IEEE-488 standard instrument bus and the M6800 MPU Family. The GPIB instrument bus provides a means of controlling and moving data between instruments connected to it.

The MC68488 will automatically handle all handshake protocol needed on the instrument bus.

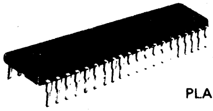
- Single- or Dual-Primary Address Recognition
- Secondary Address Capability (Talker or Listener)
- Complete Source and Acceptor Handshakes
- Programmable Interrupts
- RFD Holdoff to Prevent Data Overrun
- Operates with DMA Controller
- Serial- and Parallel-Polling Capability
- Talk-Only or Listen-Only Capability
- Selectable Automatic Features to Minimize Software
- Synchronization Trigger Output
- M6800 Bus Compatible



L SUFFIX
CERAMIC PACKAGE
CASE 715



S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711

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

Characteristics	Symbol	Value	Rating
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Cerdip		60	
Plastic		100	

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. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

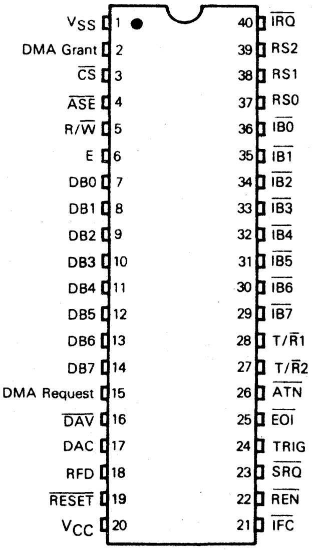
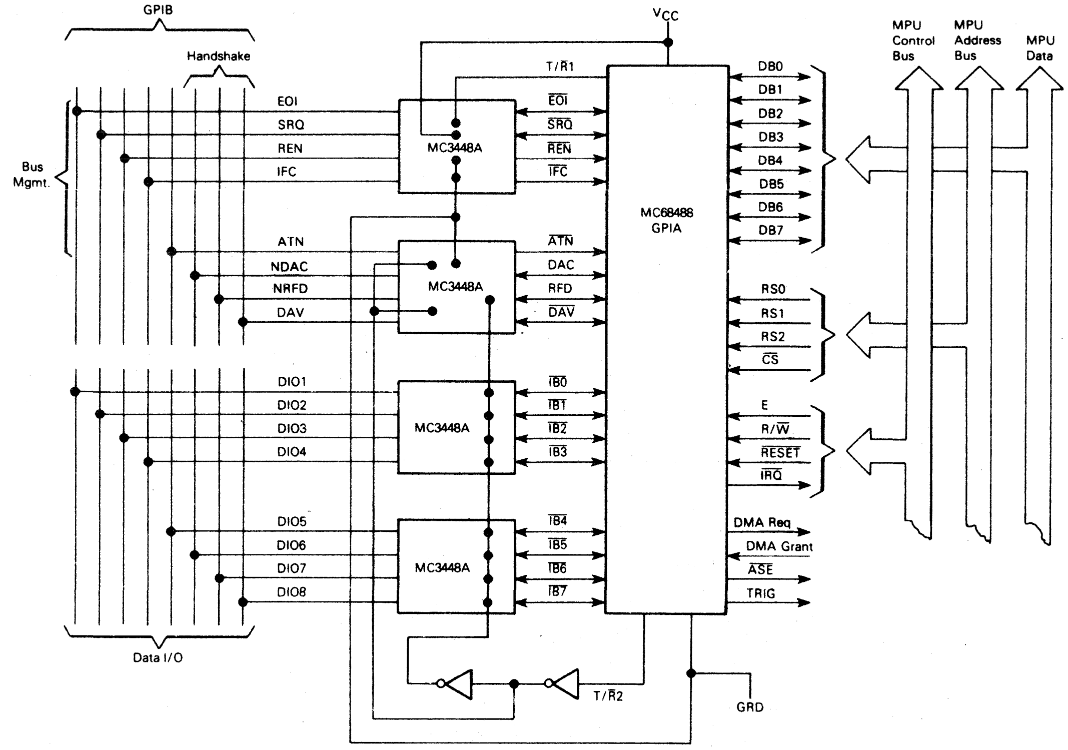


FIGURE 1 — PIN ASIGNMENT

FIGURE 2 — GPIB INTERFACE

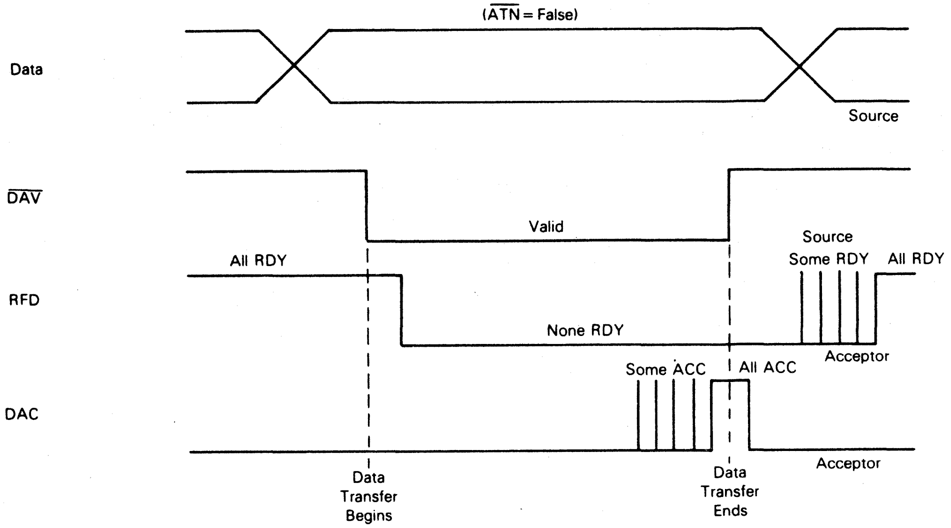


Note: The four MC3448A quad bus transceivers can be replaced by two MC3447 octal bus transceivers.

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		V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage		V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)		I_{in}	–	1.0	2.5	μA
Three State (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{TSI}	–	2.0	10	μA
DC Output High Voltage (I_{load} = −205 μA)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
DC Output Low Voltage (I_{Load} = 1.6 mA) (I_{Load} = 3.2 mA)	 D0-D7 $\overline{SRQ}$, $\overline{IRQ}$	V_{OL}	 – –	 – –	 $V_{SS}+0.4$ $V_{SS}+0.4$	V
Output Leakage Current (Off State) (V_{OH} = 2.4 V)	$\overline{SRQ}$, $\overline{IRQ}$	I_{LOH}	–	1.0	10	μA
Internal Power Dissipation		P_{INT}	–	600	750	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	 D0-D7 All Others	C_{in}	 – –	 – –	 12.5 7.5	pF

FIGURE 3 — SOURCE AND ACCEPTOR HANDSHAKE



This diagram displays logical voltage levels on the MC68488 pins. The MC68488 pins are labeled as the complement of the specified 488 bus callout; i.e., $\overline{DAV}$ rather than DAV. RFD rather than NRFD and DAC rather than NDAC. This was done to stay with standard positive logic format, which is used with all M6800 family devices.

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

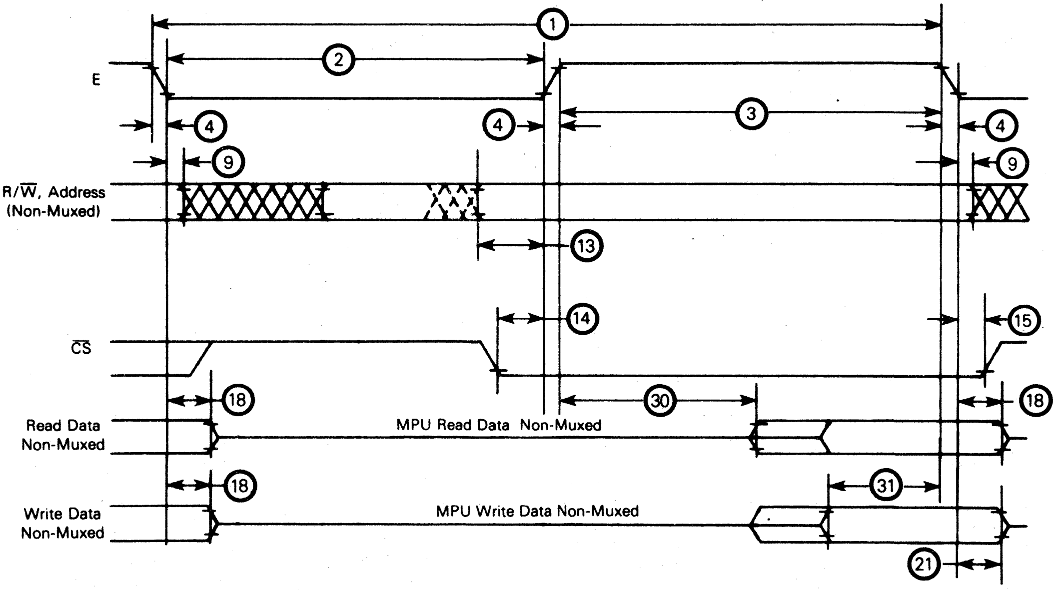
BUS TIMING (See Notes 1, 2, and 3)

Ident. Number	Characteristics	Symbol	MC68488		MC68A488*		MC68B488*		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time	t_{AH}	10	–	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	60	–	40	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	60	–	40	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	20	50**	20	50**	20	50**	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	180	–	150	ns
31	Input Data Setup Time	t_{DSW}	165	–	80	–	60	–	ns

*See Table 1 for GPIB transceiver considerations when using MC68A488 or MC68B488.

**The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high-impedance).

FIGURE 4 — BUS TIMING



NOTES:

1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \leq 0.8$ V, $V_H \geq 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

FIGURE 5 — OUTPUT BUS TIMING

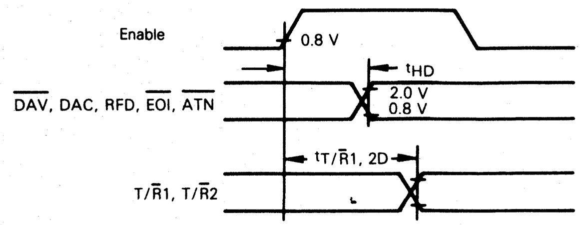


TABLE 1 — AC TIME VALUES

Characteristics		Symbol*	Typ	Unit
Settling Time for Multiple Message	SH	T_1	≥ 2	μs**
Response to $\overline{\text{ATN}}$	SH, AH, T, L	t_3	≤ 200	ns
Interface Message Accept Time‡	AH	T_3	>0	§
Response to $\overline{\text{IFC}}$ or $\overline{\text{REN}}$ False	T, TE, L, LE	t_4	<100	μs
Response to $\overline{\text{ATN}}$•$\overline{\text{EOI}}$	PP	t_5	≤ 200	ns

* Time values specified by a lower case t indicate the maximum time allowed to make a state transition. Time values specified by an upper case T indicate the minimum time that a funcion must remain in a state before exiting.

** If three-state drivers are used on the $\overline{\text{DIO}}$-$\overline{\text{DAV}}$ and $\overline{\text{EOI}}$ lines, T_1 may be:
(1) ≥ 1100 ns
(2) Or ≥ 700 ns if it is known that within the controller $\overline{\text{ATN}}$ is driven by a three-state driver.
(3) Or ≥ 500 ns for all subsequent bytes following the first sent after each false transition of $\overline{\text{ATN}}$ [the first byte must be sent in accordance with (1) or (2)].

‡ Time required for interface functions to accept, not necessarily respond to interface messages.

§ Implementation dependent.

When using an E clock of 1.5 MHz on the MC68A488, the GPIB data lines, $\overline{\text{DAV}}$, and $\overline{\text{EOI}}$ lines must have three-state drivers — See Note **.
When using an E clock of 2.0 MHz on the MC68B488 the GPIB data lines, $\overline{\text{DAV}}$, $\overline{\text{EOI}}$, and $\overline{\text{ATN}}$ lines must have three-state drivers — See Note **

GENERAL DESCRIPTION

The IEEE-488 instrument bus standard is a bit-parallel, byte-serial bus structure designed for communication to and from intelligent instruments. Using this standard, many instruments may be interconnected, remotely and automatically controlled, or programmed. Data may be taken from, sent to, or transferred between instruments. A bus controller dictates the role of each device by making the attention line true and sending talk or listen addresses on the instrument bus data lines; those devices which have matching addresses are activated. Device addresses are set into each GPIA from switches or jumpers on a PC board by a microprocessor as a part of the initialization sequence.

When the controller makes the attention line true, instrument bus commands may also be sent to single or multiple GPIAs.

Information is transmitted on the instrument bus data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as the devices can respond, but no faster than the slowest device presently addressed as active. This permits several devices of different speeds to receive the same data concurrently.

The GPIA is designed to work with standard 488-bus driver ICs (MC3447As or MC3448As) to meet the complete electrical specifications of the IEEE-488 bus. Additionally, a powered-off instrument may be powered-on without disturbing the 488 bus. With some additional logic, the GPIA could be used with other microprocessors.

The MC68488 GPIA has been designed to interface between the M6800 family microprocessor and the complex protocol of the IEEE-488 instrument bus. Many instrument bus protocol functions are handled automatically by the GPIA and require no additional MPU action. Other functions require minimum MPU response due to a large number of internal registers conveying information on the state of the GPIA and the instrument bus.

FIGURE 6 — GPIA BLOCK DIAGRAM

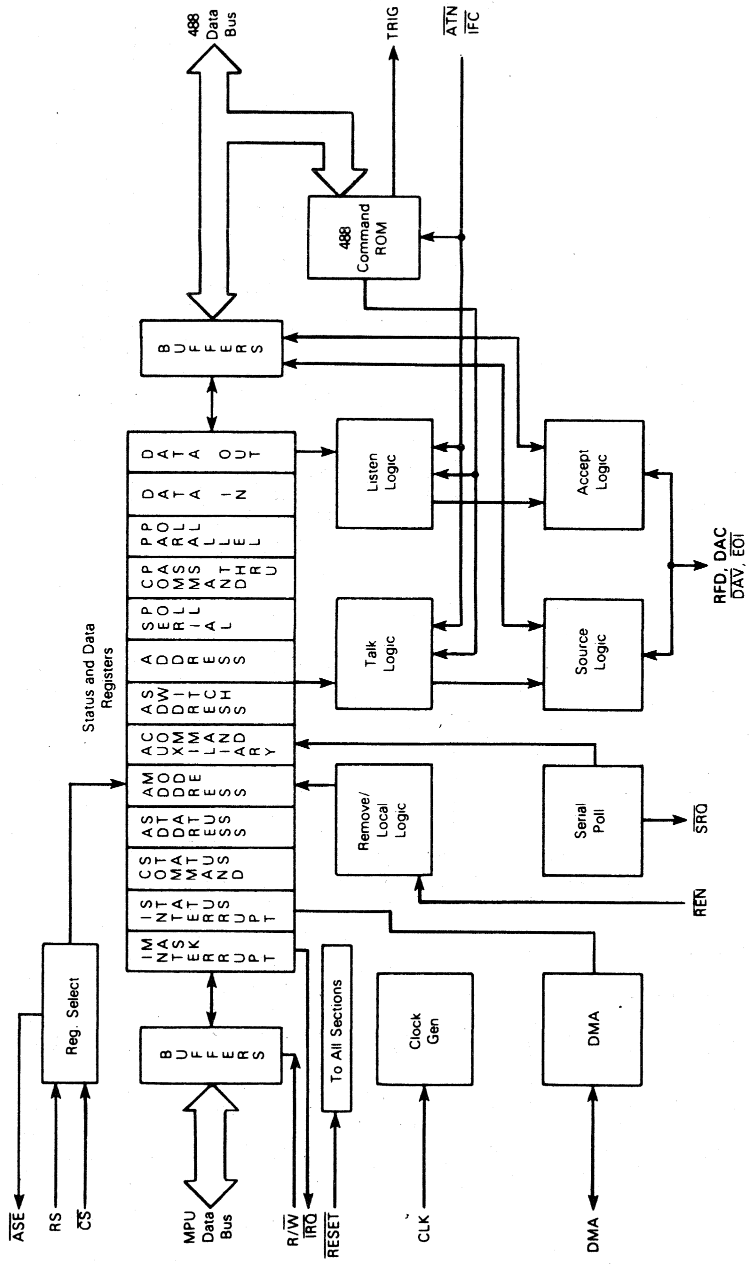
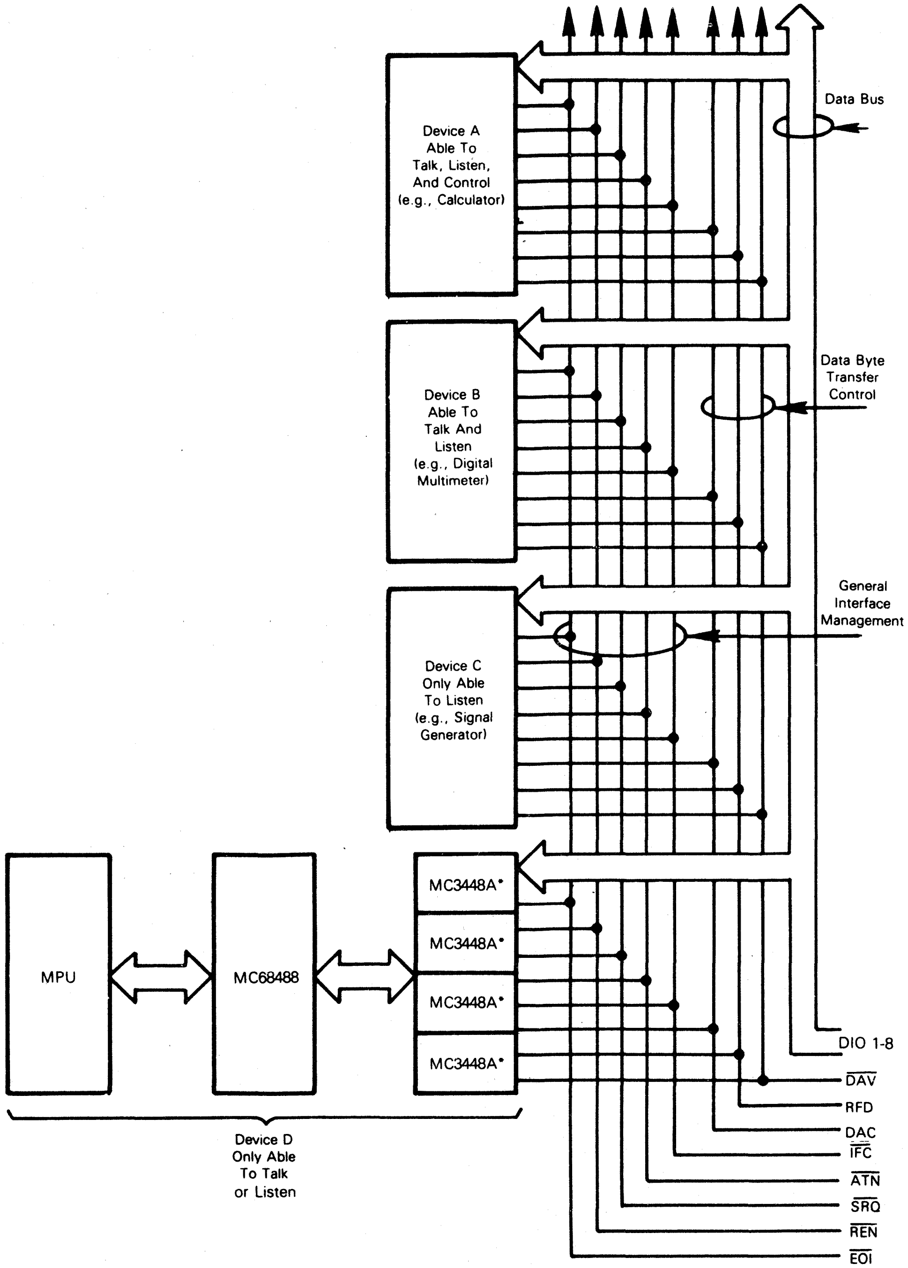


FIGURE 7 — GPIB SYSTEM



*Two MC3447A octal transceivers can be used in place of four MC3448As.

PIN DESCRIPTION

All inputs to the GPIA are high impedance and TTL compatible. All outputs from the GPIA are compatible with standard TTL. $\overline{\mathrm{IRQ}}$ (Interrupt Request) and $\overline{\mathrm{SRQ}}$, however, are open-drain outputs (no internal pullup).

INTERFACE WITH MPU

BIDIRECTIONAL DATA (D0-D7) – The bidirectional data lines allow the transfer of data between the MPU and GPIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs a GPIA read operation or the DMA controller performs a memory write operation. The Read/Write line is high when the GPIA is selected for a read operation.

CHIP SELECT ($\overline{\mathrm{CS}}$) – This input signal is used to select the GPIA. $\overline{\mathrm{CS}}$ must be low for selection of the device. Chip Select decoding is normally accomplished with external logic.

READ/WRITE INPUT (R/$\overline{\mathrm{W}}$) – This signal is generated by the MPU or DMA controller to control register access and direction of data transfer on the data bus. A low state on the GPIA Read/Write and DMA Grant lines allows for the selection of one of seven write-only reigsters when used in conjunction with register select lines RS0, RS1, and RS2. A high state on the GPIA Read/Write and low state on the DMA Grant line allows for the selection of one of eight read-only registers when used in conjunction with register select lines RS0, RS1, and RS2.

REGISTER SELECT (RS0, RS1, RS2) – The three register select inputs are used to select the various registers inside the GPIA. These three lines are used in conjunction with the Read/Write line to select a particular register that is to be written or read. Table 2 shows the register select coding.

INTERRUPT REQUEST ($\overline{\mathrm{IRQ}}$) – The $\overline{\mathrm{IRQ}}$ output goes to the common interrupt bus line for the MPU. This is an open-drain output which is wire-ORed to the $\overline{\mathrm{IRQ}}$ bus line. The $\overline{\mathrm{IRQ}}$ is asserted low when an enable interrupt occurs and stays low until the MPU reads the interrupt status register. Reading R0R will reset $\overline{\mathrm{IRQ}}$ to the high state.

TABLE 2 – REGISTER ACCESS

RS2	RS1	RS0	R/$\overline{\mathrm{W}}$	Register Title	Register Symbol
0	0	0	1	Interrupt Status	R0R
0	0	0	0	Interrupt Mask	R0W
0	0	1	1	Command Status	R1R
0	0	1	0	Unused	–
0	1	0	1	Address Status	R2R
0	1	0	0	Address Mode	R2W
0	1	1	1	Auxiliary Command	R3R
0	1	1	0	Auxiliary Command	R3W
1	0	0	1	Address Switch*	R4R
1	0	0	0	Address	R4W
1	0	1	1	Serial Poll	R5R
1	0	1	0	Serial Poll	R5W
1	1	0	1	Command Pass-Through	R6R
1	1	0	0	Parallel Poll	R6W
1	1	1	1	Data In	R7R
1	1	1	0	Data Out	R7W

*External to MC68488

$\overline{\mathrm{RESET}}$ – The $\overline{\mathrm{RESET}}$ input provides a means of resetting the GPIA from a hardware source. In the low state, the $\overline{\mathrm{RESET}}$ input causes the following:

- The Interrupt "Mask" register is reset;
- All status conditions are reset;
- The GPIA is placed in the Untalk/Unlisten state;
- The Parallel Poll, Serial Poll, Data In, and Data Out registers are reset;
- The Address register and Address mode register are cleared;
- All stored conditions in the Auxiliary Comand register except bit 7 are reset – (bit 7 is set);
- T/R1, 2 will go to the low state.

When $\overline{\mathrm{RESET}}$ returns high (the inactive state) the GPIA will remain in the reset condition until the MPU writes bit 7 of the Auxiliary Command register (R3W) low. Prior to the release of the software reset bit, the only register that can be accessed is the Address register. The conditions affected by the $\overline{\mathrm{RESET}}$ pin cannot be changed while this pin is low.

E (ENABLE CLOCK) – E activates the address inputs (CS, RS0, RS1, and RS2) and R/$\overline{\mathrm{W}}$ input and enables data transfer on the MPU data bus. It is also used internally as a state counter allowing the device to change interface states. The E input should be connected to a free-running clock source such as the MC6800 $\phi 2$ or the Enable Signal of other M6800 family MPUs.

GPIA/GPIB INTERFACE BUS SIGNALS

The GPIA provides a set of eighteen interface signal lines between the M6800 and the IEEE-488 Standard bus.

NOTE

The IEEE-488 Standard defines these signals as negative logic. In this document all MPU and MC68488 signals are defined as positive logic.

SIGNAL LINES ($\overline{\mathrm{IB0}}$-$\overline{\mathrm{IB7}}$) – These bidirectional lines allow for the flow of 7-bit ASCII interface messages and device dependent messages. Data appears on these lines in a bit-parallel byte-serial form. These lines are buffered by transceivers and applied to the IEEE-488 Standard bus (DIO1-DIO8).

BYTE TRANSFER LINES (DAC, RFD, $\overline{\mathrm{DAV}}$) – These lines allow for proper transfer of each data byte on the bus between sources and acceptors. RFD goes passively high indicating "Ready For Data". A source will indicate the "data is valid" by pulling $\overline{\mathrm{DAV}}$ low. Upon the reception of valid data, DAC will go passively high indicating that the "data has been accepted" by all acceptors. The handshake lines have internal pullup resistors.

BUS MANAGEMENT LINES ($\overline{\mathrm{ATN}}$, $\overline{\mathrm{IFC}}$, $\overline{\mathrm{SRQ}}$, $\overline{\mathrm{EOI}}$, $\overline{\mathrm{REN}}$) – These lines are used to manage an orderly flow of information across the interface lines.

ATTENTION ($\overline{\mathrm{ATN}}$) – is continuously monitored by the GPIA. The device responds to any changes on this line in less than 200 ns by activating the transmit/receive control signals. If the $\overline{\mathrm{EOI}}$ line and $\overline{\mathrm{ATN}}$ are low at the same time, GPIA will place the contents of a parallel poll register on the IEEE-488 Standard bus.

INTERFACE CLEAR ($\overline{\text{IFC}}$) — is used by a system controller to put the GPIA in a known quiescent state. The occurrence of $\overline{\text{IFC}}$ will place the GPIA in the Listener/Talker idle state (LIDS or TIDS). If the MC68488 is in a Listener Active state with a byte of data in the Data-In register (BI bit set) an $\overline{\text{IFC}}$ will place the part in LIDS but will not destroy the received byte nor the status indication (BI). Any interface function that requires the device to be in either the Listener or Talker Active state (e.g., a Serial Poll enable command) will be reset if an $\overline{\text{IFC}}$ occurs. A command that originates from the MPU (e.g., to, lo, fget, hlda) will only be affected during the occurrence of an $\overline{\text{IFC}}$ (when $\overline{\text{IFC}}$ is low) and will return to its programmed state when $\overline{\text{IFC}}$ returns high; i.e., $\overline{\text{IFC}}$ will not affect local messages. For example: if the GPIA is in TACS (Talker Active State) and has placed a byte in the Data-Out register it has made a new byte available (nba). If $\overline{\text{IFC}}$ occurs while the source handshake is in SDYS, the talker function will be returned to its idle state but nba (a local message) will not be destroyed. When the GPIA is again made a talker, the byte in the Data-Out register (placed there before $\overline{\text{IFC}}$) will be placed onto the GPIB. The address register is not affected by an $\overline{\text{IFC}}$.

SERVICE REQUEST ($\overline{\text{SRQ}}$) — is used to indicate a need for attention in addition to requesting an interruption in the current sequence of events. This indicates to the controller that a device on the bus is in need of service. This output becomes active low by setting the rsv bit (bit 6) of R5W. This line is an open drain and an external pullup resistor (nominal 3.3k ohm) must be used.

REMOTE ENABLE (REN) — is used to select one of two alternate sources of device programming data — local and remote control. When this input is low the GPIA is enabled to move to the REMS state. Note that REN being low is a necessary but not a sufficient condition for moving to REMS.

END OF IDENTIFY ($\overline{\text{EOI}}$) — Serves a dual purpose. When the GPIA is in TACS and the MPU writes bit 5 or R3W (feoi) this pin becomes an output and signals the end of a multibyte transfer. If the system controller makes the $\overline{\text{EOI}}$ line true in conjunction with ATN, the contents of the Parallel Poll register will be placed on the IEEE-488 Standard bus.

TRANSMIT/RECEIVE CONTROL SIGNALS (T/$\overline{\text{R}}$1, T/$\overline{\text{R}}$2) — These two signals are used to control the quad or octal transceivers which drive the interface bus. It is assumed that transceivers equivalent to the MC3447 or MC3448A will be used where each transceiver has a separate Transmit/Receive control pin. These pins can support one TTL load each. The outputs can then be grouped and the control for $\overline{\text{SRQ}}$ hardwired high to transmit. The Transmit/Receive inputs of $\overline{\text{REN}}$, $\overline{\text{IFC}}$, and $\overline{\text{ATN}}$ are hardwired low to receive. $\overline{\text{EOI}}$ is controlled by T/$\overline{\text{R}}$1 through the MC3447/MC3448A (or equivalents) allowing it to transmit or receive. T/$\overline{\text{R}}$1 operates exactly as T/$\overline{\text{R}}$2 except during the parallel polling sequence. During parallel poll, $\overline{\text{EOI}}$ will be made an input by T/$\overline{\text{R}}$1 while $\overline{\text{DAV}}$ and IB0/IB7 lines are outputs.

SPECIAL CONTROL SIGNALS

DMA CONTROL LINES (DMA GRANT, DMA REQUEST — The DMA request line is used to signal a DMA controller that a data transfer is pending. The DMA request line is set high if either the BI or BO status bits are set in the Interrupt Status Register (R0R). The DMA request line is cleared when the DMA Grant is true. The DMA Grant line is used to signal the GPIA that the DMA has control of the MPU data and address lines. The DMA Grant, when set high, selects register 7. It also inhibits the RS0, RS1, and RS2 lines. During this time the $\overline{\text{CS}}$ input must be high. The DMA Grant also inverts the function of the R/$\overline{\text{W}}$ line making it $\overline{\text{R}}$/W. Thus, if the DMAC supplies a write function to a memory location, this same line will perform a read of the GPIA (R7R) and vice versa.

NOTE

DMA GRANT MUST BE GROUNDED WHEN NOT IN USE.

TRIGGER OUTPUT (TRIG) — The TRIG pin provides an output corresponding to the GET and fget commands. A hardware or software reset places this output at a low level. The trigger output can be programmed high by either of two methods:

1. Setting fget (bit 0 of R3W) by the MPU causes the trigger output to be set. It remains set until the fget bit is programmed low or until a reset occurs.
2. The Trigger Output is set upon reception of a GET command from the controller. It is reset when the GPIA moves out of DTAS (Device Trigger Active State); i.e., when GET, LADS, or ACDS occur.

ADDRESS SWITCH ENABLE ($\overline{\text{ASE}}$) — The $\overline{\text{ASE}}$ output is used to enable three-state buffers that connect instrument address switches to the MPU data bus. This output pin is pulsed low when the Address Switch Register of the GPIA is read (R4R), i.e., a read of R4R will drive the $\overline{\text{ASE}}$ line low for the E clock that is used to read R4R.

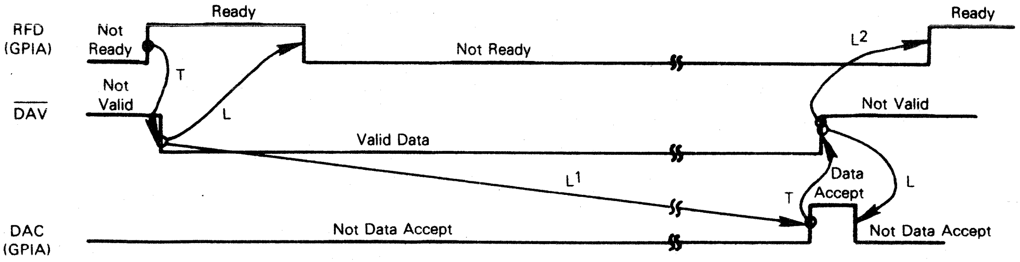
GPIB HANDSHAKE SEQUENCE

The GPIB handshake line transitions are debounced inside the GPIA with the E-clock to provide a high degree of noise immunity. Due to the asynchronous nature between the GPIB handshake line transitions and the internal debounce circuit sampling, the time required for handshake completion can vary by 1 E clock cycle.

LISTENER MODE — The handshake sequence begins when the GPIA makes RFD true (Figure 8). A second byte cannot be transferred on the GPIB until the GPIA again makes RFD true for the next handshake. The total time required by the GPIA to debounce all of the handshake lines in the appropriate time sequence is 7-8 E clock cycles. The 1 cycle variation is due to the asynchronous nature of the GPIB with respect to the GPIA debounce circuitry. To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer.

TALKER MODE — The handshake sequence begins when the listener(s) on the GPIB make the RFD line true (Figure 9). When this occurs and the MPU has written a byte to R7W the GPIA will make the $\overline{\text{DAV}}$ line low indicating to the listeners that valid data is on the GPIB. When this byte is accepted and RFD is again made true the next transfer can begin. The GPIA debounce circuitry requires 6-7 E clock cycles to complete a handshake sequence. As with the listener there is a 1 E clock fluctuation due to the asynchronous nature between the GPIB handshake and the GPIA debounce circuitry. To determine the maximum throughput rate add this number to the number of instructions or DMA cycles used to service each transfer.

FIGURE 8 — GPIA IN LISTEN MODE*



The GPIA in the listener mode controls the DAC and RFD lines. The $\overline{\text{DAV}}$ line is controlled by the Talker on the GPIB. Note that the RFD and DAC lines are wire ANDed on the GPIB; thus, these lines returning to the high state are dependent on all devices programmed as listeners releasing the lines.

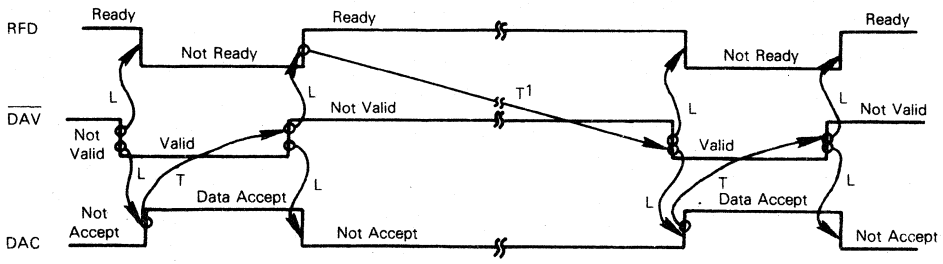
L The listener(s) on the GPIB causes this action.

T The talker on the GPIB causes this action.

1 The release of DAC may require action by the MPU (reading R7R for data byte transfers or writing dacr (R3W) high for certain commands). For some commands DAC is automatically released by the GPIA. Consult the MC68488 user's manual for details.

2 The RFD line is normally automatically released by the GPIA. Certain conditions, however, require MPU intervention to provide this release. Consult the MC68488 user's manual for details.

FIGURE 9 — GPIA IN TALKER MODE*



* The GPIA in the talker mode controls the $\overline{\text{DAV}}$ line. The RFD and DAC lines are controlled by the listener(s) on the GPIB.

L The listener(s) on the GPIB causes this action.

T The talker on the GPIB causes this action.

1 Two conditions must occur before the $\overline{\text{DAV}}$ line goes to the valid state. The RFD line must be high and a data byte must be placed in the data out register (nba must be true).

GPIA INTERNAL CONTROLS AND REGISTERS*

There are fifteen locations accessible to the MPU data bus which are used for transferring data to control the various functions on the chip and provide current chip status. Seven of these registers are write only and eight registers are read only. The various registers are accessed according to the three least-significant bits of the MPU address bus and the status of the Read/Write line. One of the fifteen registers is external to the IC but an address switch register is provided for reading the address switches. Table 2 shows actual bit contents of each of the registers.

DATA-IN REGISTER R7R – The data-in register is an actual 8-bit storage register used to move data from the interface bus when the chip is a listener. Reading the register does not destroy information in the data-out register. DAC (data accepted) will remain low until the MPU removes the bytes from the data-in register. The chip will automatically finish the handshake by allowing DAC to go high. In RFD (ready for data) holdoff mode, a new handshake is not initiated until a command is sent allowing the chip to release holdoff. This will delay a talker until the available information has been processed.

Data-In Register
(Read Only)

D17	D16	D15	D14	D13	D12	D11	D10

D10-D17 – Correspond to DIO1-DIO8 of the 488-1975 Standard and $\overline{\text{IB0}}$-$\overline{\text{IB7}}$ of the MC68488.

*NOTE: Upper and lower case type designations will be used with the register bits to indicate remote or local messages respectively.

DATA-OUT REGISTER R7W – The data-out register is an actual 8-bit storage register used to move data out of the chip onto the interface bus. Reading from the data-in register has no effect on the information in the data-out register. Writing to the data-out register has no effect on the information in the data-in register.

Data-Out Register
(Write Only)

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

DO0-DO7 – Correspond to DIO1-DIO8 of the 488-1978 Standard and $\overline{\text{IB0}}$-$\overline{\text{IB7}}$ of the MC68488.

INTERRUPT MASK REGISTER R0W – The Interrupt Mask Register is a 7-bit storage register used to select the particular events that will cause an interrupt to be sent to the MPU. The seven control bits may be set independently of each other. If dsel (bit 7 of the Address Mode Register) is set high CMD bit 2 will interrupt on SPAS or RLC. If dsel is set low CMD will interrupt on UACG, UUCG, and DCAS in addition to RLC and SPAS. The Command Status Register R1R may then be used to determine which command caused the interrupt. Setting GET bit 5 allows an interrupt to occur on Group Execute Trigger Command. END bit 1 allows an interrupt to occur if $\overline{\text{EOI}}$ is true (low) and $\overline{\text{ATN}}$ is false (high). APT bit 3 allows an interrupt to occur indicating that a secondary address is available to be examined by the MPU if apte (bit 0 of Address Mode Register) is enabled and listener or talker primary address is received and a Secondary Command Group is received. A typical response for a valid secondary address would be to set msa (bit 3 of Auxiliary Command Register) true and dacr (bit 4 Auxiliary Command Register) true, releasing the DAC handshake. BI indicates that a data

TABLE 3 – REGISTER CONTENTS

	7	6	5	4	3	2	1	0	
R0W	IRQ	BO	GET		APT	CMD	END	BI	Interrupt "Mask Register"
R0R	INT	BO	GET		APT	CMD	END	BI	Interrupt Status Register
R1R	UACG	REM	LOK		RLC	SPAS	DCAS	UUCG	Command Status Register
R1W									Unused
R2R	ma	to	lo	ATN	TACS	LACS	LPAS	TPAS	Address Status Register
R2W	dsel	to	lo		hlde	hlda		apte	Address Mode Register
R3R	RESET	DAC	DAV	RFD	msa	rtl	ulpa	fget	Auxiliary Command Register
R3W		rfdr	feoi	dacr			dacd		Auxiliary Command Register
R4R	UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1	Address Switch Register
R4W	lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1	Address Register
R5R	S7	SRQS	S5	S4	S3	S2	S1	S0	Serial Poll Register
R5W		rsv							
R6R	B7	B6	B5	B4	B3	B2	B1	B0	Command Pass-Through Register
R6W	PPR8	PPR7	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	Parallel Poll Register
R7R	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Data In Register
R7W	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	Data Out Register

Notes:
1. Upper case letters indicate a message resulting from the IEEE-488 Standard bus.
2. Lower case letters indicate a message resulting from the MPU data bus.
3. The bit terminology of the Data In and Data registers represent the numbering of the IEEE-488 Standard bus and not the 6800 MPU bus – see Section 3.1.2.

byte is waiting in the data-in register. BI is set high when data-in register is full. BO indicates that a byte from the data-out register has been accepted. BO is set when the data-out register is empty. IRQ enabled high allows any interrupt to be passed to the MPU.

Interrupt Mask Register
(Write Only)

IRQ	BO	GET	X	APT	CMD	END	BI

IRQ – Mask bit for IRQ pin
BO – Interrupt on byte output
GET – Interrupt on Group Execute Trigger
APT – Interrupt on Secondary Address Pass-Through
CMD – Interrupt on SPAS + RLC + $\overline{\text{dsel}}$ (DCAS + UUCG + UACG)
END – Interrupt on EOI and $\overline{\text{ATN}}$
BI – Interrupt on byte input

THE INTERRUPT STATUS REGISTER R0R – The Interrupt Status Register is a 7-bit storage register which corresponds to the interrupt mask register with an additional bit INT bit 7. Except for the INT bit the other bits in the status register are set regardless of the state of the interrupt mask register when the corresponding event occurs. The $\overline{\text{IRQ}}$ (MPU interrupt) is cleared when the MPU reads from the register. INT bit 7 is the logical OR of the other six bits AND-ed with the respective bit of R0W.

Interrupt Status Register
(Read Only)

INT	BO	GET	X	APT	CMD	END	BI

INT – Logical OR of all other bits in this register ANDed with the respective bits in the interrupt mask register.
BO – A byte of data has been output
GET – A Group Execute Trigger has occurred
APT – An Address Pass-Through has occurred
CMD – SPAS + RLC + $\overline{\text{dsel}}$ (DCAS + UUCG + UACG) has occurred
END – An EOI has occurred with $\overline{\text{ATN}} = 0$
BI – A byte has been received

SERIAL POLL REGISTER R4R/W – The Serial Poll Register is an 8-bit storage register which can be both written into and read by the MPU. It is used for establishing the status byte that the chip sends out when it is serial poll enabled. Status may be placed in bits 0 through 5 and bit 7. Bit 6 rsv (request for service) is used to drive the logic which controls the $\overline{\text{SRQ}}$ line on the bus telling the controller that service is needed. This same logic generated the signal SRQS which is substituted in bit 6 position when the status byte is read by the MPU $\overline{\text{IB0-IB7}}$. In order to initiate a rsv (request for service), the MPU sets bit 6 true (generating $\overline{\text{rsv}}$ signal) and this in turn causes the chip to pull down the $\overline{\text{SRQ}}$ line. SRQS is the same as rsv when SPAS is false. Bit 6 as read by the MPU will be the SRQS (Service Request State).

Serial Poll Register
(Read)

S8	SRQS	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits.
SRQS – Bus in Service Request State

Serial Poll Register
(Write)

S8	rsv	S6	S5	S4	S3	S2	S1

S1-S8 – Status bits
rsv – generate a service request

PARALLEL POLL REGISTER R6W – This register will be loaded by the MPU and the bits in this register will be delivered to the instrument bus IB0-IB7 during PPAS (Parallel Poll Active State). This register powers up in the PPO (Parallel Poll No Capability) state. The reset bit (Auxiliary Command Register bit 7) will clear this register to the PPO state.

The parallel poll interface function is executed by this chip using the PP2 subset (Omit Controller Configuration Capability). The controller cannot directly configure the parallel poll output of this chip. This must be done by the MPU. The controller will be able to indirectly configure the parallel poll by issuing an addressed command which has been defined in the MPU software.

Parallel Poll Register
(Write Only)

PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1

Bits delivered to bus during Parallel-Poll Active State (PPAS)
Register powers up in the PPO state
Parallel Poll is executed using the PP2 subset

ADDRESS MODE REGISTER R2W – The address mode register is a storage register with six bits for control: to, lo, hlde, hlda, dsel, and apte. The to bit 6 selects the talker/listener and addresses the chip to talk only. The lo bit 5 selects the talker/listener and sets the chip to listen only. The apte bit 0 is used to enable the extended addressing mode. If apte is set low the device goes from the TPAS (Talker Primary Address State) directly to the TADS (Talker Addressed State). The hlda bit 2 holds off RFD (Ready for Data) on ALL DATA until rfdr is set true. The hlde bit 3 holds off RFD on EOI enabled (low) and ATN not enabled (high). This allows the last byte in a block of data to be continually read as needed. Writing rfdr true (high) will allow the next handshake to proceed.

Address Mode Register
(Write Only)

dsel	to	lo	X	hdle	hdla	X	apte

dsel – configure for automatic completion of handshake sequence on occurrence of GET, UACG, UUCG, SDC, or DCL commands
to – set to talk-only mode
lo – set to listen-only mode
hdle – Hold-off RFD on end (END = EOI$\wedge\overline{\text{ATN}}$)
hdla – Hold-off RFD on all data
apte – Enable the address pass-through feature

ADDRESS STATUS REGISTER R2R — The address status register is not a storage register but simply an 8-bit port used to couple internal signal modes to the MPU bus. The status flags represented here are stored internally in the logic of the chip. These status bits indicate the addressed state of the talker/listener as well as flags that specify whether the chip is in the talk only or listen only mode. The ATN, bit 4, contains the condition of the Attention Line. The ma signal is true when the chip is in:

TACS — Talker Active State
TADS — Talker Addressed State
LACS — Listener Active State
LADS — Listener Addressed State
SPAS — Serial Poll Active State

Address Status Register (Read Only)

ma	to	lo	ATN	TACS	LACS	LPAS	TPAS

ma — my address has occurred
to — the talk-only mode is enabled
lo — the listen-only mode is enabled
ATN — the Attention command is asserted
TACS — GPIA is in the Talker Active State
LACS — GPIA is in the Listener Active State
LPAS — GPIA is in the Listener Primary Addressed State
TPAS — GPIA is in the Talker Primary Addressed State

ADDRESS SWITCH REGISTER R4R — The address switch register is external to the chip. There is an enable line ($\overline{\text{ASE}}$) to be used to enable three-state drivers connected between the address switches and the MPU. When the MPU addresses the address switch register the $\overline{\text{ASE}}$ line directs the switch information to be sent to the MPU. The five least-significant bits of the 8-bit register are used to specify the bus address of the device and the remaining three bits may be used at the discretion of the user. The most probable use of one or two of the bits is for controlling the listener only or talk only functions.

Address Switch Register (Read Only)

UD3	UD2	UD1	AD5	AD4	AD3	AD2	AD1

AD1-AD5 — Device address
UD1-UD3 — User definable bits
When this "register" is addressed, the $\overline{\text{ASE}}$ pin is set which allows external address switch information from the bus device to be read.

ADDRESS REGISTER R4W — The Address Register is an 8-bit storage register. The purpose of this register is to carry the primary address of the device. The primary address is placed in the five least-significant bits of the register. If external switches are used for device addressing these are normally read from the Address Switch Register and then placed in the Address Register by the MPU.

AD1 through AD5 bits 0-5 are for the device's address. The lsbe bit 7 is set to enable the Dual Primary Addressing Mode. During this mode the device will respond to two consecutive addresses, one address with AD1 equal to 0 and the other address with AD1 equal to 1. For example, if the device's address is HEX 0F, the Dual Primary Addressing Mode would allow the device to be addressed at both HEX 0F and HEX 0E. The dal bit 6 is set to disable the listener and the dat bit 5 is set to disable the talker.

This register is cleared by the $\overline{\text{RESET}}$ input only (not by the reset bit of the Auxiliary Command Register bit 7).

When $\overline{\text{ATN}}$ is enabled and the primary address is received on the $\overline{\text{IB0-7}}$ lines, the MC68488 will set bit 7 of the address status register (ma). This places the MC68488 in the TPAS or LPAS.

When $\overline{\text{ATN}}$ is disabled the GPIA may go to one of three states: TACS, LACS, or SPAS.

Address Register (Write Only)

lsbe	dal	dat	AD5	AD4	AD3	AD2	AD1

lsbe — enable dual primary addressing mode
dal — disable the listener
dat — disable the talker
AD1-AD5 — Primary device address, usually read from address switch register
Register is cleared by the $\overline{\text{RESET}}$ input pin only.

AUXILIARY COMMAND REGISTER R3R/W — Bit 7, reset, initializes the chip to the following states: (reset is set true by external $\overline{\text{RESET}}$ input pin and by writing into the register from the MPU).

SIDS — Source Idle State
AIDS — Acceptor Idle State
TIDS — Talker Idle State
LIDS — Listener Idle State
LOCS — Local State
NPRS — Negative Poll Response State
PPIS — Parallel Poll Idle State
PUCS — Parallel Poll Unaddressed to Configure State
PPO — Parallel Poll No capability

rfdr (release RFD handshake) bit 6 allows for completion of the handshake that was stopped by RFD (Ready For Data) holdoff commands hlda and hlde.

fget (force group execute trigger) bit 0 has the same effect as the GET (Group Execute Trigger) command from the controller.

rtl (return to local) bit 2 allows the device to respond to local controls and the associated device functions are operative.

dacr (release DAC handshake) bit 4 is set high to allow DAC to go passively true. This bit is set to indicate that the MPU has examined a secondary address or an undefined command.

upla (upper/lower primary address) bit 1 will indicate the state of the LSB of the address received on the DIO1-8 bus lines at the time the last Primary Address was received. This bit can be read but not written by the MPU.

msa (valid secondary address) bit 3 is set true (high) when TPAS (Talker Primary Addressed State) or LPAS (Listener Primary Addressed State) is true. The chip will become addressed to listen or talk. The primary address must have been previously received.

RFD, $\overline{\text{DAV}}$, DAC — (Ready For Data, Data Valid, Data Accepted) bits assume the same state as the corresponding signal on the MC68488 package pins. The MPU may only read this bit. These signals are not synchronized with the MPU clock.

dacd (data accept disable) bit 1 set high by the MPU will prevent completion of the automatic handshake on Addresses or Commands. dacr is used to complete the handshake.

feoi (forced end or identify) bit 5 tells the chip to send $\overline{\text{EOI}}$ low. The $\overline{\text{EOI}}$ line is then returned high after the next byte is transmitted. NOTE: The following signals are not stored but revert to a false (low) level one clock cycle (MPUϕ2) after they are set true (high):

1. rfdr
2. feoi
3. dacr

These signals can be written but not read by the MPU.

Auxiliary Command Register

reset	rfdr	feoi	dacr	msa	rtl	dacd	fget	Write
	DAC	DAV	RFE			ulpa		Read

reset — initialize the chip to the following status:
(1) all interrupts cleared
(2) following bus states are in effect: SIDS, AIDS, TIDS, LIDS, LOCS, PPIS, PUCS, and PPO
(3) bit is set by $\overline{\text{RESET}}$ input pin
msa — if GPIA is in LPAS or TDAS, setting msa will force GPIA to LADS or TADS
rtl — return to local if local lcokout is disabled
ulpa — state of LSB of bus at last-primary-address receive time
fget — force group execute trigger command from the MPU has occurred
rfdr — continue handshake stopped by RFD holdoff
feoi — set EOI true, clears after next byte transmitted
dacr — MPU has examined an undefined command or secondary address
dacd — prevents completion of automatic handshake on Addresses or Commands

COMMAND STATUS REGISTER R1R — The command status register flags command or state as they occur. These flags or states are simply coupled on the MPU bus. There are five major address commands. REM shows the remote/local state of the talker/listener. REM bit 6, set low, implies the local state. LOK bit 5 shows the local lockout status of the talker/listener. RLC bit 3 is set when a change of state of the remote/local flip-flop occurs and reset when the command status register is read. DCAS bit 1 indicates that either the device clear or selected device clear has been received activating the device clear function. SPAS bit 2 indicates that the SPE command has been received activating the device serial poll function. UACG bit 7 indicates that an undefined address command has been received and depending on programming the MPU decides whether to execute or ignore it. UUCG bit 0 indicates that an undefined universal command has been received.

Command Status Register
(Read)

UACG	REM	LOK	X	RLC	SPAS	DCAS	UUCG

UACG — Undefined Addressed Command
REM — Remote Enabled
LOK — Local Lockout Enabled
RLC — Remote/Local State Changed
SPAS — Serial Poll Active State is in effect
DCAS — Device Clear Active State is in effect
UUCG — Undefined Universal Command

COMMAND PASS-THROUGH REGISTER R6R — The command pass through is an 8-bit port with no storage. When this port is addressed by MPU it connects the instrument data bus ($\overline{\text{IB0}}$-$\overline{\text{IB7}}$) to the MPU data bus D0-D7. This port can be used to pass commands and secondary addresses that aren't automatically interpreted through to the MPU for inspection.

Command Pass-Through Register
(Read Only)

B7	B6	B5	B4	B3	B2	B1	B0

An 8-bit input port used to pass commands and secondary addresses to MPU which are not automatically interpreted by the GPIA.

PROGRAMMING CONSIDERATIONS

The following is a list of considerations when using the MH version of the MC68488:

1. Handshake Interruption

   Once a handshake sequence begins on the IEEE-488 bus it should be allowed to complete in a normal fashion, as described in the IEEE Standard. If this sequence is interrupted (e.g., the controller forces the DAV line to the not data valid state prematurely) the integrity of the data is lost and the interface devices can go to unintended states, as explained in the standard. NOTE: The MC68488 does not interrupt a handshake. It always allows the handshake to complete in the correct sequence; however, it is possible for a device, other than a MC68488, connected to the bus to interrupt the sequence. The controller can do this through an asynchronous Bus Take-over (asserting the ATN line during a handshake). If this occurs the controller should follow the asynchronous take-over with an IFC Uniline Command (ATN can be either asserted or not asserted at this time). It is also possible for some devices to interrupt the handshake by prematurely making the DAV line false (this type of interrupt should be avoided and it should be noted that the MC68488 does not interrupt the handshake sequence). If the DAV handshake line is made false (high) before DAC is made false (high) during a handshake sequence, the listener GPIA(s) will respond as follows:

   a) If IFC is sent by the controller with ATN false before another handshake sequence is initiated the MC68488 will reset back to an idle state. The GPIA at this point is ready to be reprogrammed.

The BI status bit may be set, depending on when the handshake was interrupted, but any byte in R7R cannot be considered valid. NOTE: If IFC is sent with ATN true, it must be sent again with ATN false.

b) If another handshake is initiated before IFC is sent with ATN false, the GPIA does not generate interrupts for subsequent data bytes received by the listener GPIA(s). The device responds to commands and moves into and out of TACS, LACS, etc., but no further BI interrupts are generated. The only solutions to this situation are to reset the MC68488 or have the MPU perform a read of R7R register in the GPIA.

2. Interrupt Structure

   The status bits in R0R, when set, cause an interrupt (drives the $\overline{\text{IRQ}}$ line low) if the appropriate interrupt mask bits in R0W are set. The $\overline{\text{IRQ}}$ line is sensitive to a low-to-high transition produced by the logical OR of the appropriate bits in R0R. If, for example the BI status bit is set and causes an $\overline{\text{IRQ}}$ interrupt, the MPU reads R0R (this read will reset the $\overline{\text{IRQ}}$ line but not the status bit) and detect that the BI bit is set. The software should then direct the MPU to read the data byte from R7R, which in turn causes the BI bit to be reset. If after the status register (R0R) was read and before R7R is read, another interrupt status bit is set (e.g., the CMD bit) this second condition does not cause an interrupt. The BI bit being set at the time CMD occurred prevents the $\overline{\text{IRQ}}$ line from detecting the necessary low-to-high transition and an interrupt could be missed. To prevent this, the last set of instructions in the software interrupt handler should be a reset of the interrupt mask register, followed by programming this same register to its original state. This always produces the needed low-to-high transition, preventing missed interrupts.

3. The "nba" for TACS affects "nba" for SPAS

   If nba for TACS is false (there is not a data byte pending in R7W) then the serial poll handshake sequence for the status byte to the controller occurs once. If nba for TACS is true (there is a data byte in R7R waiting for a handshake to listeners) then the status byte will be sent to the controller each time the controller completes a handshake and indicates that it is ready for more data.

4. The "nba" for SPAS affects "nba" for TACS

   The controller places the GPIA into the Serial Poll Active State (SPAS) by sending serial Poll enable, sending the device talk address, and then releasing ATN. If the controller does this and never accepts the serial Poll Status byte (never makes the RFD handshake line true) but rather the controller asserts ATN and sends Serial Poll Disable (SPD), then the GPIA moves into and out of SPAS without completing the status byte handshake routine. In this state the "nba" for SPAS remains true and affects "nba" for TACS in the following way:

   When the controller places the GPIA in TACS the part makes DAV true as soon as RFD is made true by the listeners in a normal sequence. However, the GPIA continues the handshake sequence, using the contents of R7W, over and over. (i.e., Each time the listeners accept the current data byte and makes RFD true, the GPIA makes DAV true automatically and begins another handshake sequence.) The B0 status bit is set, however, and if the MPU writes to R7W the new data byte is sent to the listeners over and over, using a handshake routine. This continual sending of data bytes occurs until the controller places the GPIA back in SPAS and completes the handshake routine for the Serial Poll Status byte making "nba" for SPAS false again.

   This situation does not occur if the controller handshakes the status byte when it places the GPIA in SPAS.

5. Dual Addressing

   Dual addressing implies the use of two adjacent primary addresses and, as such, care should be taken when selecting the primary addresses for this mode. Decimal address 30 (11110) should not be used because the dual address counterpart of decimal 30 is decimal 31 (11111). Since address 31 has the same bit code as that of either the Untalk or Unlisten Commands this value is an invalid primary address for the IEEE-488 system.

6. "Ghost Interrupts"

   A "ghost interrupt" is an interrupt that occurs as a result of the MC68488, but when the status register is checked no status bits are set. There are two conditions that can legitimately cause a "ghost interrupt." They are:

   a) SPAS status bit

      If the controller conducts a serial poll by sending Serial Poll Enable (SPE) and then sends the GPIA talk address, the SPAS status bit is set and can cause an interrupt. After the controller receives the Serial Poll Status byte it will send Serial Poll Disable (SPD) which resets the SPAS status bit. If the controller can perform this sequence of events before the interrupt handler can check the SPAS status bit, the MPU will not find any status bits set ("ghost interrupt"). The possibilities are twofold:

      1) If this device had actually requested the service, then the MPU, after receiving the interrupt ("ghost" or not), should check bit 6 of the Serial Poll register. If this bit is reset the MPU knows that a Serial Poll was conducted and can reset the rsv as per normal Serial Poll handling procedures.
      2) If this device did not request the service request and SPAS is not set, the software should detect this as a "ghost interrupt," ignore it, and proceed with normal operations.

   See "Serial Poll Procedure" (#11) for further Serial Poll operation.

b) B0 status bit

The B0 Status bit is set whenever the MC68488 is in the Talker Active State and the output register (R7W) is empty. After the listener(s) accept the current data byte on the IEEE bus, the B0 status bit will again be set and with the appropriate mask bits set, will cause an interrupt. When the talker sends the last byte of a string it is possible for the controller to detect this, synchronously take control of the bus, and untalk the talker; however, when the last byte is accepted the B0 status bit is again set and if so programmed, causes another interrupt. It is possible for the controller to untalk the device thereby resetting B0 before the MPU interrupt handler is able to check the status register. Under these conditions a "ghost interrupt" occurs. See "Send Last Byte Procedure" (#10) for further description and solution.

7. UACG Status Bit

The UACG status bit is set anytime the GPIA receives an Undefined Address Command Group (UACG) message from the controller. This bit is not qualified with the addressed state of the part. The MPU software, after detecting a UACG, must check the ma bit in the address status register to see if the device is addressed. If the UACG message is a selected command only pertinent to addressed listeners, the software, after receiving the command by reading R6R, should release the handshake (write dacr high in R3W). This allows the controller to make ATN false. If the device has been addressed to listen/talk and ATN is made false the LACS/TACS status bit in the address status register will be set. The MPU can then check these bits.

8. END Status Bit

The END status bit in R0R is used to indicate to addressed listeners that the next byte received by the addressed talker is the last byte of a string. This bit is not qualified with the handshake and thus occurs ahead of the reception of the last data byte. This alerts the MPU that the final byte will soon follow. Because of this, two interrupts, if so programmed, will occur. One for the END bit and one for the BI bit when the final byte is transferred with a handshake. For those situations where it is inconvenient to have two interrupts the END status bit can be masked, not allowing it to cause an interrupt.

9. feoi (force end or identify)

This control bit (bit 5, R3W) is used when the MC68488 is an Active Talker, to indicate to the listener(s) on the IEEE bus the end of a data string transfer. The MC68488 asserts the EOI management line when the feoi control bit is set and the device is in the Talker Active State (TACS). The feoi bit is set by the MPU writing this bit high and automatically resets one E clock cycle after it was set. The use of this function is as follows: When sending a string of data the feoi control bit should be set prior to sending the final data byte. This causes the EOI management line to be asserted (low). The final data byte can now be sent. The EOI line remains asserted until this byte is accepted, at which time it returns high.

Care must be used when setting the feoi control bit. Once feoi has been written high, the EOI line is asserted when the MC68488 is an Active Talker and remains asserted until the next data byte is sent and accepted. This is true even if feoi is written high while the device is not an Active Talker. In this case the EOI management line is asserted as soon as the MC68488 is again made an Active Talker. Once the feoi control bit is set, only a device reset prevents the END message from being sent when the MC68488 becomes an Active Talker.

10. Send Last Byte Procedure (Talker Mode)

The procedure used for sending the last byte is described below. When using the $\overline{\mathrm{EOI}}$ management line, the MPU software must first set the feoi control bit (asserting $\overline{\mathrm{EOI}}$), and then send the last byte. When the last byte is accepted by all listeners, the B0 status bit of the talker device is set. The B0 status bit is not qualified with the $\overline{\mathrm{EOI}}$ line, but is set whenever the current data byte is accepted by all listeners and the device is in the Talker Active State (TACS). (Note that when the controller asserts $\overline{\mathrm{ATN}}$ to send commands, the GPIA moves out of TACS causing B0 to reset and remains out of TACS as long as $\overline{\mathrm{ATN}}$ is asserted.) After the data block transfer, the controller takes control of the bus (asserts $\overline{\mathrm{ATN}}$) and reconfigures the GPIB system. In performing this task, the controller sends command(s) that untalk the device (MLA, OTA, UNT) or reassigns it as a Talker (MTA) asking for further data transfers. Since the GPIB operates asynchronously with respect to the device MPU bus, it is possible for the controller to take control of the GPIB and cause actions that change the state of the B0 status bit in the middle of the MPU interrupt routine. As a result, care needs to be exercised when responding to the B0 status bit interrupt occurring after transferring the last byte. Any of the following conditions can occur.

1) Device Untalked — If either My Listen Address, Other Talk Address, or the Untalk command is sent, the device is placed in the Talker Idle State (TIDS) — the device is Untalked. In this case the B0 status bit is set as soon as the last data byte is accepted, reset when the controller asserts $\overline{\mathrm{ATN}}$, and B0 will remain reset after $\overline{\mathrm{ATN}}$ is released.
   (a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{\mathrm{ATN}}$. This status indication, however, is misleading as another byte transfer is not intended. The device is soon to be Untalked.
   (b) The B0 status bit indicates a reset condition if the MPU reads the Interrupt Status Register after $\overline{\mathrm{ATN}}$ has been asserted — a "ghost interrupt" is produced. This B0 status bit remains reset after $\overline{\mathrm{ATN}}$ is made false (high).
2) Device Reassigned as a Talker — The controller reassigns the device to talk by sending My Talk Address. In this case the B0 status bit is set as

soon as the last data byte is accepted, reset when the controller asserts $\overline{ATN}$ to send MTA, and is again set when $\overline{ATN}$ is made false by the controller.

(a) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register before the controller asserts $\overline{ATN}$. This case is identical to part (a) for "Device Untalked" shown above.

(b) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register while $\overline{ATN}$ is asserted – a "ghost interrupt" is produced.

(c) The B0 status bit indicates a set condition if the MPU reads the Interrupt Status Register after $\overline{ATN}$ is made false (high). This status indication is requesting a byte transfer and should be acted upon accordingly.

To alleviate the above ambiguity and "ghost interrupt" situation, the GPIB handshake must be synchronized with action by the device MPU. The following step-by-step procedure provides this needed synchronization and eliminates the ambiguity when servicing the B0 status bit after sending the last byte.

1) Before sending the last byte of a block transfer, the feoi bit (if used) should be set. In addition, the dacd bit in R3W should be set, holding off the handshake upon reception of any command (establishes the required synchronization between MPU and controller).
2) If operating under interrupts, the B0 interrupt mask should be reset. This prevents generation of a B0 status interrupt when the last byte is received.
3) Send the last data byte.
4) The MPU now monitors the ATN bit in the Address Status Register (R2R). When the ATN bit is set, the $\overline{ATN}$ line has been asserted and it will remain asserted until completion of the handshake. The procedure, described herein, assumes that $\overline{ATN}$ line is asserted between block transfers and at least one command sent. The fact that $\overline{ATN}$ is asserted indicates that the device is no longer in TACS and, thus, the B0 status bit is reset.
5) The dacd bit in R3W can now be written low, removing the manual handshake hold-off on subsequent commands. With the same write instruction, the dacr bit should be set, releasing the handshake on the current command (write a hex 10 to R3W).
6) The B0 interrupt mask bit can now be set, enabling interrupts for another block transfer.

After following this procedure, a B0 status condition will occur only if a second block of data is requested by the controller. In addition, the possibility of a B0 "ghost interrupt" is eliminated.

11. Serial Poll Procedure

The MPU initiates a service request by writing rsv (bit 6, R5W) high in the GPIA. At the same time, the appropriate code should be placed in the other 7 bits. Bit 6 being set causes the $\overline{SRQ}$ management line to go low. The GPIA enters the Serial Poll Active State (SPAS) when it receives SPE and is an active talker. When it enters SPAS, the following occurs: the SPAS status bit (bit 2, R1R) is set, the CMD status bit (bit 2, R0R) is set, the $\overline{SRQ}$ line is asserted passively false (high), the SRQ status bit (bit 6, R5R) is reset, and the contents of R5R is placed on the GPIB data bus.

When the GPIA enters SPAS, the SPAS status bit (R1R) is set. This, in turn, causes the CMD status bit in R0R to be set. In an interrupt driven system with the CMD and IRQ mask bits set, this causes an MPU interrupt. These status bits are not latched conditions and only monitor the current state of the GPIA. If the controller places the GPIA in SPAS (sends SPE and MTA), receives the Serial Poll status byte and removes the GPIA from SPAS (sends SPD) before the MPU reads the Interrupt Status register, the contents of this register shows hex 10. Since the MPU knows that this device issued the service request, it should check bit 6 of R5W if an MPU interrupt is generated but no status bit is set. If bit 6, R5R, is reset, the MPU will know the controller has performed a Serial Poll on it. However, the SRQ status bit being reset does not indicate that the status byte was accepted by the controller – that is, the handshake was completed. Rather, it indicates that the GPIA has been placed in SPAS and that the status byte has been placed on the GPIB. In systems with slow responding controllers, the SRQ bit in R5R can be reset while the SPAS status bit is still set. In this case to determine when the status byte was accepted, the MPU can monitor SPAS status bit. This bit is reset when the controller has removed the GPIA from the SPAS. Once in SPAS, the controller must accept the Serial Poll byte before removing the device from SPAS. The rsv bit cannot be written low until the status byte has been accepted, but should be written low as soon as the status byte has been accepted by the controller.

If this device has issued a service request to the controller, the following provides a procedure for handling a SPAS interrupt. The procedure only discusses Serial Poll (SPAS) interrupts. Interrupts resulting from other sources need to be incorporated as appropriate for the system application. In an interrupt driven system, the MPU normally reads the Interrupt Status Register to find the cause of the interrupt. The Interrupt Status Register must be read to release the $\overline{IRQ}$ line and, in most cases, it will be read to check if something other than SPAS caused the interrupt. However, since it is possible that the SPAS status can be set and then reset before the MPU reads the register, the following procedure should also be used (even though the SPAS status is reset).

1) The MPU should monitor the SRQ bit in the Serial Poll Register. This can occur as a result of either an interrupt or a polling routine.
2) When the SRQ bit returns to zero, it indicates that the MC68488 has been placed in the Serial Poll Active State (SPAS). This does not mean that the device is in SPAS, because the con-

troller could have placed the MC68488 in SPAS and then removed the device from SPAS before the MPU reads the Serial Poll Register (R5R).

3) After the SRQ bit in R5R returns to zero, the MPU should read the Command Status Register and Monitor the SPAS status bit. When this bit returns to 0, it indicates that the Serial Poll Status byte has been accepted by the controller and that the MC68488 has been removed from the Serial Poll Active State (SPAS).
4) After the SPAS status bit returns to 0, the rsv bit (in R5W) should be written low.

The GPIA uses the source handshake to send the Serial Poll status byte to the controller. The GPIA does this by placing the status byte on the GPIB, and when the controller makes RFD true, the GPIA makes $\overline{\mathrm{DAV}}$ true (low), and the handshake takes place according to the IEEE-488 Standard handshake protocol. If nba for the GPIA TACS function is false at this time, the GPIA will send this byte only once; i.e., the GPIA does not make $\overline{\mathrm{DAV}}$ true (low) a second time. If nba for the GPIA TACS function is true at this time, the GPIA sends this byte over and over, provided the controller continually makes RFD true at the end of the handshake without reconfiguring the device; i.e., the GPIA in this situation makes $\overline{\mathrm{DAV}}$ true (low) each time it receives an RFD true from the controller. The only time nba can be true for TACS is if the device was an active talker prior to the Serial Poll sequence, and the GPIA MPU had loaded a byte in R7W. Now, if the controller synchronously takes over the bus before this byte is placed on the GPIB, the nba for TACS will be true.

NOTE

After a Serial Poll has been conducted on the GPIA, and the SRQ bit (bit 6, R5W = 0) is reset, the MPU must write the rsv (bit 6, R5W) low before another service request can be initiated.

APPENDIX

GPIA MASK SET DIFFERENCES

There have been two mask sets produced for the GPIA (MC68488). They are:

G6G MASK SET – sampled in the fall of '77 (first mask set). This mask set was produced through December of 1978 and can be identified by the letters "GG" preceding the date code on top of the package.

M2H MASK SET – parts available January '79 (final mask set). Any parts ordered after this date will be M2H parts. The M2H mask set replaces the G6G mask set and can be identified by the letters MH or M2H preceding the date code on top of the package. The mask set designation for later production runs is P9W. The P9W mask set is identical to the M2H in all aspects.

There are seven areas of differences between the G6G and M2H/P9W mask sets. They are:

1. RLC Status bit

   This bit is used to implement the Remote/Local interface function. In the GG mask set the Remote/Local option should not be used, because the RLC status bit in R1R will lock up in the zero state. In the MH mask version the RLC bit is completely functional and will report any change in the REM status bit.

2. Extended Addressing

   The GG mask version of the GPIA will not discontinue secondary addressing when the primary address of another GPIA is sent by the controller; i.e., after entering LPAS, the primary address of another device will not transfer the GPIA to LPIS. This transition from LPAS to LPIS was not fully implemented in the GG mask set. The MH mask set has fully implemented this interface function. With this version, if the GPIA is programmed for extended addressing and receives its primary address, it will move to LPAS. If at this point the primary address of another controller is sent, the GPIA will go to its idle state (LIDS/TIDS) as per IEEE-488 1978 standard requirements.

3. TPAS and LPAS Status Bits

   In the GG mask set the LPAS status bit will report either LPAS or LADS. Likewise, the TPAS status bit will report either TPAS or TADS. In the MH mask set these bits only report LPAS and TPAS respectively.

4. DAC Release

   When the GPIA (GG mask set) in the listener mode receives a byte of data from the IEEE bus the DAC handshake will be held off until the MPU reads this data byte. The E-pulse that reads this data from R7R also releases DAC, indicating to the talker that the byte has been accepted. In the GG mask set the DAC handshake line is released on the low-to-high transition (leading edge) of the E-pulse, but the data is actually read (accepted by the MPU) on the high-to-low transition (trailing edge). If there is a very fast talker or long E-pulse width, it is possible for the talker to receive a data accept (DAC) and place the next data byte in the Data-In Register before the current one has been read out by the MPU. This will overwrite the data on the MPU bus and result in missed data. For this to occur the talker must be able to detect the DAC line going high and place the next data byte on the bus (making DAV true) before the E-pulse goes low. For a 1 MHz E-pulse this is approximately 400 ns. The MH or M2H mask set corrects this by releasing DAC after the trailing edge of the E-pulse.

5. dsel (deselect)

   One of the functions of the dsel bit (bit 7 of R2W) is to deselect the Group Execute Trigger (GET) command from setting the GET status bit and causing an interrupt. The GG mask version of the GPIA, when dsel is set, prevents the GET status bit from being set, but it is still possible to get an IRQ (IRQ output goes low), if enabled, when the GET command is detected. Thus, dsel inhibits the GET status condition, but not the associated interrupt. The result is a "ghost interrupt" whenever the controller sends the GET command. The MH mask set, when in dsel mode, inhibits both GET status and its associated interrupt and eliminates this "ghost interrupt."

6. hold-on-all-data (hlda)

   When in the listener mode, the GPIA provides a means of holding off the handshake on reception of data until the MPU releases the handshake. This mode occurs if the hlda (hold-on-all-data) bit in R2W is set. The MPU releases the handshake by writing rfdr (ready-for-data-release) in R3W high. In the GG mask version, if while receiving data in the listener mode the controller takes over synchronously and makes the GPIA a talker and then changes the GPIA at a later time, back to a listener, the RFD handshake will be held off on the listener command rather than waiting for the first data byte. The MH mask only holds off the handshake on data and does not hold off the handshake on any command.

7. new-byte-available (nba) During a Serial Poll

   In the GG mask version, if the GPIA had been in the talker active state prior to the controller conducting a serial poll, it is possible for nba to be lost. The situation is as follows: if a byte of data has been written into R7W and the controller takes over the bus synchronously at SDYS (SH state diagram, Figure 3, page 20, IEEE-488 1978 Specification) to perform a serial poll, the GG mask version of the GPIA will respond in one of two ways:

   a) If the controller never requests the contents of the Serial Poll Register from the GPIA, and when the GPIA is returned as a talker, the data in R7W is available to the listeners on the bus. (Data byte is not destroyed).

   b) If the controller requests the contents of the Serial Poll Register from the active talker, when the controller returns the GPIA to the Active Talker State, the data which was in R7W will have been handshaked as though it had been accepted by the active listeners (data byte will be destroyed).

The original IEEE Standard had a discrepancy as to what happens to this data byte (nba) under these circumstances. This discrepancy has been alleviated. The MH mask conforms to the latest revision and does not destroy the data in R7W when a Serial Poll occurs; i.e., if in TACS with a nba pending when the controller releases the bus to the talker, the byte in R7W will be transferred, via handshake, to the listeners (data byte is not destroyed).

SOFTWARE DIFFERENCES BETWEEN MASK SETS

The seven changes mentioned in the previous sections are the only changes from the GG to the MH mask set. All of these changes except number 3 (TPAS and LPAS status bits) are transparent to the user software.

The change to TPAS and LPAS status bits is a functional change. In the GG mask, user software could monitor LPAS and TPAS for address recognition in the primary address mode because LPAS is set as soon as the GPIA receives its Listen Address (MLA) and TPAS is set as soon as the GPIA receives its Talker Address (MTA); i.e., the LPAS bit is set when the GPIA enters LADS and the TPAS bit is set when the GPIA enters TADS. In the MH mask set these bits do not report LADS/TADS and as such they can only be used in the extended address mode. In the primary address mode the software for the MH mask set should monitor LACS/TACS (bits 2 and 3 of the address status register) rather than LPAS/TPAS: TACS/LACS indicates when the device is in the Talker/Listener Active State.

MOTOROLA

MC2652/MC68652
MC2652_2/MC68652_2

Advance Information

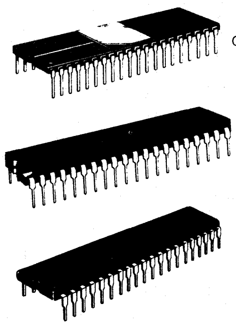
MOS
(N-CHANNEL, SILICON-GATE)

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER (MPCC)

The MC2652/MC68652 MPCC formats, transmits, and receives synchronous serial data while supporting Bit-Oriented (BOP) or Byte-Control (BCP) protocols. The parallel bus of the MPCC readily interfaces with M6800 and M68000 Microprocessor Families as well as many other 8- or 16-bit processors. Typical applications include intelligent terminals, front-end communications, remote-data concentrators, communication test equipment, and computer-to-computer links.

- DC to 2 Mbps Data Rate
- Bit-Oriented Protocols (BOP): SCLC, ADCCP, HDLC, X.25
  - Character Length—1-to-8 Bits
  - Address Comparison
  - Automatic Detection and Generation of Special Control Characters, i.e., FLAG, ABORT, GA
  - Automatic Zero Insertion and Deletion
  - Short Last Character
  - Idle Transmission of FLAG or ABORT Characters
  - Automatic Generation and Checking of CRC-CCITT FCS
- Byte-Control Protocols (BCP): DDCMP, BISYNC (external CRC)
  - Character Length—5-to-8 Bits
  - SYNC Generation Detection and Stripping
  - Idle Transmission of SYNC or MARK Characters
  - Automatic Generation and Checking of CRC-16 or VRC
- Maintenance Mode for Self-Checking
- Bidirectional, Three-State, 8- or 16-Bit Data Bus
- TTL Compatible
- Compatible with MC2653/MC68653 Polynomial Generator Checker



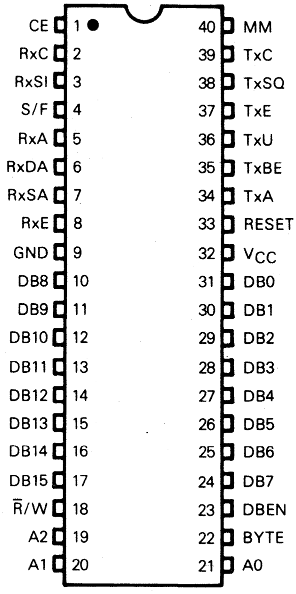
L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

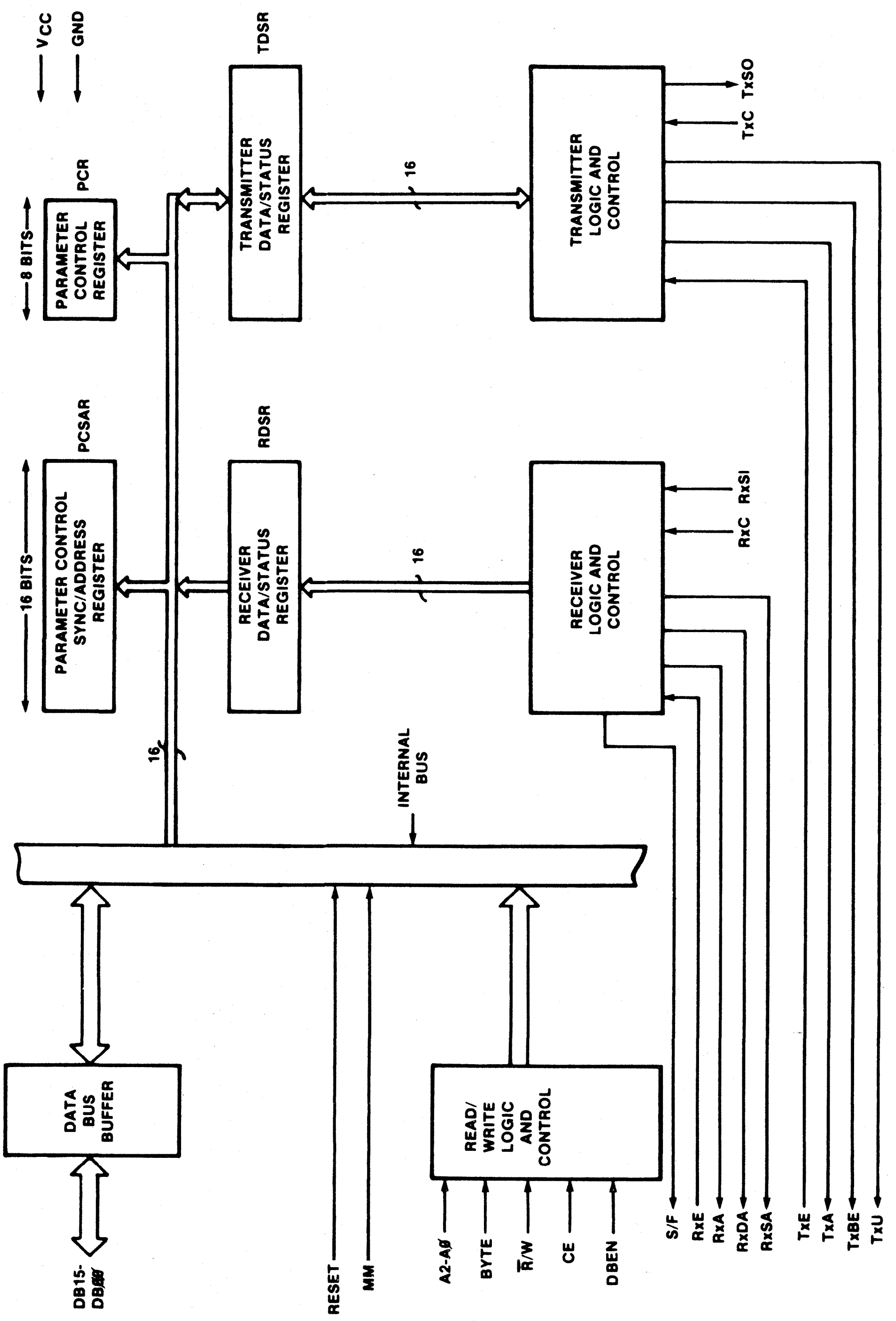
P SUFFIX
PLASTIC PACKAGE
CASE 711

"_" = PACKAGE SUFFIX

PIN ASSIGNMENT



ADI-870



BLOCK DIAGRAM

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DB15-DB00	17-10 24-31	I/O	**Data Bus:** DB07-DB00 contain bidirectional data while DB15-DB08 contain control and status information to or from the processor. Corresponding bits of the high and low order bytes can be WIRE OR'ed onto an 8-bit bus. The data bus is floating if either CE or DBEN are low.
A2-A0	19-21	I	**Address Bus:** A2-A0 select internal registers. The four 16-bit registers can be addressed on a word or byte basis. See Register Address section.
BYTE	22	I	**Byte:** Single byte (8 bit) data bus transfers are specified when this input is high. A low level specifies 16 bit data bus transfers.
CE	1	I	**Chip Enable:** A high input permits a data bus operation when DBEN is activated.
$\overline{\text{R}}$/W	18	I	**Read/Write:** $\overline{\text{R}}$/W controls the direction of data bus transfer. When high, the data is to be loaded into the addressed register. A low input causes the contents of the addressed register to be presented on the data bus.
DBEN	23	I	**Data Bus Enable:** After A2-A0, CE, BYTE and $\overline{\text{R}}$/W are set up, DBEN may be strobed. During a read, the 3-state data bus (DB) is enabled with information for the processor. During a write, the stable data is loaded into the addressed register and TxBE will be reset if TDSR was addressed.
RESET	33	I	**Reset:** A high level initializes all internal registers (to zero) and timing.
MM	40	I	**Maintenance Mode:** MM internally gates TxSO back to RxSI and $\overline{\text{TxC}}$ to RxC for off line diagnostic purposes. The RxC and RxSI inputs are disabled and TxSO is high when MM is asserted.
RxE	8	I	**Receiver Enable:** A high level input permits the processing of RxSI data. A low level disables the receiver logic and initializes all receiver registers and timing.
RxA	5	O	**Receiver Active:** RxA is asserted when the first data character of a message is ready for the processor. In the BOP mode this character is the address. The received address must match the secondary station address if the MPCC is a secondary station. In BCP mode, if strip-SYNC (PCSAR$_{13}$) is set, the first non-SYNC character is the first data character; if strip-SYNC is zero, the character following the second SYNC is the first data character. In the BOP mode, the closing FLAG resets RxA. In the BCP mode, RxA is reset by a low level at RxE.
RxDA*	6	O	**Receiver Data Available:** RxDA is asserted when an assembled character is in RDSR$_L$ and is ready to be presented to the processor. This output is reset when RDSR$_L$ is read.
RxC	2	I	**Receiver Clock:** RxC(1X) provides timing for the receiver logic. The positive going edge shifts serial data into the RxSR from RxSI.
S/F	4	O	**SYNC/FLAG:** S/F is asserted for one RxC clock time when a SYNC or FLAG character is detected.
RxSA*	7	O	**Receiver Status Available:** RxSA is asserted when there is a zero to one transition of any bit in RDSR$_H$ except for RSOM. It is cleared when RDSR$_H$ is read.
RxSI	3	I	**Receiver Serial Input:** RxSI is the received serial data. Mark = '1', space = '0'.
TxE	37	I	**Transmitter Enable:** A high level input enables the transmitter data path between TDSR$_L$ and TxSO. At the end of a message, a low level input causes TxSO = 1 (mark) and TxA = 0 after the closing FLAG (BOP) or last character (BCP) is output on TxSO.
TxA	34	O	**Transmitter Active:** TxA is asserted after TSOM (TDSR$_8$) is set and TxE is raised. This output will reset when TxE is low and the closing FLAG (BOP) or last character (BCP) has been output on TxSO.
TxBE*	35	O	**Transmitter Buffer Empty:** TxBE is asserted when the TDSR is ready to be loaded with new control information or data. The processor should respond by loading the TDSR which resets TxBE.
TxU*	36	O	**Transmitter Underrun:** TxU is asserted during a transmit sequence when the service of TxBE has been delayed for one character time. This indicates the processor is not keeping up with the transmitter. Line fill depends on PCSAR$_{11}$. TxU is reset by RESET or setting of TSOM (TDSR$_8$), synchronized by the falling edge of TxC.
TxC	39	I	**Transmitter Clock:** TxC (1X) provides timing for the transmitter logic. The positive going edge shifts data out of the TxSR to TxSO.
TxSO	38	O	**Transmitter Serial Output.** TxSO is the transmitted serial data. Mark = '1', space = '0'.
V$_{CC}$	32	I	**+5V:** Power supply.
GND	9	I	**Ground:** 0V reference ground.

*Indicates possible interrupt signal.

REGISTERS		NO. OF BITS	DESCRIPTION*
Addressable			
PCSAR	Parameter Control Sync/Address Register	16	PCSAR_H and PCR contain parameters common to the receiver and transmitter. PCSAR_L contains a programmable SYNC character (BCP) or secondary station address (BOP).
PCR	Parameter Control Register	8	
RDSR	Receive Data/Status Register	16	RDSR_H contains receiver status information. RDSR_L = RxDB contains the received assembled character.
TDSR	Transmit Data/Status Register	16	TDSR_H contains transmitter command and status information. TDSRL = TxDB contains the character to be transmitted.
Internal			
CCSR	Control Character Shift Register	8	These registers are used for character assembly (CCSR, HSR, RxSR), disassembly (TxSR), and CRC accumulation/generation (RxCRC, TxCRC).
HSR	Holding Shift Register	16	
RxSR	Receiver Shift Register	8	
TxSR	Transmitter Shift Register	8	
RxCRC	Receiver CRC Accumulation Register	16	
TxCRC	Transmitter CRC Generation Register	16	

NOTE
*H = High byte - bits 15-8
L = Low byte - bits 7-0

Table 1 GLOSSARY

CHARACTER	DESCRIPTION
FCS	Frame Check Sequence is transmitted/received as 16 bits following the last data character of a BOP message. The divisor is usually CRC-CCITT ($X^{16}+X^{12}+X^5+1$) with dividend preset to 1's but can be otherwise determined by ECM. The inverted remainder is transmitted as the FCS.
BCC	Block Check Character is transmitted/received as two successive characters following the last data character of a BCP message. The polynomial is CRC-16 ($X^{16}+X^{15}+X^2+1$) or CRC-CCITT with dividend preset to 0's (as specified by ECM). The true remainder is transmitted as the BCC.

Table 2 ERROR CONTROL

FUNCTIONAL DESCRIPTION

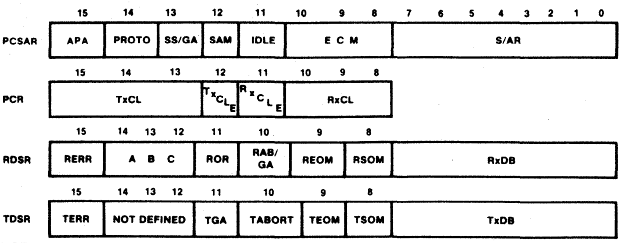
The MPCC can be functionally partitioned into receiver logic, transmitter logic, registers that can be read or loaded by the processor, and data bus control circuitry. The register bit formats are shown in figure 1 while the receiver and transmitter data paths are depicted in figures 2 and 3.

OPERATION	BIT PATTERN	FUNCTION
BOP		
FLAG	01111110	Frame message
ABORT	11111111 generation 01111111 detection	Terminate communication
GA	01111111	Terminate loop mode repeater function
Address	$(\text{PCSAR}_L)^1$	Secondary station address
BCP		
SYNC	(PCSAR_L) or $(\text{TxDB})^2$ generation	Character synchronization

NOTES
1. (∝) refers to contents of ∝
2. For IDLE = 0 or 1 respectively

Table 3 SPECIAL CHARACTERS

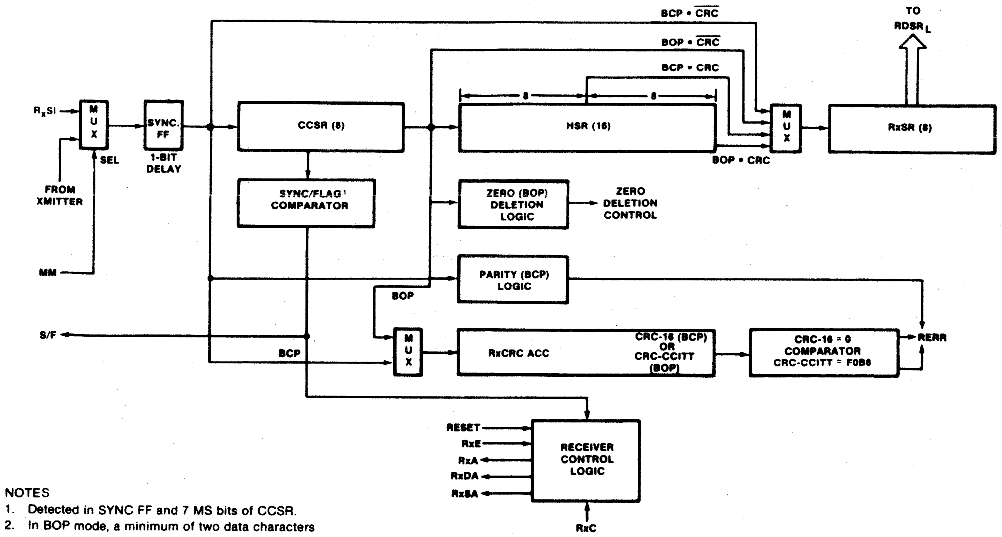
SHORT FORM REGISTER BIT FORMATS



NOTE
Refer to Register Formats for mnemonics and description.

Figure 1

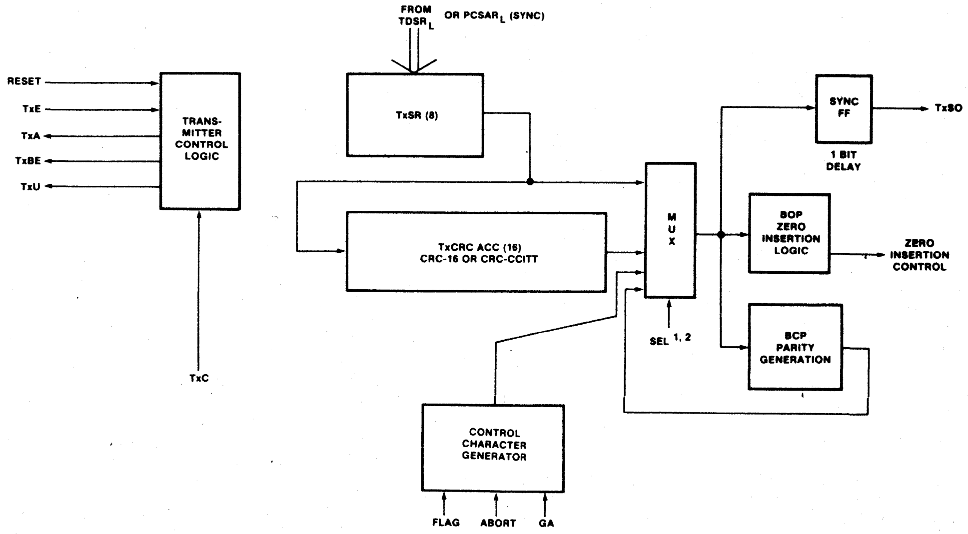
MPCC RECEIVER DATA PATH



NOTES
1. Detected in SYNC FF and 7 MS bits of CCSR.
2. In BOP mode, a minimum of two data characters must be received to turn the receiver active.

Figure 2

MPCC TRANSMITTER DATA PATH



NOTES
1. TxCRC selected if TEOM = 1 and the last data character has been shifted out of TxSR.
2. In BCP, parity selected will be generated after each character is shifted out of TxSR.

Figure 3

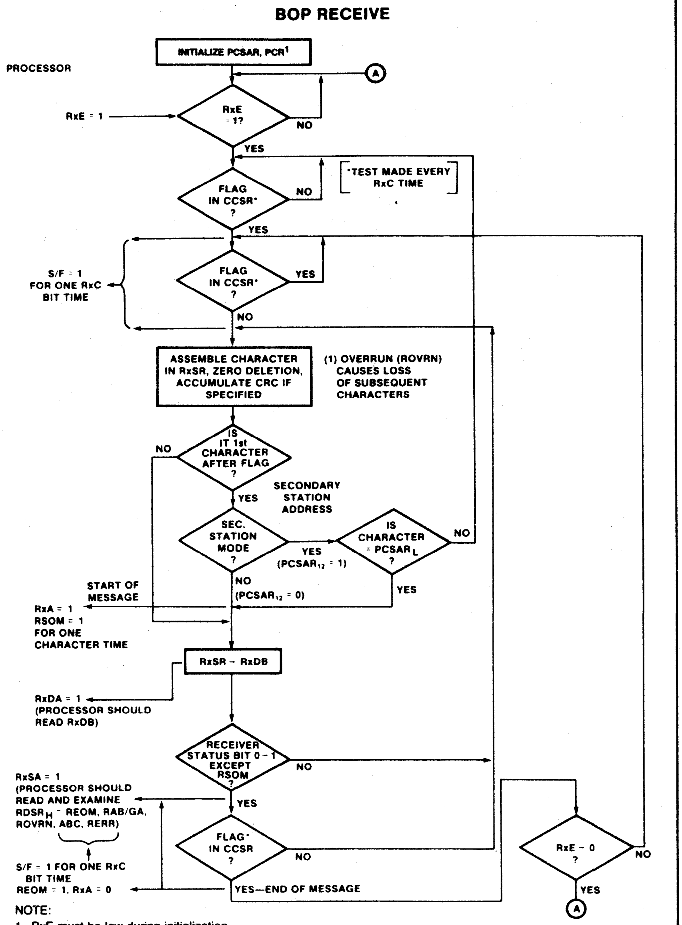
RECEIVER OPERATION

General

After initializing the parameter control registers (PCSAR and PCR), the RxE input must be set high to enable the receiver data path. The serial data on the RxSI is synchronized and shifted into an 8-bit Control Character Shift Register (CCSR) on the rising edge of RxC. A comparison between CCSR contents and the FLAG (BOP) or SYNC (BCP) character is made until a match is found. At that time, the S/F output is asserted for one RxC time and the 16-bit Holding Shift Register (HSR) is enabled. The receiver then operates as described below.

BOP Operation

A flow chart of receiver operation in BOP mode appears in figure 4. Zero deletion (after five ones are received) is implemented on the received serial data so that a data character will not be interpreted as a FLAG, ABORT, or GA. Bits following the FLAG are shifted through the CCSR, HSR, and into the Receiver Shift Register (RxSR). A character will be assembled in the RxSR and transferred to the RDSR$_L$ for presentation to the processor. At that time the RxDA output will be asserted and the processor must take the character no later than one RxC time after the next character is assembled in the RxSR. If not, an overrun (RDSR$_{11}$ = 1) will occur and succeeding characters will be lost.



NOTE:

1. RxE must be low during initialization of PCSAR and PCR.

Figure 4

The first character following the FLAG is the secondary station address. If the MPCC is a secondary station (PCSAR$_{12}$ = 1), the contents of RxSR are compared with the address stored in PCSAR$_L$. A match indicates the forthcoming message is intended for the station; the RxA output is asserted, the character is loaded into RDSR$_L$, RxDA is asserted and the Receive Start of Message bit (RSOM) is set. No match indicates that another station is being addressed and the receiver searches for the next FLAG.

If the MPCC is a primary station (PCSAR$_{12}$ = 0), no secondary address check is made; RxA is asserted and RSOM is set once the first non-FLAG character has been loaded into RDSR$_L$ and RxDA has been asserted. Extended address field can be supported by software if PCSAR$_{12}$ = 0.

When the 8 bits following the address character have been loaded into RDSR$_L$ and RxDA has been asserted, RSOM will be cleared. The processor should read this 8-bit character and interpret it as the Control field.

Received serial data that follows is read and interpreted as the Information field by the processor. It will be assembled into character lengths as specified by PCR$_{8-10}$. As before, RxDA is asserted each time a character has been transferred into RDSR$_L$ and is cleared when RDSR$_L$ is read by the processor. RDSR$_H$ should only be read when RxSA is asserted. This occurs on a zero to one transition of any bit in RDSR$_H$ except for RSOM. RxSA and all bits in RDSR$_H$ except RSOM are cleared when RDSR$_H$ is read. The processor should check RDSR$_{9-15}$ each time RxSA is asserted. If RDSR$_9$ is set, then RDSR$_{12-15}$ should be examined.

Receiver character length may be changed dynamically in response to RxDA: read the character in RxDB and write the new character length into RxCL. The character length will be changed on the next receiver character boundary. A received residual (short) character will be transferred into RxDB after the previous character in RxDB has been read, i.e. there will not be an overrun. In general the last two characters are protected from underrun.

The CRC-CCITT, if specified by PCSAR$_{8-10}$, is accumulated in RxCRC on each character following the FLAG. When the closing FLAG is detected in the CCSR, the received CRC is in the 16-bit HSR. At that time, the Receive End of Message bit (REOM) will be set; RxSA and RxDA will be asserted. The processor should read the last data character in $RDSR_L$ and the receiver status in $RDSR_{9-15}$. If $RDSR_{15} = 1$, there has been a transmission error; the accumulated CRC-CCITT is incorrect. If $RDSR_{12-14} \neq 0$, the last data character is not of prescribed length. Neither the received CRC nor closing FLAG are presented to the processor. The processor may drop RxE or leave it active at the end of the received message.

BCP Operation

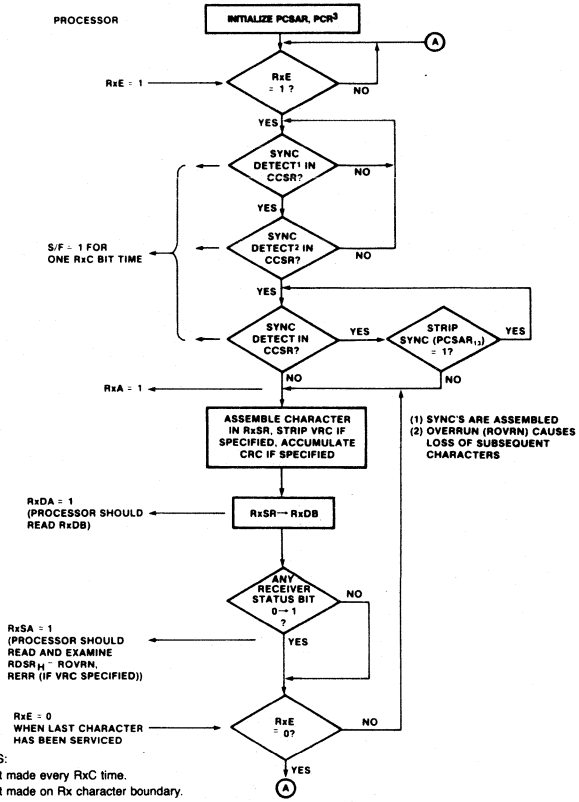
The operation of the receiver in BCP mode is shown in figure 5. The receiver initially searches for two successive SYNC characters, of length specified by PCR_{8-10}, that match the contents of PCSARL. The next non-SYNC character or next SYNC character, if stripping is not specified ($PCSAR_{13} = 0$), causes RxA to be asserted and enables the receiver data path. Once enabled, all characters are assembled in RxSR and loaded into $RDSR_L$. RxDA is active when a character is available in $RDSR_L$. RxSA is active on a 0 to 1 transition of any bit in $RDSR_H$. The signals are cleared when $RDSR_L$ or $RDSR_H$ are read respectively.

If CRC-16 error control is specified by $PCSAR_{8-10}$, the processor must determine the last character received prior to the CRC field. When that character is loaded into $RDSR_L$ and RxDA is asserted, the received CRC will be in CCSR and HSR_L. To check for a transmission error, the processor must read the receiver status ($RDSR_H$) and examine $RDSR_{15}$. This bit will be set for one character time if an error free message has been received. If $RDSR_{15} = 0$, the CRC-16 is in error. The state of RDSR15 in BCP CRC mode does not set RxSA. Note that this bit should be examined only at the end of a message. The accumulated CRC will include all characters starting with the first non-SYNC character if $PCSAR_{13} = 1$, or the character after the opening two SYNC's if $PCSAR_{13} = 0$. This necessitates external CRC generation/checking when supporting IBM's BISYNC. This can be accomplished using the MC68653 Polynomial Generator/Checker. See Typical Applications.

If VRC had been selected for error control, parity (odd or even) is regenerated on each character and checked when the parity bit is received. A discrepancy causes $RDSR_{15}$ to be set and RxSA to be asserted. This must be sensed by the processor. The received parity bit is stripped before the character is presented to the processor.

When the processor has read the last character of the message, it should drop RxE which disables the receiver logic and initializes all receiver registers and timing.

BCP RECEIVE



NOTES:
1. Test made every RxC time.
2. Test made on Rx character boundary.
3. RxE must be low during initialization of PCSAR and PCR.

Figure 5

TRANSMITTER OPERATION

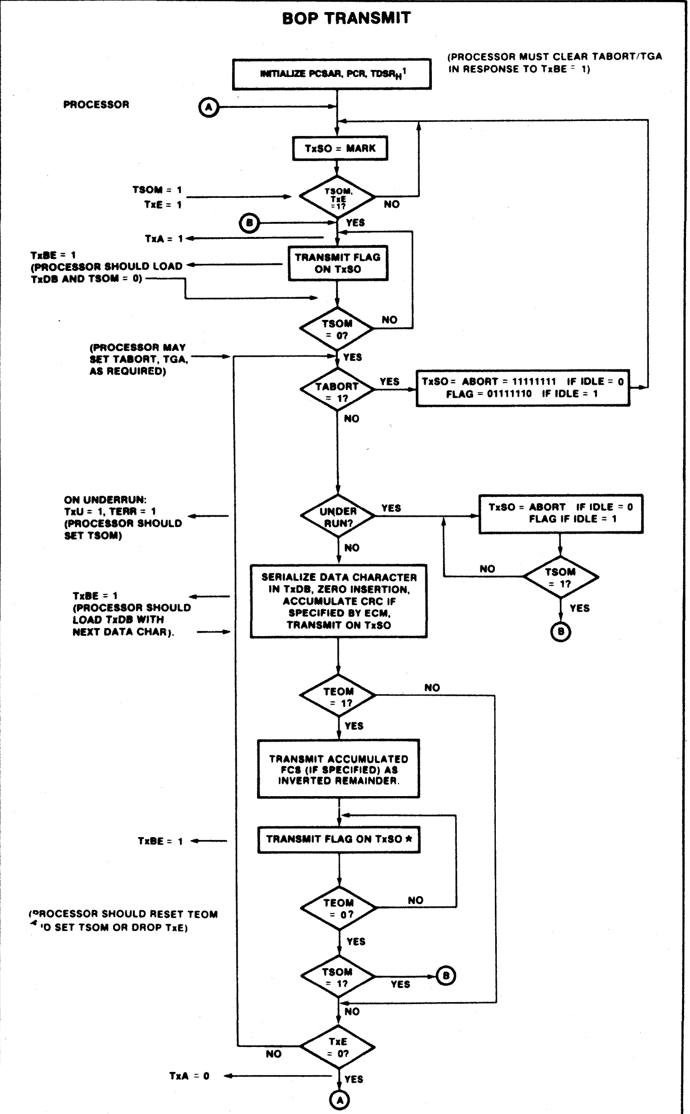
General

After the parameter control registers (PCSAR and PCR) have been initialized, TxSO is held at mark until TSOM ($TDSR_8$) is set and TxE is raised. Then, transmitter operation depends on protocol mode.

BOP Operation

Transmitter operation for BOP is shown in figure 6. A FLAG is sent after the processor sets the Transmit Start of Message bit (TSOM) and raises TxE. The FLAG is used to synchronize the message that follows. TxA will also be asserted. When TxBE is asserted by the MPCC, the processor should load $TDSR_L$ with the first character of the message. TSOM should be cleared at the same time $TDSR_L$ is loaded (16-bit data bus) or immediately thereafter (8-bit data bus). FLAGs are sent as long as TSOM = 1. For counting the number of FLAGs, the processor should reassert TSOM in response to the assertion of TxBE.

All succeeding characters are loaded into $TDSR_L$ by the processor when TxBE = 1. Each character is serialized in TxSR and transmitted on TxSO. Internal zero insertion logic stuffs a "0" into the serial bit stream after five successive "1s" are sent. This insures a data character will not match a FLAG, ABORT, or GA reserved control character. As each character is transmitted, the Frame Check Sequence (FCS) is gener-



★ GA will be transmitted if TGA is set together with TEOM

NOTE:
1. TxE must be low during initialization of PCSAR and PCR.

Figure 6

ated as specified by Error Control Mode (PCSAR$_{8-10}$). The FCS should be the CRC-CCITT polynomial ($X^{16}+X^{12}+X^5+1$) preset to 1s. If an underrun occurs (processor is not keeping up with the transmitter), TxU and TERR (TDSR$_{15}$) will be asserted with ABORT or FLAG used as the TxSO line fill depending on the state of IDLE (PCSAR$_{11}$). The processor must set TSOM to reset the underrun condition. To retransmit the message, the processor should proceed with the normal start of message sequence.

A residual character of 1 to 7 bits may be transmitted at the end of the Information field. In response to TxBE, write the residual character length into TxCL and load TxDB with the residual character. Dynamic alteration of character length should be done in exactly the same sequence. The character length will be changed on the next transmit character boundary.

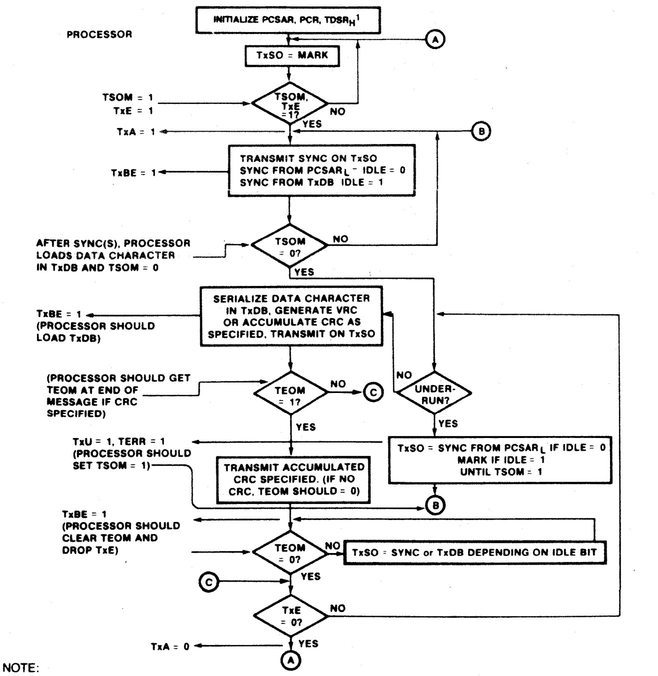
After the last data character has been loaded into TDSR$_L$ and sent to TxSR (TxBE = 1), the processor should set TEOM (TDSR$_9$). The MPCC will finish transmitting the last character followed by the FCS and the closing FLAG. The processor should clear TEOM and drop TxE when the next TxBE is asserted. This corresponds to the start of closing FLAG transmission. When TxE has been dropped, TxA will be low $1^1/_2$ bit times after the last bit of the closing FLAG has been transmitted. TxSO will be marked after the closing FLAG has been transmitted.

If TxE and TEOM are high, the transmitter continues to send FLAGs. The processor may initiate the next message by resetting TEOM and setting TSOM, or by loading TDSR$_L$ with a data character and then simply resetting TEOM (without setting TSOM).

BCP Operation

Transmitter operation for BCP mode is shown in figure 7. TxA will be asserted after TSOM = 1 and TxE is raised. At that time SYNC characters are sent from PCSAR$_L$ or TDSR$_L$ (IDLE = 0 or 1) as long as TSOM = 1. TxBE is asserted at the start of transmission of the first SYNC character. For counting the number of SYNC's, the processor should reassert TSOM in response to the assertion of TxBE. When TSOM = 0 transmission is from TDSR$_L$, which must be loaded with characters from the processor each time TxBE is asserted. If this loading is delayed for more than one character time, an underrun results: TxU and TERR are asserted and the TxSO line fill depend on IDLE (PCSAR$_{11}$). The processor must set TSOM

BCP TRANSMIT



NOTE:

1. TxE must be low during initialization of PCSAR and PCR.

Figure 7

and retransmit the message to recover. This is not compatible with IBM's BISYNC, so that the user must not underrun when supporting that protocol.

CRC-16, if specified by PCSAR$_{8-10}$, is generated on each character transmitted from TDSR$_L$ when TSOM = 0. The processor must set TEOM = 1 after the last data character has been sent to TxSR (TxBE = 1). The MPCC will finish transmitting the last data character and the CRC-16 field before sending SYNC characters which are transmitted as long as TEOM = 1. If SYNCs are not desired after CRC-16 transmission, the processor should clear TEOM and lower TxE when the TxBE corresponding to the start of CRC-16 transmission is asserted. When TEOM = 0, the line is marked and a new message may be iniated by setting TSOM and raising TxE.

If VRC is specified, it is generated on each data character and the data character length must not exceed 7 bits. For software LRC or CRC, TEOM should be set only if SYNC's are required at the end of the message block.

Special Case

The capability to transmit 16 spaces is provided for line turnaround in half duplex mode or for a control recovery situation. This is achieved by setting TSOM and TEOM, clearing TEOM when TxBE = 1, and proceeding as required.

PROGRAMMING

Prior to initiating data transmission or reception, PCSAR and PCR must be loaded with control information from the processor. The contents of these registers (see Register Format section) will configure the MPCC for the user's specific data communication environment. These registers should be loaded during power-on initialization and after a reset operation. They can be changed at any time that the respective transmitter or receiver is disabled.

The default value for all registers is zero. This corresponds to BOP, primary station mode, 8-bit character length, FCS = CRC-CCITT preset to 1s.

For BOP mode the character length register (PCR) may be set to the desired values during system initialization. The address and control fields will automatically be 8-bits. If a residual character is to be transmitted, TxCL should be changed to the residual character length prior to transmission of that character.

DATA BUS CONTROL

The processor must set up the MPCC register address (A2-A0), chip enable (CE), byte select (BYTE), and read/write (R̄/W) inputs before each data bus transfer operation.

During a read operation (R̄/W = 0), the leading edge of DBEN will initiate an MPCC read cycle. The addressed register will place its contents on the data bus. If BYTE = 1, the 8-bit byte is placed on DB15-08 or DB07-00 depending on the H/L status of the register addressed. Unused bits in RDSR$_L$ are zero. If BYTE = 0, all 16 bits (DB15-00) contain MPCC information. The trailing edge of DBEN will reset RxDA and/or RxSA if RDSR$_L$ or RDSR$_H$ is addressed respectively.

DBEN acts as the enable and strobe so that the MPCC will not begin its internal read cycle until DBEN is asserted.

During a write operation (R̄/W = 1), data must be stable on DB$_{15-08}$ and/or DB$_{07-00}$ prior to the leading edge of DBEN. The stable data is strobed into the addressed register by DBEN. TxBE will be cleared if the addressed register was TDSR$_H$ or TDSR$_L$.

	A2	A1	A0	REGISTER
BYTE = 0	**16-BIT DATA BUS = DB$_{15}$ - DB$_{00}$**			
	0	0	X	RDSR
	0	1	X	TDSR
	1	0	X	PCSAR
	1	1	X	PCR*
BYTE = 1	**8-BIT DATA BUS = DB$_{7-0}$ or DB$_{15-8}$****			
	0	0	0	RDSR$_L$
	0	0	1	RDSR$_H$
	0	1	0	TDSR$_L$
	0	1	1	TDSR$_H$
	1	0	0	PCSAR$_L$
	1	0	1	PCSAR$_H$
	1	1	0	PCR$_L$*
	1	1	1	PCR$_H$

NOTES

* PCR lower byte does not exist. It will be all "0"s when read.

** Corresponding high and low order pins must be tied together.

Table 4 MPCC REGISTER ADDRESSING

BIT	NAME	MODE	FUNCTION
00-07	Not Defined		
08-10	RxCL	BOP/BCP	Receiver Character Length is loaded by the processor when RxCLE = 0. The character length is valid after transmission of single byte address and control fields have been received. **10 9 8 Char. length (bits)** 0 0 0 8 0 0 1 1 0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7
11	RxCLE	BOP/BCP	Receiver Character Length Enable should be zero when the processor loads RxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
12	TxCLE	BOP/BCP	Transmitter Character Length Enable should be zero when the processor loads TxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
13-15	TxCL	BOP/BCP	Transmitter Character Length is loaded by the processor when TxCLE = 0. Character bit length specification format is identical to RxCL. It is valid after transmission of single byte address and control fields.

Table 5 PARAMETER CONTROL REGISTER (PCR)-(R/W)

BIT	NAME	MODE	FUNCTION
00-07	S/AR	BOP	SYNC/ADDRESS Register. Contains the secondary station address if the MPCC is a secondary station. The contents of this register is compared with the first received non-FLAG character to determine if the message is meant for this station.
		BCP	SYNC character is loaded into this register by the processor. It is used for receive and transmit bit synchronization with bit length specified by RxCL and TxCL.
08-10	ECM	BOP/BCP	**Error Control Mode — 10 9 8 — Suggested Mode — Char. length** CRC-CCITT preset to 1's — 0 0 0 — BOP — 1-8 CRC-CCITT preset to 0's — 0 0 1 — BCP — 8 Not used — 0 1 0 — --- CRC-16 preset to 0's — 0 1 1 — BCP — 8 VRC odd — 1 0 0 — BCP — 5-7 VRC even — 1 0 1 — BCP — 5-7 Not used — 1 1 0 — --- No error control — 1 1 1 — BCP/BOP — 5-8 ECM should be loaded by the processor during initialization or when both data paths are idle.
11	IDLE		Determines line fill character to be used if transmitter underrun occurs (TxU asserted and TERR set) and transmission of special characters for BOP/BCP.
		BOP	IDLE = 0, transmit ABORT characters during underrun and when TABORT = 1. IDLE = 1, transmit FLAG characters during underrun and when TABORT = 1.
		BCP	IDLE = 0 transmit initial SYNC characters and underrun line fill characters from the S/AR. IDLE = 1 transmit initial SYNC characters from TxDB and marks TxSO during underrun.
12	SAM	BOP	Secondary Address Mode = 1 if the MPCC is a secondary station. This facilitates automatic recognition of the received secondary station address. When transmitting, the processor must load the secondary address into TxDB. SAM = 0 inhibits the received secondary address comparison which serves to activate the receiver after the first non-FLAG character has been received.
13	SS/GA		Strip SYNC/Go Ahead. Operation depends on mode.
		BOP	SS/GA = 1 is used for loop mode only and enables GA detection. When a GA is detected as a closing character, REOM and RAB/GA will be set and the processor should terminate the repeater function. SS/GA = 0 is the normal mode which enables ABORT detection. It causes the receiver to terminate the frame upon detection of an ABORT or FLAG.
		BCP	SS/GA = 1, causes the receiver to strip SYNC's immediately following the first two SYNC's detected. SYNC's in the middle of a message will not be stripped. SS/GA = 0, presents any SYNC's after the initial two SYNC's to the processor.
14	PROTO		Determines MPCC Protocol mode
		BOP	PROTO = 0
		BCP	PROTO = 1
15	APA	BOP	All Parties Address. If this bit is set, the receiver data path is enabled by an address field of '11111111' as well as the normal secondary station address.

Table 6 PARAMETER CONTROL SYNC/ADDRESS REGISTER (PCSAR)-(R/W)

BIT	NAME	MODE	FUNCTION
00-07	TxDB	BOP/BCP	Transmit Data Buffer. Contains processor loaded characters to be serialized in TxSR and transmitted on TxSO.
08	TSOM		Transmitter Start of Message. Set by the processor to initiate message transmission provided TxE = 1.
		BOP	TSOM = 1 generates FLAGs. When TSOM = 0 transmission is from TxDB and FCS generation (if specified) begins. FCS, as specified by PCSAR$_{8-10}$, should be CRC-CCITT preset to 1's.
		BCP	TSOM = 1 generates SYNCs from PCSAR$_L$ or transmits from TxDB for IDLE = 0 or 1 respectively. When TSOM = 0 transmission is from TxDB and CRC generation (if specified) begins.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15)

BIT	NAME	MODE	FUNCTION
09	TEOM	 BOP BCP	Transmit End of Message. Used to terminate a transmitted message. TEOM = 1 causes the FCS and the closing FLAG to be transmitted following the transmission of the data character in TxSR. FLAGs are transmitted until TEOM = 0. ABORT or GA are transmitted if TABORT or TGA are set when TEOM = 1. TEOM = 1 causes CRC-16 to be transmitted (if selected) followed by SYNCs from PCSAR_L or TxDB (IDLE = 0 or 1). Clearing TEOM prior to the end of CRC-16 transmission (when TxBE = 1) causes TxSO to be marked following the CRC-16. TxE must be dropped before a new message can be initiated. If CRC is not selected, TEOM should not be set.
10	TABORT	BOP	Transmitter Abort = 1 will cause ABORT or FLAG to be sent (IDLE = 0 or 1) after the current character is transmitted. (ABORT = 11111111)
11	TGA	BOP	Transmit Go Ahead (GA) instead of FLAG when TEOM = 1. This facilitates repeater termination in loop mode. (GA = 01111111)
12-14	Not Defined		
15	TERR	Read only BOP BCP	Transmitter Error = 1 indicates the TxDB has not been loaded in time (one character time $-1/2$ TxC period after TxBE is asserted) to maintain continuous transmission. TxU will be asserted to inform the processor of this condition. TERR is cleared by setting TSOM. See timing diagram. ABORT's or FLAG's are sent as fill characters (IDLE = 0 or 1) SYNC's or MARK's are sent as fill characters (IDLE = 0 or 1). For IDLE = 1 the last character before underrun is not valid.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15) (Cont'd)

BIT	NAME	MODE	FUNCTION
00-07	RxDB	BOP/BCP	Receiver Data Buffer. Contains assembled characters from the RxSR. If VRC is specified, the parity bit is stripped.
08	RSOM	BOP	Receiver Start of Message = 1 when a FLAG followed by a non-FLAG has been received and the latter character matches the secondary station address if SAM = 1. RxA will be asserted when RSOM = 1. RSOM resets itself after one character time and has no effect on RxSA.
09	REOM	BOP	Receiver End of Message = 1 when the closing FLAG is detected and the last data character is loaded into RxDB or when an ABORT/GA character is received. REOM is cleared on reading RDSR_H, reset operation, or dropping of RxE.
10	RAB/GA	BOP	Received ABORT or GA character = 1 when the receiver senses an ABORT character if SS/GA = 0 or a GA character if SS/GA = 1. RAB/GA is cleared on reading RDSR_H, reset operation, or dropping of RxE. A received ABORT does not set RxDA.
11	ROR	BOP/BCP	Receiver Overrun = 1 indicates the processor has not read last character in the RxDB within one character time $+1/2$ RxC period after RxDA is asserted. Subsequent characters will be lost. ROR is cleared on reading RDSR_H, reset operation, or dropping of RxE.
12-14	ABC	BOP	Assembled Bit Count. Specifies the number of bits in the last received data character of a message and should be examined by the processor when REOM = 1 (RxDA and RxSA asserted). ABC = 0 indicates the message was terminated (by a FLAG or GA) on a character boundary as specified by PCR_{8-10}. Otherwise, ABC = number of bits in the last data character. ABC is cleared when RDSR_H is read, reset operation, or dropping RxE. The residual character is right justified in RDSR_L.
15	RERR	BOP/BCP	Receiver Error indicator should be examined by the processor when REOM = 1 in BOP, or when the processor determines the last data character of the message in BCP with CRC or when RxSA is set in BCP with VRC. CRC-CCITT preset to 1's/0's as specified by PCSAR_{8-10}: RERR = 1 indicates FCS error (CRC $\neq$ F0B8/$\neq$ 0) RERR = 0 indicates FCS received correctly (CRC = F0B8 / = 0) CRC-16 preset to 0's on 8-bit data characters specified by PSCAR_{8-10}: RERR = 1 indicates CRC-16 received correctly (CRC = 0). RERR = 0 indicates CRC-16 error (CRC $\neq$ 0) VRC specified by PCSAR_{8-10}: RERR = 1 indicates VRC error RERR = 0 indicates VRC is correct

Table 8 RECEIVER DATA/STATUS REGISTER (RDSR)-(Read Only)

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
T_A Operating ambient temperature[2]	0 to +70	°C
T_{stg} Storage temperature	−65 to +150	°C
Input or output voltages with respect to GND[3]	−0.3 to +15	V
V_{CC} With respect to GND	−0.3 to +7	V

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		100	
Ceramic		50	
Cerdip		60	

NOTES

1. 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 at any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (IQ ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maximum.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at 0.8 V or 2.0 V. Input voltage levels for testing are 0.4 V and 2.4 V.
6. Output load C_L = 100 pF.
7. m = TxC low and applies to writing to $TDSR_H$ only.

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = +5V ±5%[4,5]

PARAMETER	TEST CONDITIONS	LIMITS			UNIT
		Min	Typ	Max	
Input voltage					V
V_{IL} Low				0.8	
V_{IH} High		2.0			
Output voltage					V
V_{OL} Low	I_{OL} = 1.6mA			0.4	
V_{OH} High	I_{OH} = −100µA	2.4			
I_{CC} Power supply current	V_{CC} = 5.25V, T_A = 0°C			150	mA
Leakage current					µA
I_{IL} Input	V_{IN} = 0 to 5.25V			10	
I_{OL} Output	V_{OUT} = 0 to 5.25V			10	
Capacitance					pF
C_{IN} Input	V_{IN} = 0V, f = 1MHz			20	
C_{OUT} Output	V_{OUT} = 0V, f = 1MHz			20	

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 5%[4,5,6]

PARAMETER	MC2652/MC68652			MC2652 _ 2/MC68652 _ 2			UNIT
	Min	Typ	Max	Min	Typ	Max	
Setup and hold time							ns
t_{ACS} Address/control setup	50			50			
t_{ACH} Address/control hold	0			0			
t_{DS} Data bus setup (write)	50			50			
t_{DH} Data bus hold (write)	0			0			
t_{RXS} Receiver serial data setup	150			150			
t_{RxH} Receive serial data hold	150			150			
Pulse width							ns
t_{RES} RESET	250			250			
t_{DBEN} DBEN	250		m[7]	200		m[7]	
Delay time							ns
t_{DD} Data bus (read)			200			170	
t_{TxD} Transmit serial data			325			250	
t_{DBEND} DBEN to DBEN delay	200			200			
t_{DF} Data bus float time (read)			150			150	ns
f Clock (RxC, TxC) frequency			1.0			2.0	MHz
t_{CLK1} Clock high (MM = 0)	340			165			ns
t_{CLK1} Clock high (MM = 1)	490			240			
t_{CLK0} Clock low	490			240			

POWER CONSIDERATIONS

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

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

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 $P_{PORT} < P_{INT}$ 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:

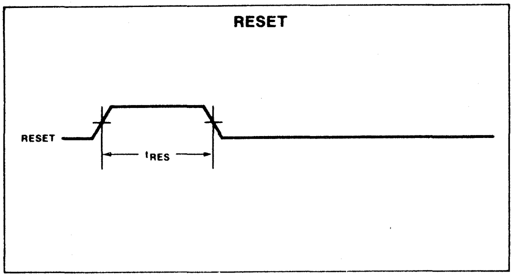
$$P_D = K + (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

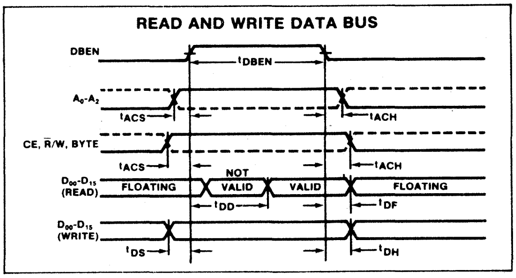
$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \qquad (3)$$

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.

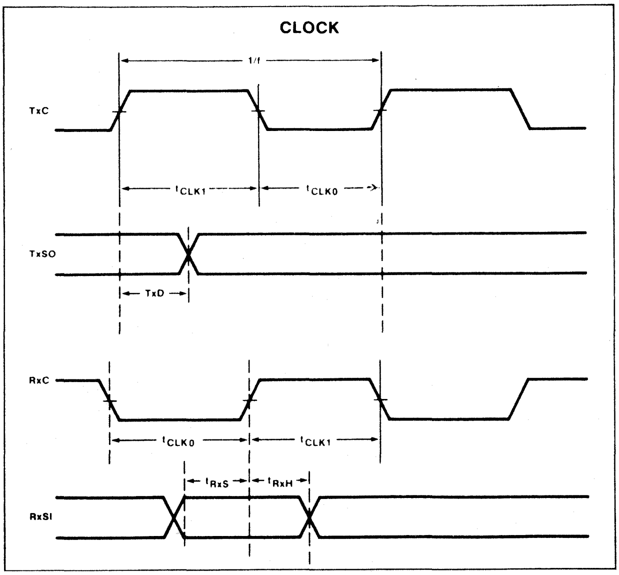
TIMING DIAGRAMS



RESET



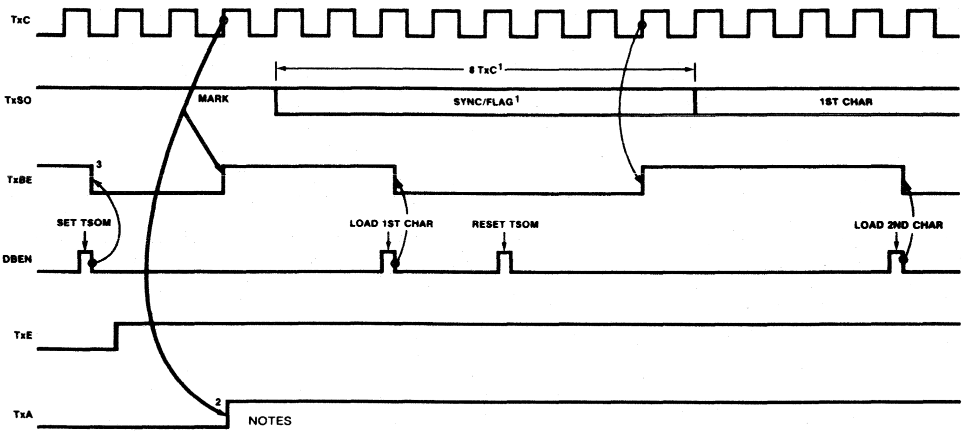
READ AND WRITE DATA BUS



CLOCK

TIMING DIAGRAMS (Cont'd)

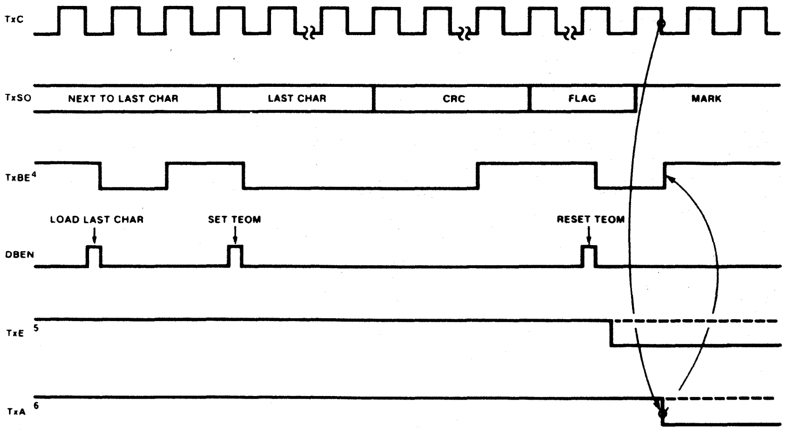
TRANSMIT—START OF MESSAGE



NOTES

1. SYNC may be 5 to 8 bits and will contain parity bit as specified.
2. TxA goes high relative to TxC rising edge after TSOM has been set and TxE has been raised.
3. TxBE goes low relative to DBEN falling edge on the first write transfer into TDSR. It is reasserted 1 TxC time before the first bit of the transmitted SYNC/FLAG. TxBE then goes low relative to DBEN falling edge when writing into $TDSR_H$ and/or $TDSR_L$. It is reasserted on the rising edge of the TxC that corresponds to the transmission of the last bit of each character, except in BOP mode when the CRC is to be sent as the next character (see Transmit Timing—End of Message).

TRANSMIT—END OF BOP MESSAGE

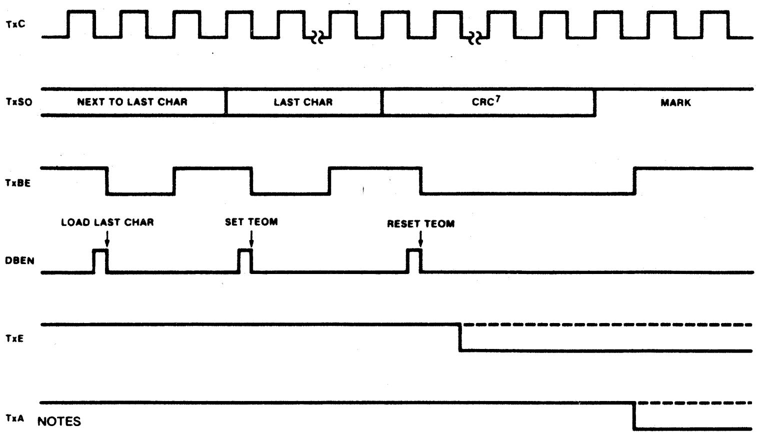


NOTES

4. TxBE goes low relative to the falling edge of DBEN corresponding to loading $TDSR_{H/L}$. It goes high one TxC before character transmission begins and also when TxA has been dropped.
5. TxE can be dropped before resetting TEOM if TxBE (corresponding to the closing FLAG) is high. Alternatively TxE can remain high and a new message initiated.
6. TxA goes low after TxE has been dropped and 1 1/2 TxC's after the last bit of the closing FLAG has been transmitted.

TIMING DIAGRAMS (Cont'd)

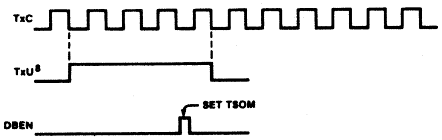
TRANSMIT TIMING—END OF BCP MESSAGE



NOTES

7. When 2652 generated CRC is not required, TEOM should only be set if SYNCs are to follow the message block. In that case, TxE should be dropped in response to TxBE (which corresponds to the start of transmission of the last character). When CRC is required, TxE must be dropped before CRC transmission is complete. Otherwise, the contents of TxDB will be shifted out on TxSO. This facilitates transmission of contiguous messages.

TRANSMIT UNDERRUN

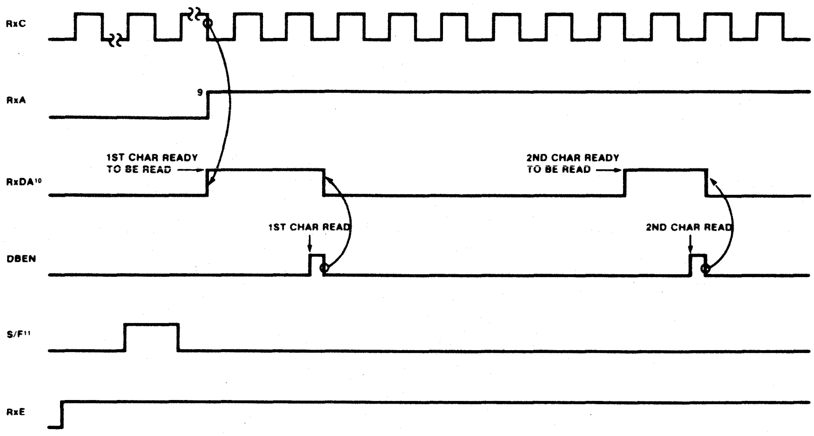


NOTES

8. TxU goes active relative to TxC falling edge if TxBE has not been serviced after n-1/2 TxC times (where n = transmit character length). TxU is reset on the TxC falling edge following assertion of the TSOM command.
9. An underrun will occur at the next character boundary if TEOM is reset and the transmitter remains enabled, unless the TSOM command is asserted or a character is loaded into the TxDB.

TIMING DIAGRAMS (Cont'd)

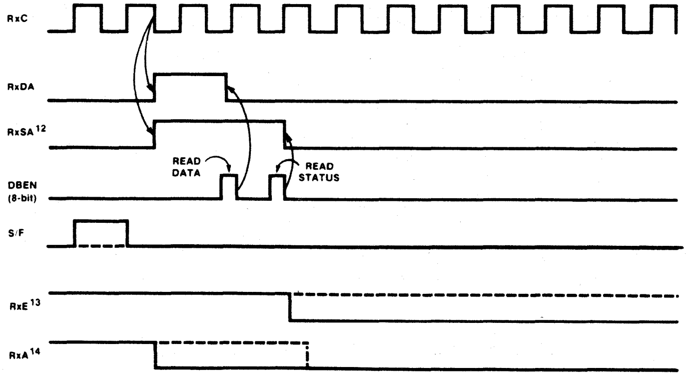
RECEIVE—START OF MESSAGE



NOTES

9. RxA goes high relative to falling edge of RxC when RxE is high and:
   - a. A data character following two SYNC's is in RxDB (BCP mode)
   - b. Character following FLAG is in RxDB (BOP primary station mode)
   - c. Character following FLAG is in RxDB and character matches the secondary station address or All Parties Address (BOP secondary station mode).
10. RxDA goes high on RxC falling edge when a character in RxDB is ready to be read. It comes up before RxSA and goes low on the falling edge of DBEN when RxDB is read.
11. S/F goes high relative to rising edge of RxC anytime a SYNC (BCP) or FLAG (BOP) is detected.

RECEIVE END OF MESSAGE

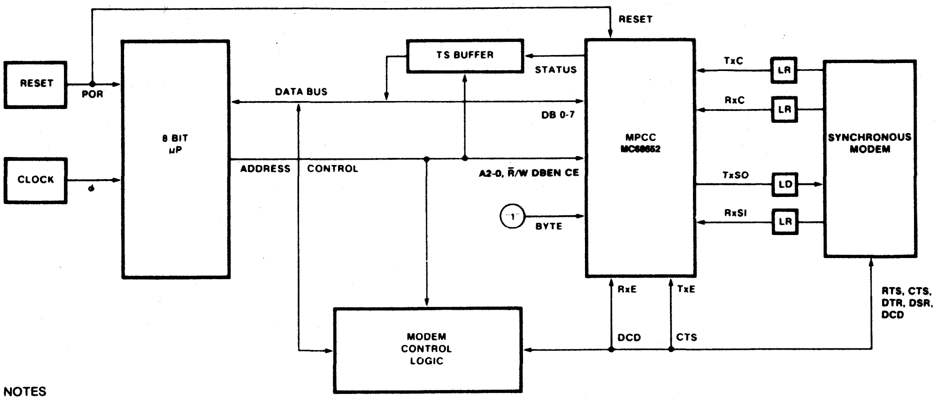


NOTES

12. At the end of a BOP message, RxSA goes high when FLAG detection (S/F = 1) forces REOM to be set. Processor should read the last data character (RDSR$_{L}$) and status (RDSR$_{H}$) which resets RxDA and RxSA respectively. For BCP end of message, RxSA may not be set and S/F = 0. The processor should read the last data character and status.
13. RxE must be dropped for BCP with non-contiguous messages. It may be left on at the end of a BOP message (see BOP Receive Operation).
14. RxA is reset relative to the falling edge of RxC after the closing FLAG of a BOP message (REOM = 1 and RxSA active;) or when RxE is dropped.

TYPICAL APPLICATIONS

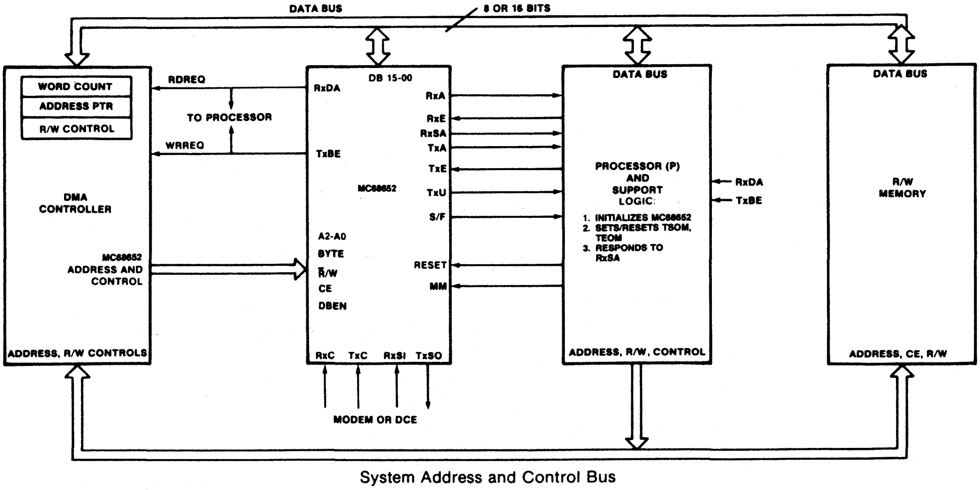
2652 MPCC MICROPROCESSOR INTERFACE



NOTES

1. Possible μP interrupt requests are:
   RxDA, RxSA, TxBE, TxU.
2. Other MC68652 status signals and possible uses are:
   S/F line: idle indicator, frame delimiter.
   RxA handshake on RxE, line turn around control
   TxA handshake on TxE, line turn around control.
3. Line Drivers/Receivers (LD/LR) convert EIA to TTL voltages and vice-versa
4. RTS should be dropped after the CRC (BCP) or FLAG (BOP) has been transmitted.
   This forces CTS low and TxE low.
5. Corresponding high and low order bits of DB must be OR tied.

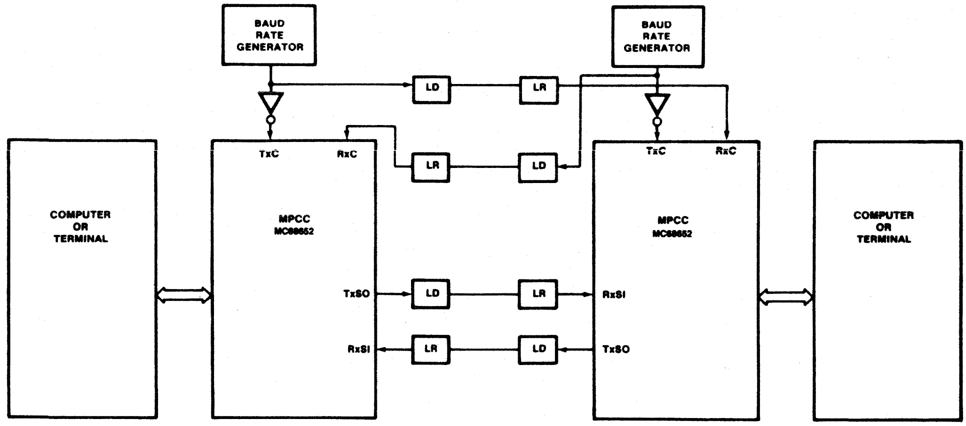
DMA/PROCESSOR INTERFACE



For non-DMA operation, TxBE and RxDA are sent to the processor which then loads or reads data characters as required.

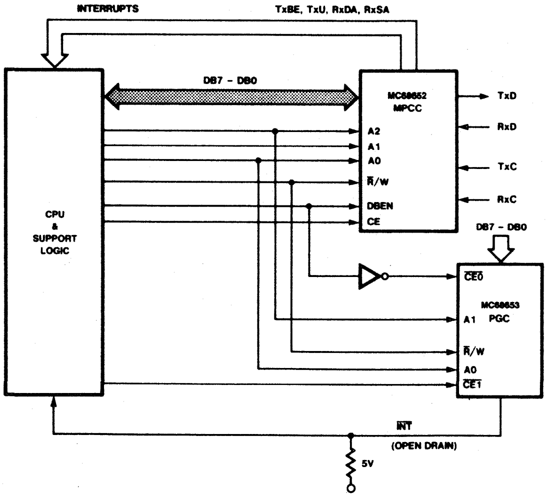
TYPICAL APPLICATIONS (Cont'd)

CHANNEL INTERFACE



No Modem—DC Baseband Transmission

MC68652/MC68653 INTERFACE
TYPICAL PROTOCOLS: BISYNC, DDCMP, SDLC, HDLC



MC68000-to-MC68652 INTERFACE CIRCUIT

This circuit will generate the DBEN select signal required by the MC68652 Multiple Protocol Communications Controller (MPCC) and the DTACK signal required by the MC68000. This allows the MPCC to interface with the MC68000's asynchronous bus.

The interface circuit is shown in Figure 8. The MPCC has a 16-bit data bus (DO-D15), three register select signals (A0, A1, and A2), a chip enable (CE), and a read/write signal (R/W). The data bus is connected directly to the MPU data bus and the register selects are connected to the A1 and A2 address lines of the processor. A0 of the MPCC is generated from LDS. Together, UDS and LDS determine the width of the transfer on the data bus. If UDS and LDS are both asserted, the BYTE input is low and a word is transferred; otherwise the byte input is high, indicating byte transfers. The MPU's R/W line is inverted and connected to the R/W line of the MPCC. Since DBEN controls all data transfers and operations internal to the MPCC, CE is held asserted and DBEN is used as the chip select.

An access begins when AS is asserted by the MPU, indicating a valid address on the address bus. UDS and/or LDS are then asserted, enabling flip flop U2 to assert DBEN on the next rising edge of the system clock. The one clock cycle delay between SELECT and DBEN allows the MPCC setup time to be met. DTACK is then asserted on the next rising edge, forcing the processor to insert four wait states. This guarantees that the MPCC access time is met. At the end of the cycle, AS and the data strobes are negated, causing DBEN to be negated. The timing for a write cycle is shown in Figure. 9.

Figure 8. MC68000/MC68652 Interface Circuit

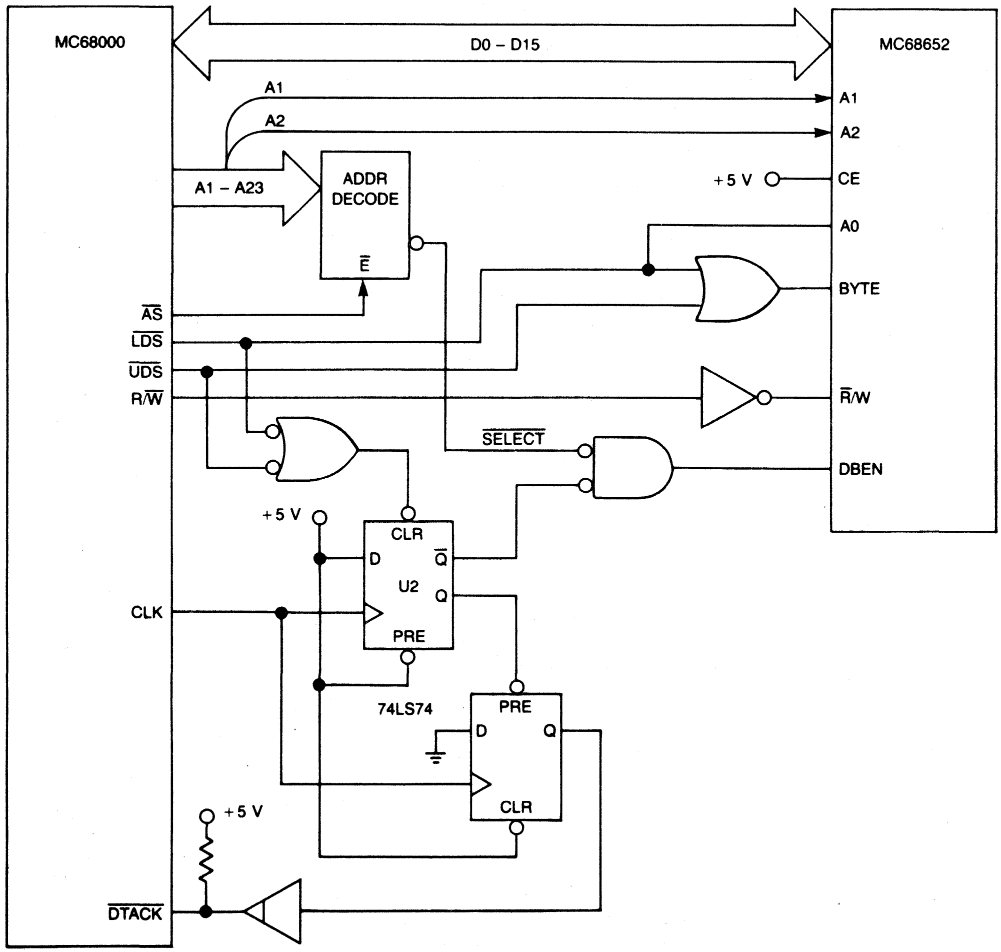
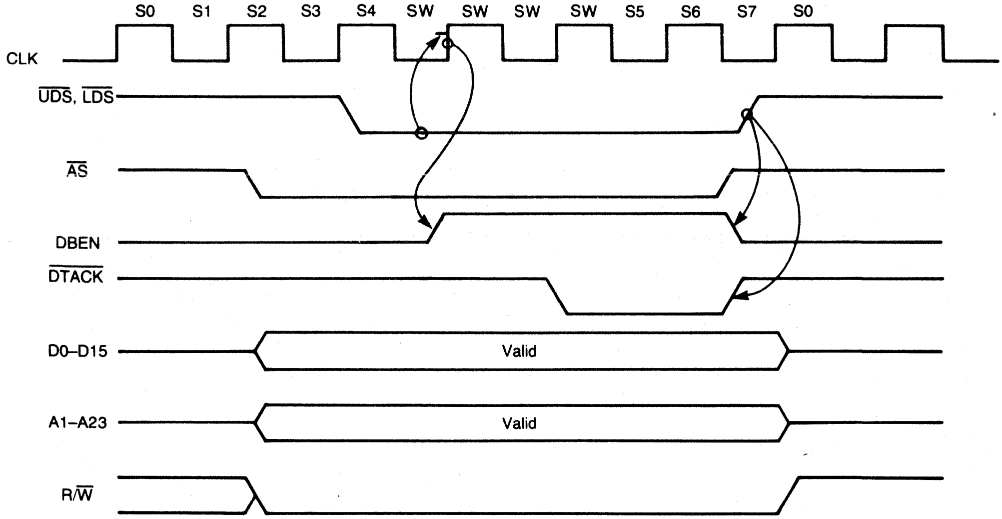


Figure 9. MPCC Write Cycle Timing



MOTOROLA

MC2653/MC68653

Advance Information

POLYNOMIAL GENERATOR CHECKER

The MC2653/MC68653 Polynomial Generator Checker (PGC) is a polynomial generator checker/character comparator circuit that complements a receiver/transmitter (R/T or USART/USRT/UART) in the support of character oriented data-link controls. Table 1 defines many of the more commonly used PGC terms and abbreviations.

Parallel data characters transferred between the MPU and R/T are monitored by the PGC which performs block check character (BCC) and parity (VRC) generation/checking, single character detection, and two character sequence detection. Since the PGC operates on parallel characters, the data transmission format may be serial (synchronous or asynchronous) or parallel. See Figure 1 for a typical system configuration.

- Parallel Block Check Character Accumulation/Checking: CRC-16, CRC-12, LRC-8
- BISYNC Normal and Transparent Modes
- Automatic- or Single-Character Accumulation Modes
- Character Detection up to 128 Characters
- Two Character Sequence Detection; Examples: DLE-STX, ACK 0, ACK 1, WACK, RVI, DISC, WBT
- 6-, 7-, or 8-Bit Characters
- VRC Generation/Checking on Data Bus
- Four Maskable Interrupt Conditions
- Internal Power-on Reset
- Maximum Character Accumulation Rate of 500 kHz (4 Mbps)
- Directly Compatible with Motorola's MC68652, MC68661, and Signetics 2651
- No System Clock Required
- TTL Compatible Inputs and Outputs
- Single 5 V Supply

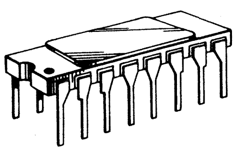
APPLICATIONS

- Character Oriented Data Link Control
  - Dedicated to One USART/USRT
  - Multiplexed Among Several USART/USRTs
- Automated BISYNC with MC68661 (Minimal Software Intervention)
- BCC and VRC Generation/Detection on a Block of Memory or Peripheral Data
- Programmable Character Array Comparator

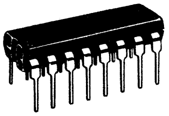
MOS

(N-CHANNEL, SILICON-GATE)

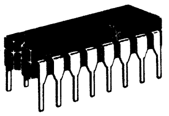
POLYNOMIAL GENERATOR CHECKER



L SUFFIX
CERAMIC PACKAGE
CASE 690

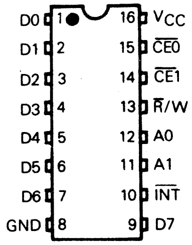


P SUFFIX
PLASTIC PACKAGE
CASE 648



S SUFFIX
CERDIP PACKAGE
CASE 620

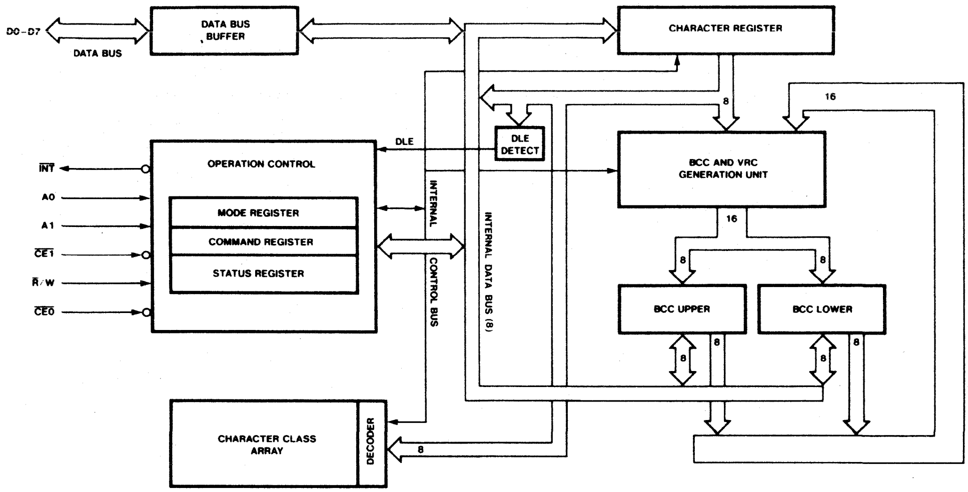
PIN ASSIGNMENT



ADI-871

BLOCK DIAGRAM

The PGC consists of six major sections. These are the operation control, character class array, DLE ROM, character register, BCC and parity generators, and BCC registers. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the CPU data bus via a data bus buffer.



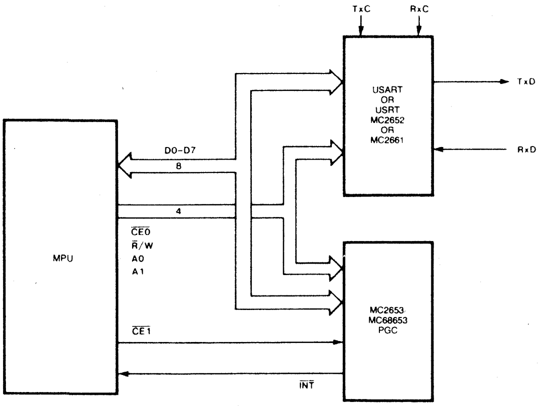
PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
VCC	16	I	**+5 V:** Power supply
GND	8	I	**Ground**
A1-A0	11,12	I	**Address Lines:** Used to select internal PGC registers or character class array
$\overline{\text{R}}$/W	13	I	**Read/Write:** Read command when low, write command when high
$\overline{\text{CE0}}$	15	I	**Chip Enable:** Connected to chip enable input of a receiver/transmitter (R/T) circuit. It is used to strobe data being transferred between the MPU and the $\overline{\text{R}}$/T into the PGC character register.
$\overline{\text{CE1}}$	14	I	**Chip Enable:** Used in conjunction with the $\overline{\text{R}}$/W signal to enable the transfer of data between the PGC and the MPU or DMA controller and to initialize the PGC registers.
D7-D0	9,7-1	I/O	**Data Bus:** 8-bit three-state bidirectional bus used to transfer data to or from the PGC via $\overline{\text{CE0}}$ or $\overline{\text{CE1}}$. All data, mode words, command words, and status information are transferred on this bus. D0 is the least significant bit; D7 is the most significant bit.
$\overline{\text{INT}}$	10	O	**Interrupt:** Open drain active low interrupt output that signals the MPU that one or more maskable conditions are true: BCC error, VRC error, BTC/SC detect, SSC detect. The true conditions can be determinnd by reading the status register which in turn deactivates $\overline{\text{INT}}$. A power on, clear BCC, or master reset command causes $\overline{\text{INT}}$ to be inactive.

Table 1. GLOSSARY

TERM/ABBREVIATION	DEFINITION
BCC	Block check character
BTC	Block terminating character
SC	Search character
SSC	Second search character (preceded by DLE)
CRC-16	$X^{16} + X^{15} + X^2 + 1$ divisor, dividend pre-cleared
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$ divisor, dividend pre-cleared
LRC-8	Horizontal parity on least significant 7 bits; vertical parity on most significant bit
VRC	Vertical redundancy check (character parity)
R/T	Receiver/transmitter circuit. Also known as USART/USRT/UART/EPCI/MPCC
BISYNC	IBM binary synchronous communications (BSC), ANSI X3.28, ISO 1745
MSB	Most significant bit
LSB	Least significant bit
Rx	Receive
Tx	Transmit

TYPICAL SYSTEM CONFIGURATION



NOTES
1. Open drain $\overline{\text{INT}}$ may be OR tied with 2651 PCI, MC68661 EPCI, or other open drain interrupt outputs.
2. No external circuitry necessary if 2651 or MC68661 is the USART.

Figure 1

Operation Control Unit

This functional block stores configuration and operation instructions from the MPU and generates appropriate signals to control the device operation. It also contains read and write circuits to permit communications between the MPU and the PGC registers via the data bus. The mode, command, and status registers are in this logic block.

Character Register

Characters to be considered for BCC generation, parity generation and checking, or character comparisons are loaded into this register by either $\overline{\text{CE0}}$ or $\overline{\text{CE1}}$. This register serves as an input to the BCC and VRC generator, where the accumulation and parity generation takes place. The character register also serves as the input for character class array and DLE comparisons.

Character Class Array

This 128 x 2 array holds the character class associated with each of 128 possible 7-bit characters. The array is zero after a master reset. When the character class array is loaded (see PGC Addressing), the character on the data bus is placed in the class specified by the contents of command register bits CR2 and CR3. The PGC uses these two command bits to represent four different character classes. These are:

1. Normal class (included in the accumulation)
2. SYN character/BISYNC not included class
3. Block terminating character/search class
4. Second search character class (preceded by DLE)

These encoded character classes are used by the PGC:

1. To control the BCC accumulation of associated characters in BISYNC modes only. BCC accumulation in automatic or single accumulation modes is carried out independent of the character classes.
2. To detect characters and two character sequences in all modes of accumulation and to set the control character detect bits in the status register.

It should be noted that any number of characters (up to 128 for CRC-16 or LRC-8; up to 64 for CRC-12) can be put into any one class.

If VRC is specified along with CRC-16 then the least significant 7 bits of the character are used for character array comparison. If VRC is not enabled, but CRC-16 is, the MSB of the character then determines whether a character comparison is to take place. If the MSB is 0, the comparison takes place; if the MSB is 1, the comparison does not take place and the character is processed as though it were in the normal class. This enables the PGC to detect all communication control characters and DLE-SSC sequences.

Only the first 64 locations of the array are accessed if CRC-12 is selected. The user should right justify each six bit character (D0-D5) to be written into the character class array. Bit 6 must be zero.

If VRC is enabled, the generated parity becomes the most significant bit of the character to be compared. VRC is not allowed in BISYNC transparent mode.

The method in which the character register contents is compared against the character class array depends on the BCC polynomial chosen. Figure 2 illustrates the comparison process.

DLE Read Only Memory

The DLE characters are stored internally and are selected by the error polynomial as follows:

CRC-12: 01 1111
LCR-8 or CRC-16:
No VRC or odd VRC: 0001 0000
Even VRC: 1001 0000

BCC and Parity Generator

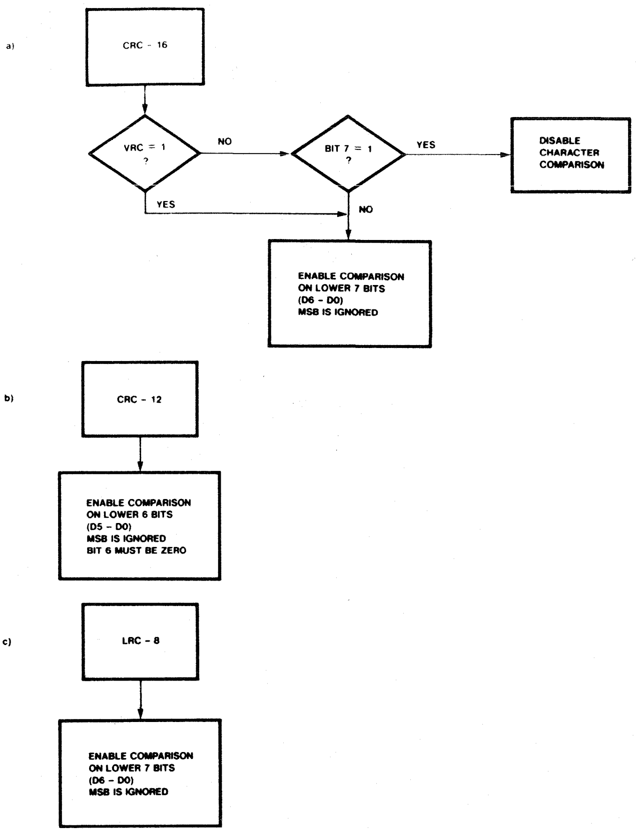
This functional block performs all the necessary computation to generate and update the BCC accumulation on a character by character basis. It contains the three generator polynomials (CRC-16, CRC-12, and LRC-8) that can be selected to compute the BCC. This block also checks and generates odd or even parity for 7-bit (ASCII) characters.

BCC Registers

This block consists of two 8-bit registers (BCC upper and BCC lower) which contain the high and low order bytes of the BCC accumulation. The result of the accumulation from the BCC and parity generator is stored in these registers. A recirculating register address pointer is initialized by a power on, master reset, or clear BCC command. The pointer alternately selects BCC upper and lower on successive BCC register accesses for CRC-16 or CRC-12. For LRC-8, BCC upper is always selected.

BCC upper and lower are cleared by a clear BCC or master reset command. The highest term of the BCC polynomial is always represented by bit 0 of BCC upper; the lowest term is always represented by bit 7 of BCC lower (see figure 3, Orientation of BCC Polynomials.)

CHARACTER CLASS ARRAY COMPARISON OPERATION



NOTE
Transparent mode always disables the SSC comparison.

Figure 2

The length of the block check character depends on the error checking polynomial that is selected. If LRC-8 is chosen, the BCC result is stored entirely in BCC upper. The BCC lower remains unchanged from previous setting. Both BCC registers are used when CRC-16 is specified. When CRC-12 is selected, the block check character is 12 bits long. The six least significant bits of the BCC are stored in the least significant bits of the BCC lower. The remaining upper six bits of the BCC are stored in least significant bits of BCC upper. The two most significant bits in each BCC register are filled with zero.

The BCC register(s) are read by the CPU after the last data character is transmitted. They can then be sent to the R/T to complete a transmitted block of data. These registers are read and loaded when one PGC is time-shared by several R/Ts. Refer to Applications Information - Multiplexed PGC.

PGC Addressing

All internal registers and the character class array are selected by the unique address codes shown in table 2.

ORIENTATION OF BCC POLYNOMIALS

$$\frac{M(X)}{G(X)} = Q(X) + \frac{R(X)}{G(X)}, \quad \text{WHERE } R(X) = A_nX^n + A_{n-1}X^{n-1} \cdots + A_0X^0$$

M (X) : BINARY POLYNOMIAL (DATA STREAM)
G (X) : FIXED DIVISOR TO GENERATE BCC
Q (X) : QUOTIENT AFTER BCC GENERATION
R (X) : REMAINDER AFTER BCC GENERATION

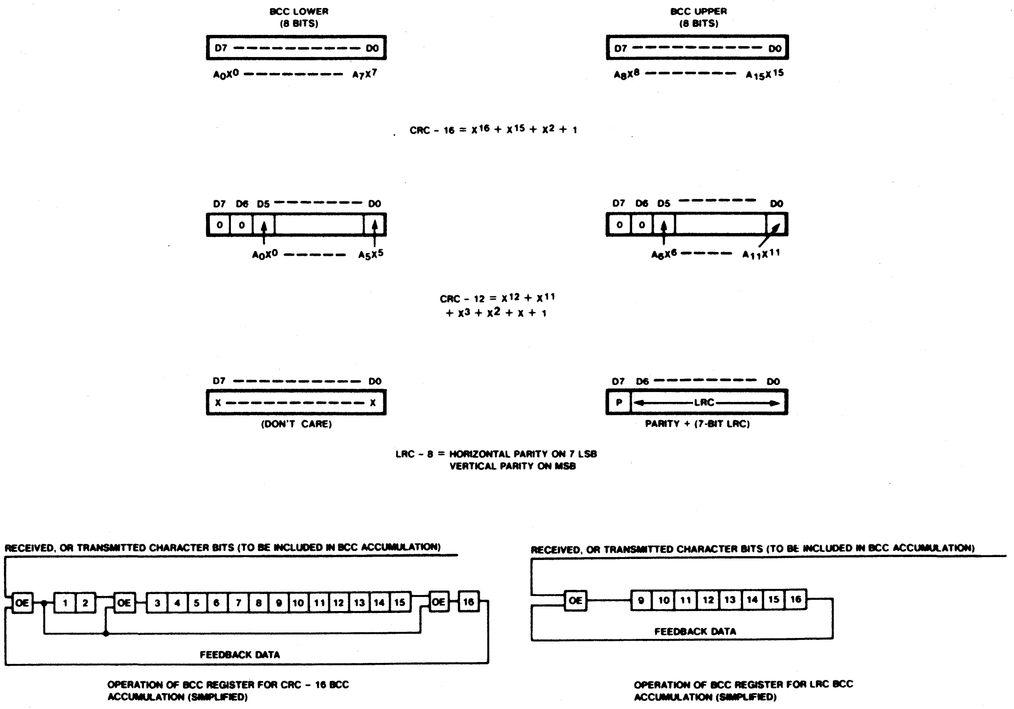


Figure 3

Table 2. ADDRESS CODES

$\overline{CE0}$	$\overline{CE1}$	A1	A0	$\overline{R}/W$	FUNCTION
0	0	X	X	X	Operation not guaranteed
0	1	0	0	0	If MR2 = 0 load data bus into character register If MR2 = 1 PGC not selected[1]
0	1	0	0	1	If MR2 = 1 load data bus into character register If MR2 = 0 PGC not selected[1]
0	1	0	1	X	PGC not selected[1]
0	1	1	0	X	PGC not selected[1]
0	1	1	1	X	PGC not selected[1]
1	0	0	0	0	Read character register
1	0	0	0	1	Load data bus into character register if MR1,0 ≠ 00[2]; write character class array using CR3, CR2 class code if MR1,0 = 00[3,4]
1	0	0	1	0	Read Status register
1	0	0	1	1	Write command register
1	0	1	0	0	Read mode register
1	0	1	0	1	Write mode register
1	0	1	1	0	Read BCC upper/lower[5]
1	0	1	1	1	Write BCC upper/lower[5]
1	1	X	X	X	PGC not selected[1]

NOTES

1. Data bus is 3-state
2. Character will not be accumulated unless MR3 = 1.
3. Character will not be accumulated even if MR3 = 1.
4. The mode bits MR1 and MR0 are cleared to 00 by power-on-reset, master reset, or by loading the mode register bits MR1 and MR2.
5. Recirculating internal pointer selects BCC upper on first access, BCC lower on next access for all BCCs except for LRC-8; in case of LRC-8, the pointer only selects BCC upper.

INTERFACE SIGNALS AND TIMING

PGC data transfers are controlled by A1, A0, and $\overline{R}$/W which must be stable prior to the active low going chip enable pulse. $\overline{CE0}$ is used for PGC monitoring of data transfers between a MPU/DMA controller and a R/T; $\overline{CE1}$ is used for direct MPU-to-PGC transfers. MR3 must be set prior to loading the character register in order to accumulate or compare characters via $\overline{CE1}$. The active low (leading) edge of chip enable initiates a PGC read/write cycle; the rising (trailing) edge ends the cycle and also serves as a write strobe.

When loading the character, mode, or command register, the data bus is strobed into the selected register on the trailing (rising) edge of the appropriate $\overline{CE}$. When writing into the character class array, the data on the bus (the special character) is placed in the class specified by command register bits CR3 and CR2.

Characters are transferred into the character register when $\overline{CE0}$ is active (low) depending on the state of MR2 and the $\overline{R}$/W input. Characters from the R/T are loaded into the character register when in receive mode (MR2 = 0 and $\overline{R}$/W = 0) while CPU/DMA characters are loaded into the character register when in transmit mode (MR2 = 1 and $\overline{R}$/W = 1). The time between consecutive chip enables is given by t_{CEC} or t_{CED}.

The open drain active low interrupt signal ($\overline{INT}$) goes active whenever one or more of four maskable status conditions (SR0-SR3) are true (= 1). A status read deactivates $\overline{INT}$.

The same techniques used in interfacing the 2651 PC1 to 8-bit microprocessors can be used to interface the PGC. Note that when addressing the R/T's holding registers, the PGC pins must have A1,A0 = 00 and that the address and $\overline{R}$/W signals must be stable (set up) prior to the active low chip enable. When using the MC2661/MC68661 as the R/T, the PGC's A1, A0, $\overline{R}$/W, and $\overline{CE0}$ are directly connected to comparable 2651 or MC68661 signals. Schematics of a MC68653 monitoring data transfers to/from the MC68661 or MC68652 are shown in figures 4 and 5.

An alternate interfacing technique is to treat the PGC as an independent peripheral device. This necessitates a write character register instruction after the CPU reads or writes a character to or from the R/T.

MC68661/MC68653 INTERFACE

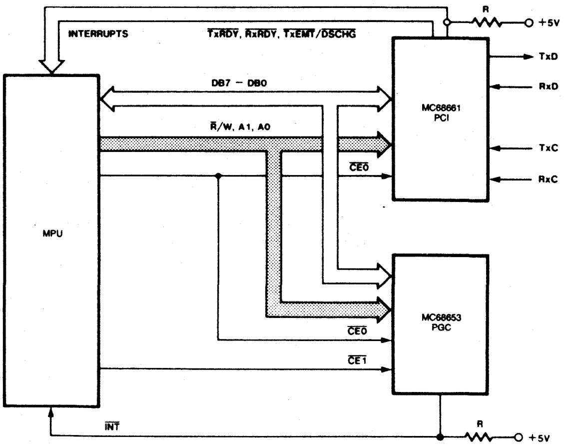


Figure 4

MC68652/MC68653 INTERFACE
TYPICAL PROTOCOLS: BISYNC, DDCMP, SDLC, HDLC

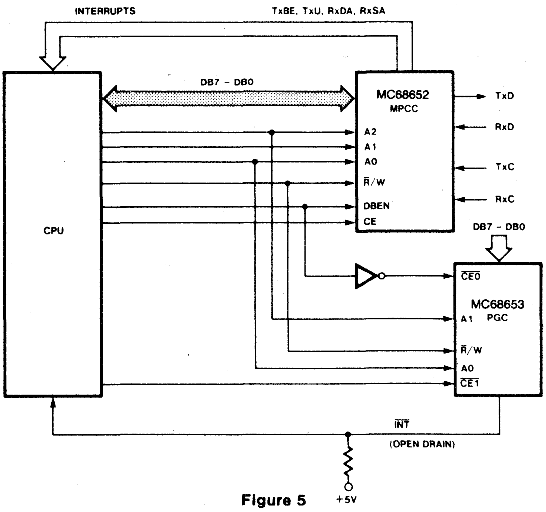


Figure 5

PGC PROGRAMMING

The PGC operational mode must be initially programmed by the CPU (see figure 6). The mode register, command register and character class array should be written into, after a power-on-reset or a master reset command. The character class array should be programmed only for the classes pertinent to the application. After a master reset, the character class array is zero which places all characters in the normal class (included in the BCC accumulation).

OPERATION

The PGC should be initially configured by the CPU (via $\overline{CE1}$) prior to systems operation. This is done by loading the mode register, command register and character class array (see PGC PROGRAMMING). Characters may then be loaded into the character register for BCC accumulation, VRC generation/checking, BTC/SC and DLE-SSC comparisons. See table 3 for a summary of BCC accumulation modes.

BCC accumulation depends on the mode selected.

BISYNC Normal

In BISYNC normal mode, all characters loaded into the character register are accumulated except those in the SYN/BISYNC not included class. During receive (MR2 = 0), a BTC/SC match will cause the BCC accumulation to stop after the next one (LRC-8) or two (CRC-12 or CRC-16) characters have been accumulated. At that time, if the BCC accumulation does not equal zero, the BCC error bit (SR0) will be set and $\overline{INT}$ will go active if the corresponding mask bit (CR4) is enabled (= 1). In transmit (MR2 = 1), the BCC accumulation is automatically stopped once the BTC/SC character has been accumulated. The CPU must read the BCC upper and BCC lower (CRC-12 or CRC-16) register(s) and transmit them to the R/T.

Note that the received BCCs are not subject to VRC if CRC-16 is selected. If LRC-8 is selected, the received BCC is subject to VRC. An incorrect result will set the VRC error bit (SR1). After its accumulation, the least significant 7 bits of BCC upper are checked and a non-zero result will set the BCC error bit (SR0). BCCs are not checked against the character class array nor are they compared to the DLE ROM.

Second search character (SSC) detection is enabled so that a DLE-STX or two character communication control sequence can be detected.

Table 3. SUMMARY OF BCC ACCUMULATION MODES

ACCUMULATION MODES	START ACCUMULATION	STOP ACCUMULATION	CHARACTERS EXCLUDED FROM ACCUMULATION
BISYNC normal and BISYNC transparent	Clear BCC registers command Mode register is loaded with BISYNC or automatic mode Start accumulation command Load BCC registers	After BTC has been detected and received BCC is accumulated After transmitted BTC has been accumulated Single mode is selected	SYN/BISYNC not included class in normal mode DLE-SYN/not included class and first DLE of a DLE non SYN pair in transparent mode. These characters are not excluded if preceded by an odd number of DLEs
Automatic	Same as above	Single mode selected	None
Single	Start accumulation command	After each character has been accumulated	Up to user who must generate start accumulation command for each character to be included

PGC PROGRAMMING

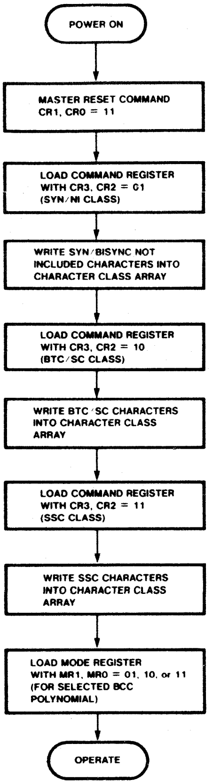
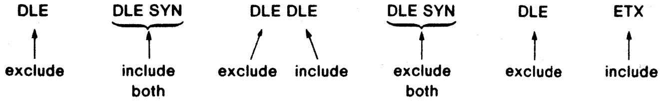


Figure 6

BISYNC Transparent

BISYNC transparent mode should be used for data blocks beginning with DLE-STX if the DLEs are transferred between MPU and R/T (CE0) or MPU and PGC ($\overline{CE1}$), i.e., DLEs are *not* stripped. VRC should be disabled in this mode. Characters excluded from the BCC accumulation are the first DLE of a DLE-non SYN sequence pair and the DLE-SYN sequence if not preceded by an odd number of DLEs. For example, consider the following transparent mode character string:



In receive and transmit modes, the termination of BCC accumulation works exactly as in BISYNC normal, except that the BTC/SC must be immediately preceded by an odd number of DLEs to be identified as a BTC/SC.

Second search character detection is not enabled in BISYNC transparent.

After a BTC/SC class character is detected by the PGC when receiving in either BISYNC mode, the following one or two characters are accumulated (depending on LRC-8 or CRC-12/16, respectively) and the PGC will automatically stop further accumulation. However, *the PGC can continue the accumulation* if a start accumulate command is issued or either BISYNC mode is loaded into the mode register. The start accumulate command should be given to the PGC before loading the character that follows the detected BTC/SC. This procedure enables a special search character to be detected (the BTC/SC detect bit (SR2) will be set and an interrupt generated if CR6 = 1) with the BCC accumulation continuing (see figures 7 and 8).

Automatic Accumulate

All characters loaded into the character register are accumulated, BTC/SC and SSC detection is enabled. The BCC accumulation is not automatically terminated. (The MPU must use single accumulate mode to stop the accumulation). When in receive mode, the BCC error bit (SR0) is set/reset after accumulating each character so that the MPU must examine this bit after the last character is accumulated. SR0 = 0 if the accumulated remainder in the BCC register(s) is zero; otherwise SR0 = 1. Examples of use of automatic accumulate mode usage include an R/T (MC68661 or 2651) in transparent DLE/SYN strip mode and asynchronous/synchronous/parallel DDCMP.

Single Accumulate

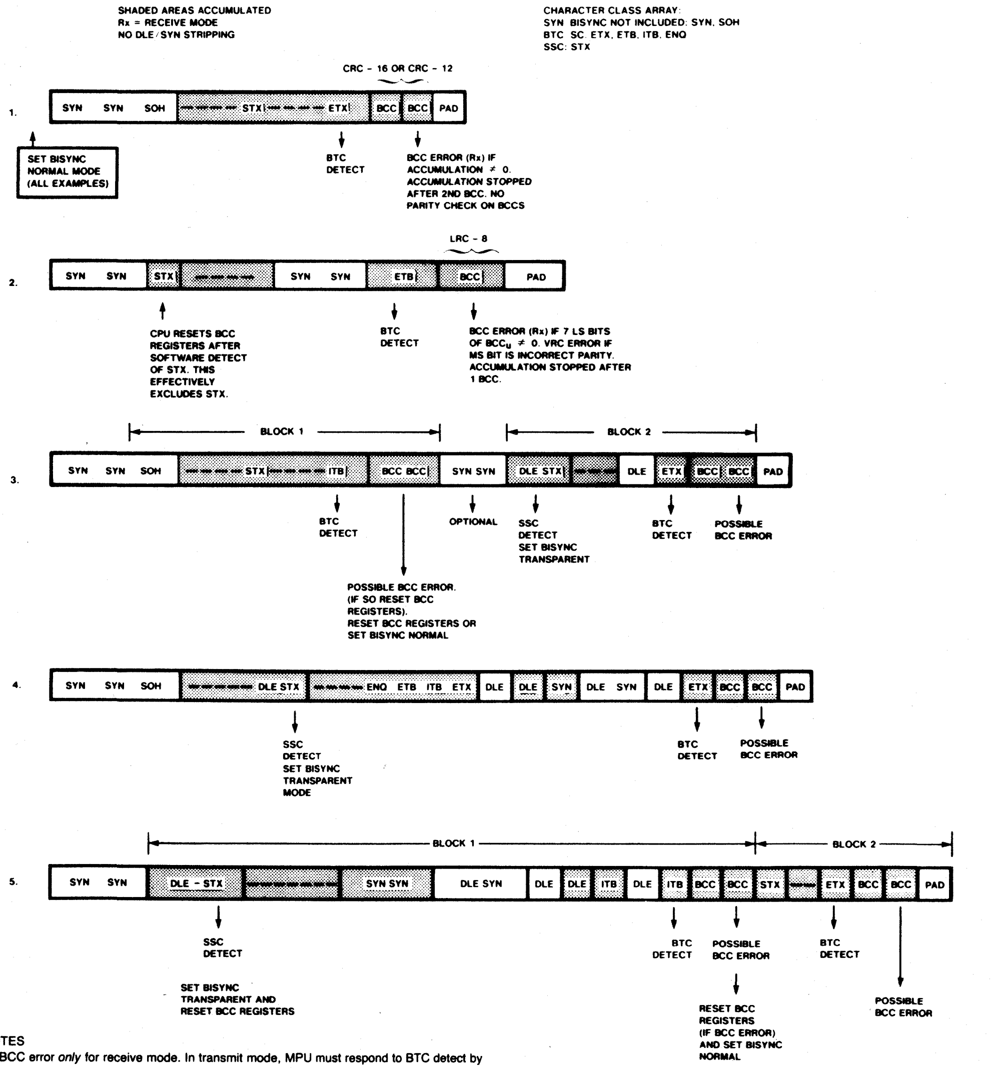
All characters for which a start accumulate command (CR1, CR0 = 01) is given are accumulated and compared against the character class array. If not given, the BCC accumulation is not updated and BTC/SC and SSC detection is disabled. Operation in this mode is otherwise identical to automatic accumulate.

Single accumulate mode can be used to selectively accumulate characters under MPU control or to accumulate characters that were unintentionally excluded in one of the other modes.

Polynomial Selection and DLE Comparison

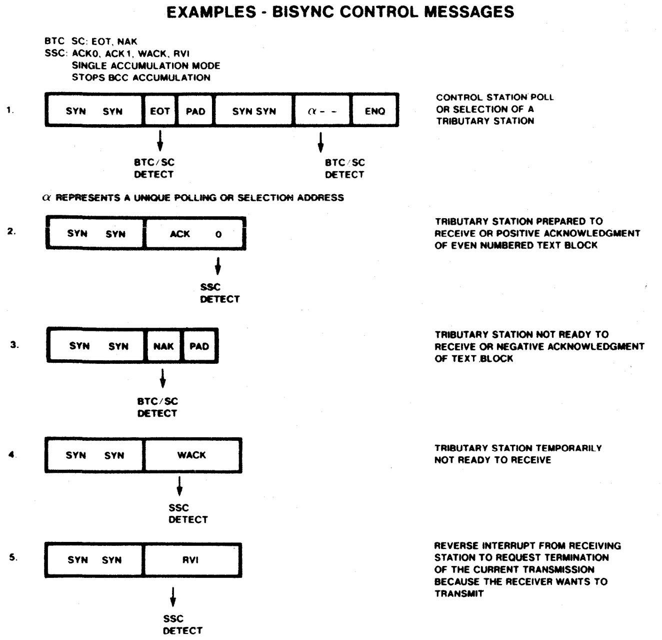
The BCC polynomial may be CRC-16, CRC-12 or LRC-8. The cyclic redundancy check (CRC) is generated by dividing the binary value of a character in the character register by the selected polynomial. The quotient is discarded and the remainder is used as the BCC (two 6-bit characters for CRC-12, two 8-bit characters for CRC-16). CRC-16 uses all 8 bits of each BCC register. CRC-12 uses the least significant 6 bits of the BCC registers. The two most significant bits of the BCC registers are cleared to zero whenever CRC-12 is selected (see figure 3).

EXAMPLES - BISYNC TEXT MESSAGE BCC ACCUMULATION



NOTES
1. BCC error *only* for receive mode. In transmit mode, MPU must respond to BTC detect by reading the BCC register(s) and sending them to the R/T. The accumulation is stopped after the BTC is accumulated.
2. ENQ (DLE-ENQ) in a text message should be treated as an abort.
3. Opening SYNs may be stripped by the R/T.
4. The single accumulate mode and command can be used to accumulate a character that inadvertently was excluded. (For example, the DLE of a DLE-STX if the PGC was in transparent mode and there was not a line turnaround prior to the DLE). The single accumulation should be done using $\overline{\text{CE1}}$ after the BCC(s) have been accumulated.

Figure 7



NOTES

1. BCC accumulation should be ignored for control messages. This can be effected by single accumulate mode without single accumulate commands.
2. Characters programmed as SSCs should be the binary equivalent of the second character of the DLE-SSC sequence.

Figure 8

When the PGC is in receive mode (MR2 = 0), the received BCC will be accumulated. The result will be zero for an error free message.

CRC-12 is used with 6-bit codes. The internal 6-bit transcode DLE character hex 1F is selected by CRC-12. VRC should be disabled (MR4 = 0) for CRC-12 operation. The two most significant bits of the character register are ignored when compared to the internal 6-bit DLE. When the character is checked against the character class array, the MSB is ignored and the next MSB (bit 6) is assumed to be zero. If CRC-12 is specified, the user must write to the character class array with bit 6 cleared.

CRC-16 or LRC-8 implies the use of ASCII or EBCDIC although any 7-bit plus parity or 8-bit no parity code may be used (with DLE = hex 10 or hex 90). The DLE character compare is on an 8-bit basis with the generated parity (if VRC is enabled) as the MSB. When the character is compared against the character class array, the MSB is not used. This may result in a false BTC or SSC detection if there is a VRC error. However, the VRC error bit (SR1) will be set under that condition.

The LRC-8 is generated by the exclusive OR of the 7 least significant bits of the character register and the BCC upper. The most significant bit of the LRC-8 check character is a vertical odd/even parity bit (MR5 = 0/1), which is generated on the least significant bits of that character. The selection of LRC-8 implies VRC is enabled and that only the BCC upper is used for the BCC accumulation. The BCC lower remains unchanged from previous setting.

VRC Generation and Detection

Parity (VRC) is enabled by MR4 and specified as odd or even by MR5. VRC should be disabled when in BISYNC transparent mode and whenever CRC-12 or CRC-16 (EBCDIC) is selected as the BCC polynomial. MR4 = 1 enables VRC generation and detection for *both* receive and transmit operations. Characters loaded into the character register will have VRC generated on the least significant 7 bits with the generated parity bit written into the character register MSB. If the generated parity does not match the MSB of the loaded character, the VRC error bit (SR1) is set and $\overline{INT}$ asserted if the corresponding mask bit was enabled (CR5 = 1). Thus, if 7-bit characters are to be transmitted with VRC, CR5 should be zero and SR1 ignored. 8-bit characters with a VRC bit in the MSB position are parity checked by the PGC in both transmit (to R/T) and receive (from R/T) modes, i.e., the PGC operates as a data bus parity checker.

CHARACTER CLASSES

Normal (Included in the Accumulation)

Any character that belongs to this class is normal data, i.e., the character is not a communication control or other special character. Characters in this class are always accumulated in BISYNC, automatic and single accumulation modes.

SYN Character/BISYNC Not Included

SYN characters are never accumulated in BISYNC normal accumulation mode. In BISYNC transparent accumulation mode, the DLE-SYN character pair is not accumulated, but a SYN not preceded by a DLE is accumulated. (DLE is implied as an odd number of DLEs).

Block Terminating Character (BTC)/Search Character (SC)

BTC/SC characters have two functions in the PGC: termination of BCC accumulation and character detection. In BISYNC transparent mode, a BTC/SC must be preceded by an odd number of DLEs to be recognized.

- **Termination of BCC Accumulation**

  In BISYNC normal and transparent accumulation modes, the PGC will stop the accumulation upon the detection of the BTC/SC character. Examples of BTCs are ETX, ETB, ITB, ENQ.

  In receive mode, the accumulation is stopped after the following one (LRC-8) or two (CRC-12, CRC-16) character(s) have been accumulated. In transmit mode, the accumulation is stopped after the BTC/SC character has been accumulated. The BTC/SC character is always accumulated in all of the accumulation modes.

- **Character Detection**
  BTC/SC characters will be detected in any of the four accumulation modes when that character is being accumulated. The BTC/SC status bit (SR2) is set on detection. Since detection also stops BISYNC BCC accumulation, the BISYNC accumulation must be restarted if the character is not a BTC. This can be effected by loading BISYNC mode into the mode register or generating a start accumulation command.

Second Search Character Class (SSC)

Control functions in character oriented data link control procedures can be represented by a sequence of two characters, the first character being a DLE. Examples include ACK0, ACK1, WACK, RVI, DISC, WBT and the initiation of transparent text (DLE-STX). The PGC will detect such sequences, except in BISYNC transparent mode, when an SSC class character is being accumulated after being immediately preceded by an odd number of DLEs. Under those conditions, the SSC status bit (SR3) will be set.

The SSC character is always accumulated in all of the accumulation modes.

REGISTER BIT DESCRIPTION

The operation of the PGC is determined by programming the mode register and the command register. The status register provides feedback on potential interrupt conditions. Formats of these registers are shown in table 4.

Table 4. PGC REGISTER BIT FORMATS

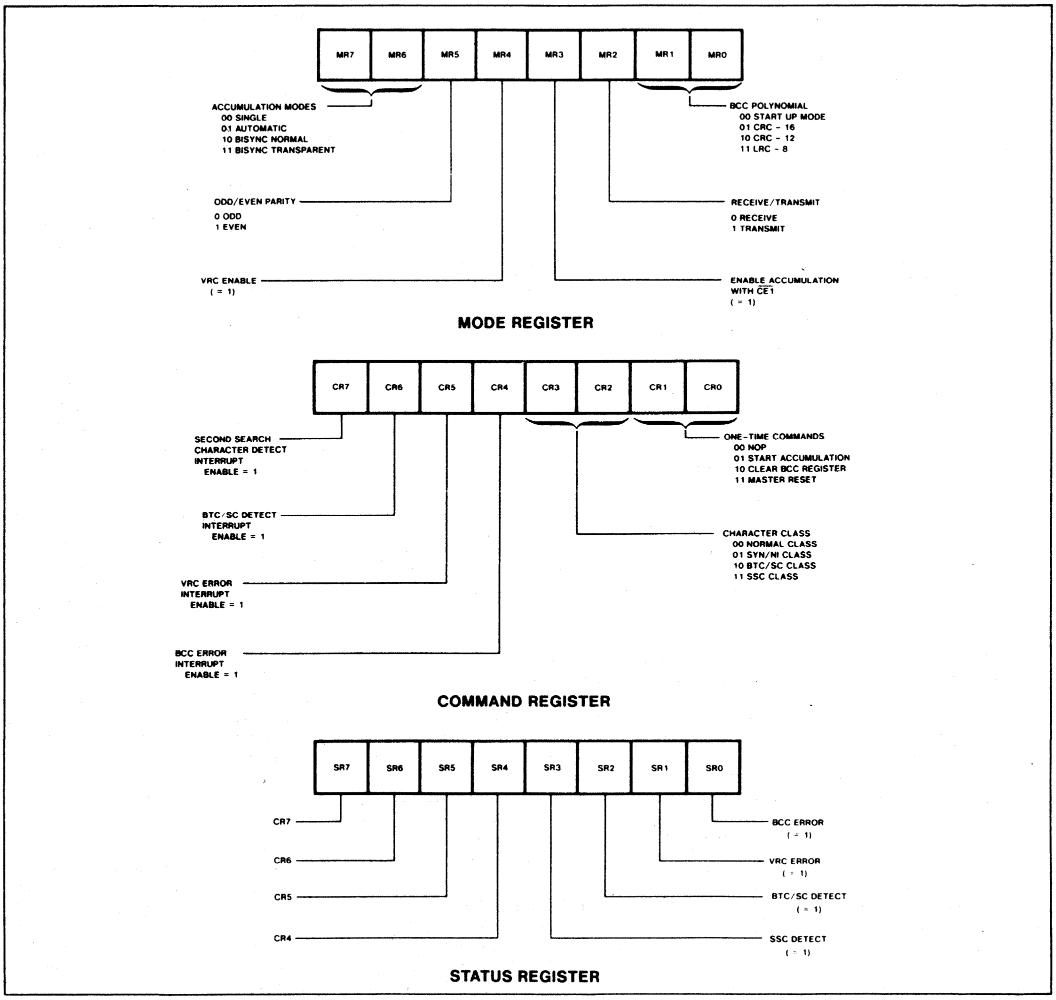


Table 5. BCC ACCUMULATION BY CHARACTER CLASS

CR3	CR2	CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTOMATIC ACCUM	SINGLE ACCUM
0	0	Normal	Yes	Yes	Yes	Yes
0	1	SYN/BISYNC not included	No	Yes, unless preceded by an odd number of DLEs	Yes	Yes
1	0	BTC/SC	Yes	Yes	Yes	Yes
1	1	SSC*	Yes	Yes	Yes	Yes

NOTE

*Preceded by DLE

Mode Register

The mode register defines general PGC operation characteristics. MR1 and MR0 = 00 permit the character class array to be programmed. These bits will be zero after a power on or master reset command. After the character class array is programmed, these bits should be set to 01, 10, or 11 to select the CRC-16, CRC-12 or LRC-8 polynomials.

MR2 (Tx/$\overline{\text{R}}$x) determines whether or not the PGC is to generate (Tx) or generate and check (Rx) the BCC. It is used with $\overline{\text{R}}$/W to determine if the data bus is to be loaded into the character register when $\overline{\text{CE0}}$, $\overline{\text{CE1}}$, A1, A0 = 0100.

If MR2 = 1: 1) the PGC will generate the BCC, but will never set the BCC error bit (SR0). 2) If the $\overline{\text{R}}$/W pin is high when $\overline{\text{CE0}}$, $\overline{\text{CE1}}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{\text{R}}$/W is low under these conditions, the PGC is not selected.

If MR2 = 0: 1) the PGC will accumulate the BCC and set the BCC error bit (SR0) when appropriate. 2) If the $\overline{\text{R}}$/W pin is low when $\overline{\text{CE0}}$, $\overline{\text{CE1}}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{\text{R}}$/W is high under these conditions, the PGC is not selected.

MR3 is a $\overline{\text{CE1}}$ accumulate/compare enable bit. If MR3 = 0, characters loaded into the character register by $\overline{\text{CE1}}$ are *not* accumulated, checked against the character class array, or compared to the DLE ROM. Parity will be generated and checked if VRC is enabled (MR4 = 1). The primary use of MR3 = 0 is to generate parity on a 7-bit character which is to be transmitted to an R/T. The MPU loads the character register with the 7-bit character and reads the 8-bit VRC generated character via $\overline{\text{CE1}}$. This 8-bit character is then transferred to the R/T via $\overline{\text{CE0}}$. Another application of MR3 = 0 is for a CPU to interleave parity checking on memory data ($\overline{\text{CE1}}$) with on line R/T data transfers ($\overline{\text{CE0}}$).

If MR3 = 1, characters loaded into the character register by $\overline{\text{CE1}}$ will be accumulated (according to the BCC accumulation mode selected) and compared against the character class array and DLE ROM. This bit setting should be used when the MPU/DMA controller sends data characters to be accumulated or compared to the PGC and the R/T is inactive (off line). If the R/T were active, then a DLE or BTC loaded into the character register via $\overline{\text{CE0}}$ would cause incorrect accumulation and character comparisons if the next character was loaded via $\overline{\text{CE1}}$.

MR4 is a VRC enable bit. If MR4 = 1, VRC is enabled as odd/even by MR5. VRC is generated on the 7 LS bits of the character and the MS bit is checked against the generated parity. If not equal, SR1 is set. If MR4 = 0, VRC is not enabled. MR4 = 0 is used for BISYNC transparent mode with ASCII code, and for both BISYNC modes for EBCDIC and SBT.

MR5 is an odd/even VRC bit. If MR5 = 1, the total number of 1 bits in the character including the parity bit is odd. If MR5 = 0, the total number of bits is even. This bit is ignored if MR4 = 0.

MR7, MR6 select the BCC accumulation mode. These modes have been previously discussed in the operation section.

Command Register

The command register contains four interrupt enables, a 2-bit character class code used when programming the character class array, and 2 bits that specify three one time commands and a NOP.

CR1, CR0 = 00 is a NOP. This bit setting is used when changing CR7-CR2 without affecting any of the 3 one time commands.

CR1, CR0 = 01 is a start BCC accumulation command. In single accumulation mode, the character accumulated is the character that is in the character register at the time the command is given. The accumulation stops immediately after the character has been accumulated. If the command is given in either of the BISYNC or automatic accumulation modes, it enables the PGC to accumulate the BCC starting with the *next* character loaded into the character register. This is a means of restarting a BISYNC normal accumulation after detection of a BTC/SC that is not a valid BTC (example; CR, LF, TAB). In all accumulation modes, a previously detected DLE will not be cancelled by this command.

CR1, CR0 = 10 is a clear BCC registers command. Both BCC registers are cleared along with the associated internal pointer and SR0-SR3. The pointer points to BCC upper. $\overline{\text{INT}}$ is forced high. This command permits BCC accumulation, starting with the next character loaded into the character register in BISYNC or auto modes. Single accumulate mode requires a start BCC accumulation command.

CR1, CR0 = 11 is a master reset command. All internal registers (except the character register), the internal pointer, and the entire character class array are cleared. $\overline{\text{INT}}$ is negated.

CR3 and CR2 are used for programming the character class array. During a write character class array instruction, the character corresponding to the 7 LS bits of the data bus is placed in the class contained in CR3 and CR2. The encoded character classes control the accumulation of the associated character as shown in table 5.

Detection operates under the conditions shown in table 6.

Table 6. BTC/SC AND SSC DETECTION CONDITIONS

CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTO ACCUM	SINGLE ACCUM
BTC/SC	Yes	Yes*	Yes	Yes †
SSC	Yes*	No	Yes*	Yes*†

NOTES

* Only if immediately preceded by an odd number of DLEs.

† Start accumulate command necessary for detection.

CR7, CR6, CR5, CR4 are interrupt enables that individually enable/disable $\overline{\text{INT}}$ when the corresponding status register condition is true (set). Each bit is set in order to enable $\overline{\text{INT}}$ upon the condition. Each bit is reset to disable $\overline{\text{INT}}$ upon the condition. The state of these bits may be read via the status register (SR7, SR6, SR5, SR4).

The corresponding status bits (SR3, SR2, SR1, SR0) are set independent of the interrupt enables. The bit assignments are:

CR4 - BCC error interrupt enable

CR5 - VRC error interrupt enable

CR6 - BTC/SC detect interrupt enable

CR7 - DLE-SSC detect interrupt enable

Status Register

This register reflects the status of the 4 conditions that are potential interrupt ($\overline{\text{INT}}$) sources and the 4 interrupt enables in the command register. A status register read clears SR0, SR1, SR2, SR3 and deactivates $\overline{\text{INT}}$. These bits are also cleared by a master reset or clear BCC command.

SR0 is a BCC error bit. This bit can only be set in receive mode (MR2 = 0). In BISYNC normal and BISYNC transparent modes, SR0 will be set/reset once the accumulation has been stopped by the detection of the BTC/SC character and accumulation of the BCC(s).

In automatic and single accumulate modes, SR0 is set/reset after each character in the character register has been accumulated.

The rules for the detection of a BCC error are:

SR0 = 1 LRC-8: 7 least significant bits of BCC upper ≠ 0
CRC-12, CRC-16: BCC upper *or* BCC lower ≠ 0

SR0 = 0 LRC-8: 7 least significant bits of BCC upper = 0
CRC-12, CRC-16: BCC upper *and* BCC lower = 0

SR1 is a VRC error bit. When set, this bit reports a character parity error (on receive or transmit) when parity is enabled (MR4 = 1). Parity is odd/even as specified by MR5. The parity bit will be regenerated in the character register.

SR2 is a BTC/SC detect bit. When set, this bit indicates the character being accumulated is of the BTC/SC class for BISYNC normal, automatic and single accumulate modes. In BISYNC transparent mode, the BTC/SC character being accumulated must be *immediately* preceded by an odd number of DLEs for this bit to be set.

SR3 is a DLE SSC detect bit. This bit can only be set when in BISYNC normal, auto, or single accumulate modes. When set, it indicates that the character being accumulated is of the SSC class when that character was *immediately* preceded by an odd number of DLEs.

SR7, SR6, SR5, SR4 are interrupt enables. These 4 bits reflect the state of the interrupt enable command bits CR7, CR6, CR5, as follows:

SR4 - BCC error

SR5 - VRC error

SR6 - BTC/SC detect

SR7 - SSC detect

APPLICATIONS INFORMATION

Dedicated PGC

The most efficient use of the MC68653 is to dedicate one to each R/T for two way alternate (half duplex) operation or two to each R/T for two way simultaneous (full duplex) operation (see figure 9). The MPU configures each PGC (using $\overline{\text{CE1}}$) by initializing the mode register, command register, and character class array. Data transfers to or from the R/T can then be on a DMA basis with each receiver holding register ready signal used as a read request (RREQ) and each transmit holding register available signal used as a write request (WREQ) to the DMA controller. The MPU needs only to respond to enabled interrupts from each of the PGCs. The individual $\overline{\text{INT}}$ outputs can be wire-OR'd into a single MPU interrupt with one pull up resistor. Each PGC in this system has a unique address that is decoded into the respective chip enables.

The MPU or DMA controller could send a block of memory data to the PGC to be error checked without sending that data to the R/T. In that case, $\overline{\text{CE1}}$ is used.

Multiplexed PGC

One PGC may be time-shared among a few R/Ts if the MPU saves and restores the mode register and partial BCC result in the BCC registers. These registers are accessed via $\overline{\text{CE1}}$. There must be a separate save area for each R/T (serial channel) and a channel pointer indicating the last R/T that transferred or received a data character (see figure 10).

The loading of the BCC registers will clear SR0-SR3 and all previously detected special characters, i.e., DLE, BTC/SC, BCC (BISYNC modes). The BCC accumulation will start again when the next character is loaded into the character register in all accumulation modes except single. That mode requires a start accumulation command.

Figures 11 and 12 represent software flow diagrams for transmit and receive service requests. Note that interrupts from all other R/Ts must be masked during a read or write to the BCC registers so as not to affect the internal BCC address pointer. It is recommended that all R/T interrupts be masked while servicing an interrupt that accesses any PGC register.

BISYNC Operation

Table 7 is a concise listing of MC68661/2651 operating modes with recommended corresponding MC68653 BCC accumulation modes.

Character Comparator

The PGC can be used as a programmable data bus character comparator which monitors data bus transfers (MPU — peripheral, MPU — MPU, MPU — memory, memory — peripheral (via DMA)). The user selectively loads the character class array with BTC/SC and SSC characters to be compared. Status bits will be set and an interrupt can be generated upon SC and DLE - SSC detection. A match on one to 128 different characters or DLE - SSC sequences can be programmed.

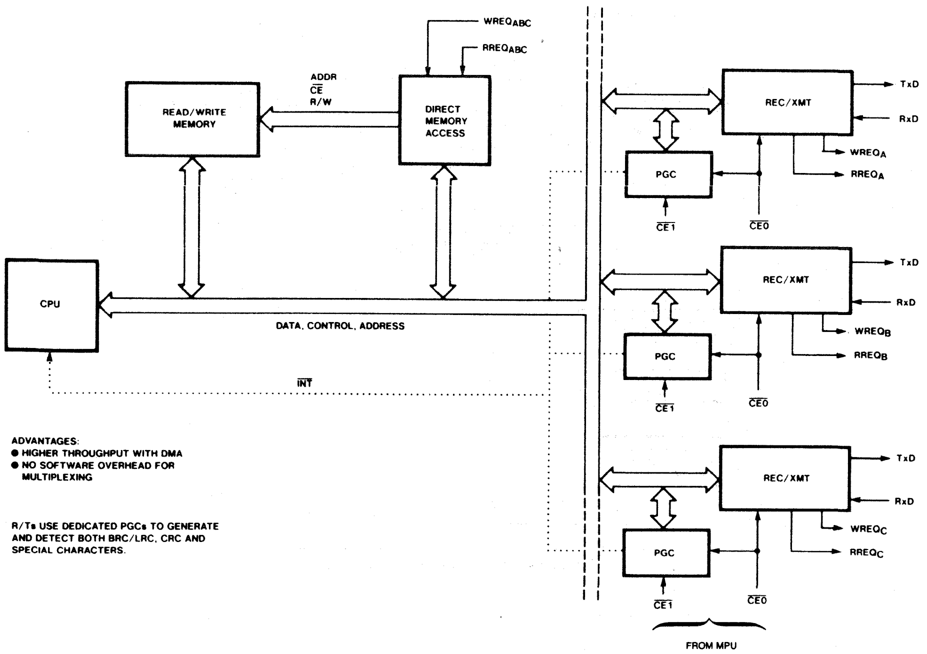
Figure 13 depicts an arrangement where the DMA controller or slave MPU handles data bus transfers, the PGC interrogates the data bus, and the host MPU responds to PGC interrupts.

Table 7. BISYNC (ANSI 3.28, ISO 1745) Modes for MC68661/2651 and MC68653

MC68661/2651 OPERATING MODES	MC68653 BCC ACCUMULATION MODES
Sync normal non-strip	BISYNC normal
Sync transparent non-strip	BISYNC transparent
Normal SYN/DLE strip[1]	BISYNC normal
Transparent SYN/DLE strip[1]	Automatic accumulate[2]
Async (with SYN/DLE characters)	BISYNC normal

NOTES
1. MPU should switch to non-strip mode after BTC detect. Otherwise a received BCC could be inadvertently stripped.
2. SSC detect should be ignored.

ONE PGC PER RECEIVER/TRANSMITTER

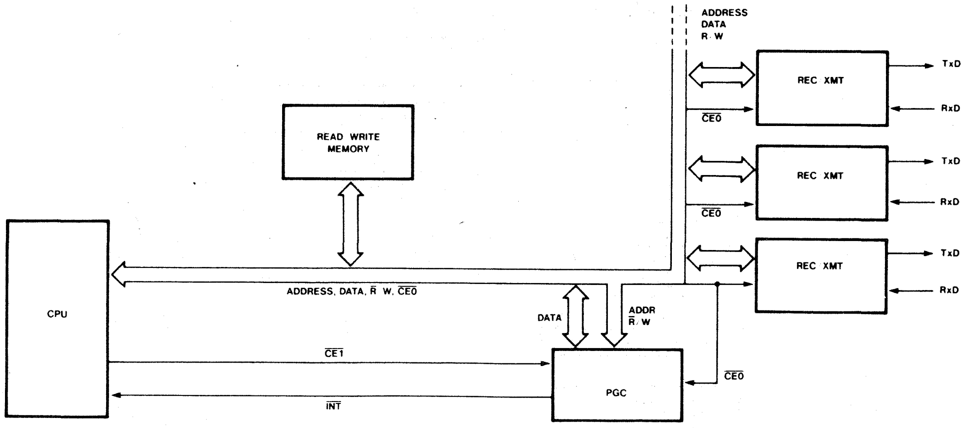


ADVANTAGES:
- HIGHER THROUGHPUT WITH DMA
- NO SOFTWARE OVERHEAD FOR MULTIPLEXING

R/Ts USE DEDICATED PGCs TO GENERATE AND DETECT BOTH BRC/LRC, CRC AND SPECIAL CHARACTERS.

Figure 9

PGC SERVICES MULTIPLE RECEIVERS/TRANSMITTERS



ONE PGC IS TIME-SHARED BY THREE R/Ts.
THE MPU READS AND RESTORES THE PARTIAL BCC REMAINDER FOR EACH SERIAL CHANNEL.

Figure 10

MULTIPLEXED PGC - TRANSMIT TO R/T

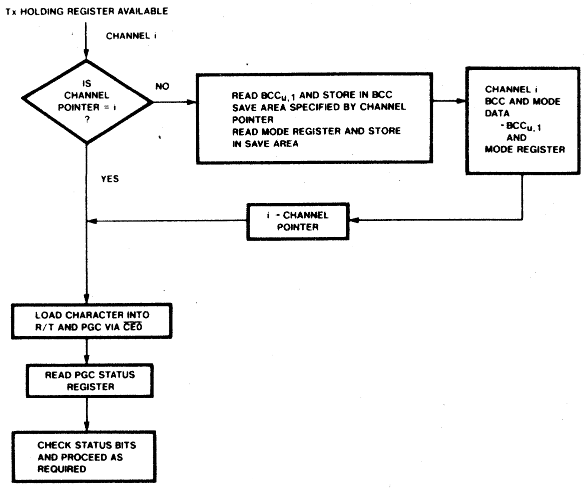


Figure 11

MULTIPLEXED PGC - RECEIVE FROM R/T

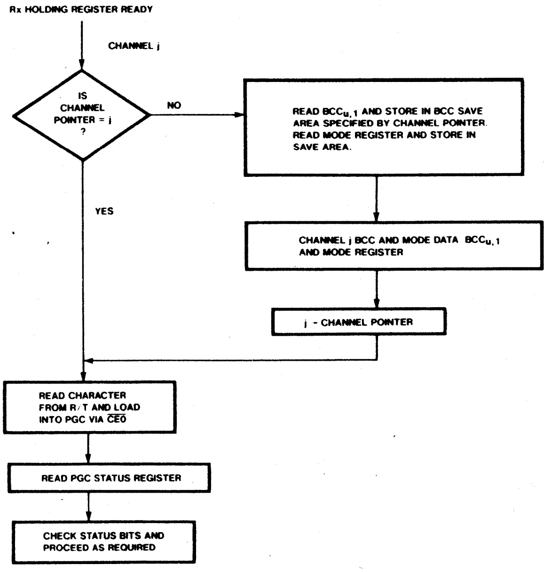
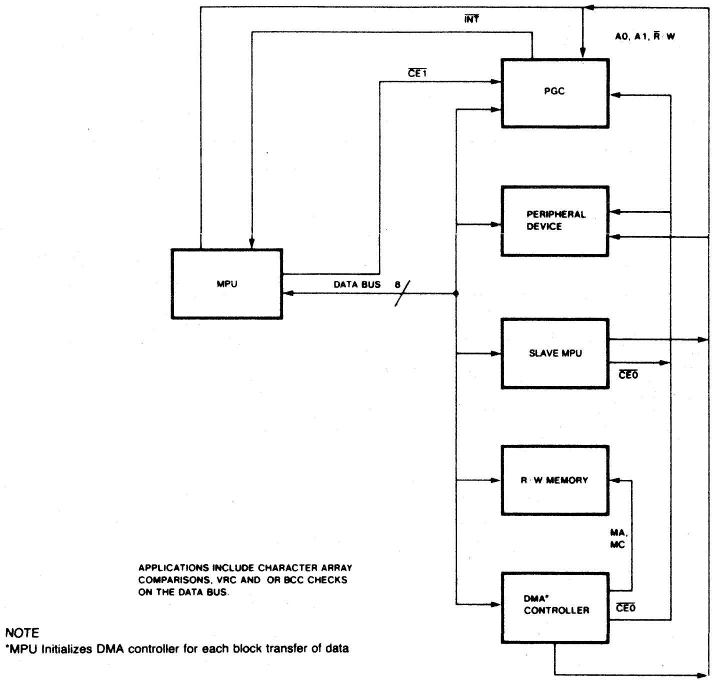


Figure 12

PGC DATA BUS MONITORING WITH DMA TRANSFERS



APPLICATIONS INCLUDE CHARACTER ARRAY COMPARISONS, VRC AND OR BCC CHECKS ON THE DATA BUS.

NOTE
*MPU Initializes DMA controller for each block transfer of data

Figure 13

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	-65 to +150	°C
All voltages with respect to ground[3]	-0.5 to +6.0	V

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	80	°C/W
Plastic		140	
Cerdip		80	

NOTES

[1] 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 at any other condition above those indicated in the operation sections of this specification is not implied.

[2] For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature and thermal resistance of 80°C/W junction to ambient (ceramic package) or 137°C/W (plastic package).

[3] This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. However, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

POWER CONSIDERATIONS

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

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

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} 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:

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

Solving equations 1 and 2 for K gives:

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

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

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0 V ± 5%

CHARACTERISTIC	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	2.0	—	V
Input Low Voltage	V_{IL}	—	0.8	V
Input Leakage Current (V_{IN} = 0 to 5.25 V)	I_{in}	—	10	μA
Output High Voltage (I_{OH} = −400 μA)	V_{OH}	2.4	—	V
Output Low Voltage (I_{OL} = 2.2 mA)	V_{OL}	—	0.45	V
Output Leakage Current (V_{OUT} = 0 to 5.25V) Data Bus Open Drain	I_{OL}	 — —	 10 10	μA
Internal Power Dissipation (Measured at T_A = T_L)	P_{INT}	—	375	mW

AC CHARACTERISTICS T_A = 0° to +70°C, V_{CC} = 5.0 V ± 5%[1, 2, 3]

CHARACTERISTIC	Symbol	Min	Max	Unit
Chip Enable Pulse Width Active	t_{CE}	250	—	ns
Chip Enable Pulse Width Inactive, Period D	t_{CED}	1750	—	ns
Chip Enable Pulse Width Inactive, Period C[1]	t_{CEC}	1750	—	ns
Address Setup	t_{AS}	10	—	ns
Address Hold	t_{AH}	10	—	ns
Control Setup	t_{CS}	10	—	ns
Control Hold	t_{CH}	10	—	ns
Data Setup[2]	t_{DSW}	150	—	ns
Data Hold	t_{DHW}	10	—	ns
Data Delay Time For Read[3]	t_{DDR}	—	200	ns
Data Bus Float Time For Read[3]	t_{DFR}	—	100	ns
Interrupt Low Delay[4]	t_{INTL}	—	1600	ns
Interrupt High Delay[4]	t_{INTH}	—	600	ns

1. t_{CEC} = 600ns during PGC initialization when no BCC accumulation is in progress.
2. t_{DS} = 50ns whenever $\overline{CE0}$ is used.
3. Test conditions: C_L = 150 pF.
4. $\overline{INT}$ is an open drain output.

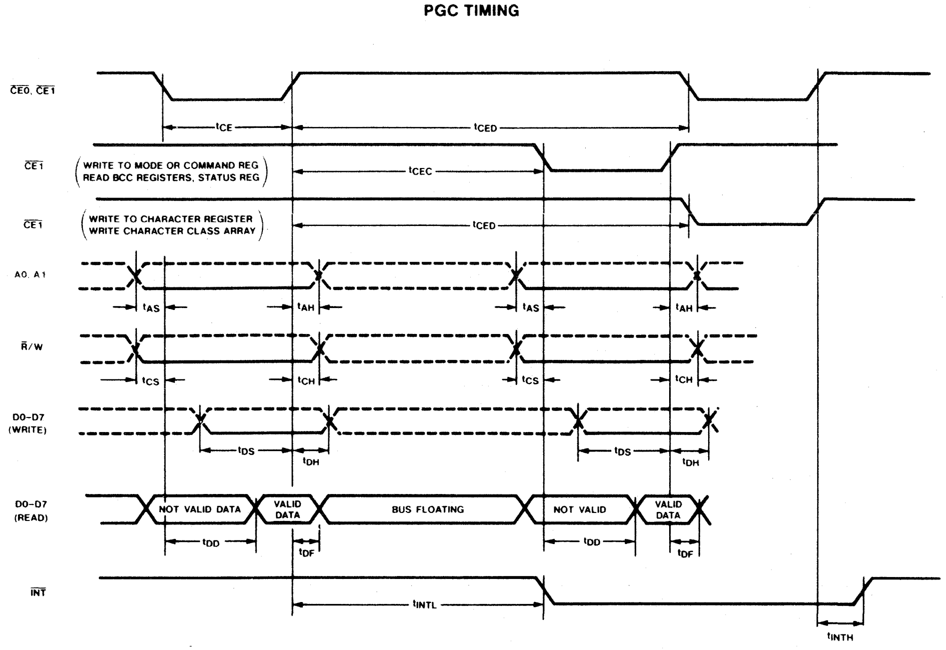


Figure 14

MC68000-to-MC68661 and MC68653 INTERFACE CIRCUIT

This circuit allows the MC68661 Enhanced Programmable Communications Interface (EPCI) and the MC68653 Polynomial Generator Checker (PGC) to be interfaced asynchronously to the MC68000 MPU bus. The circuit generates the chip-enable signals for both peripherals and the DTACK signal for the delay period of the peripherals for reliable operation. The interface circuit is shown in Figure 15.

The PGC has two chip-enable pins (CE0 and CE1). CE0 is used to access the character register used to accumulate the CRC. The CE0 input can also be used to select the data register of the EPCI and thus a write to both the EPCI and the PGC can be made at the same time. CE1 is used to select the command and status registers in the PGC and should be decoded at a different address. Since the CE period for the EPCI (t_{CED}) is 600 ns, it cannot be accessed and then re-accessed sooner than 600 ns. A number of MC68000 instructions cause accesses to the same address on successive bus cycles, the worst case being two consecutive read instructions. When this occurs, AS is re-asserted in a minimum of 1.5 clock cycles (187 ns at 8 MHz). This violates t_{CED} on the EPCI and t_{CED} and t_{CED} on the PGC (t_{CED} = t_{CED} = 1750 ns on the PGC). In order to prevent the PGC from being re-accessed during this period, the CE signals must be delayed. A state machine consisting of a 74LS161 binary counter and a 74LS74 D flip-flop are used as a digital "one shot" to generate an INHIBIT signal to delay access to the EPCI and PGC until the delay period has expired. If both the EPCI and PGC are used, the time delay should be 1750 ns or 14 clock cycles at 8 MHz. The state machine is programmed to count 14 cycles and then halt. If only the EPCI is connected, the time delay should be 600 ns which corresponds to a minimum of 5 cycles.

Both the EPCI and the PGC have 8-bit data buses. These buses are connncted to the lower byte of the MPU data bus (D0–D7) but, since both data strobes are used in the decode, either word or byte operations may be performed. If a word read operation is performed, the upper byte read will contain undefined data.

A timing diagram for the circuit in Figure 15 is shown in Figure 16. Two consecutive read bus cycles are used to show the delayed access by the INHIBIT signal. During the first read, INHIBIT is low allowing the access to occur. The rising edge of CE starts the counter and INHIBIT is set high. The next access is disabled by the INHIBIT delaying CE and the ENABLE signal generated by flip-flop U. When the counter times out, INHIBIT goes low allowing CE to reach the peripheral and generating DTACK. Timing for a write cycle is shown in Figure 17.

If the clock rate for the MC68000 is not 8 MHz, the number of clock cycles should be adjusted to meet the required time delay.

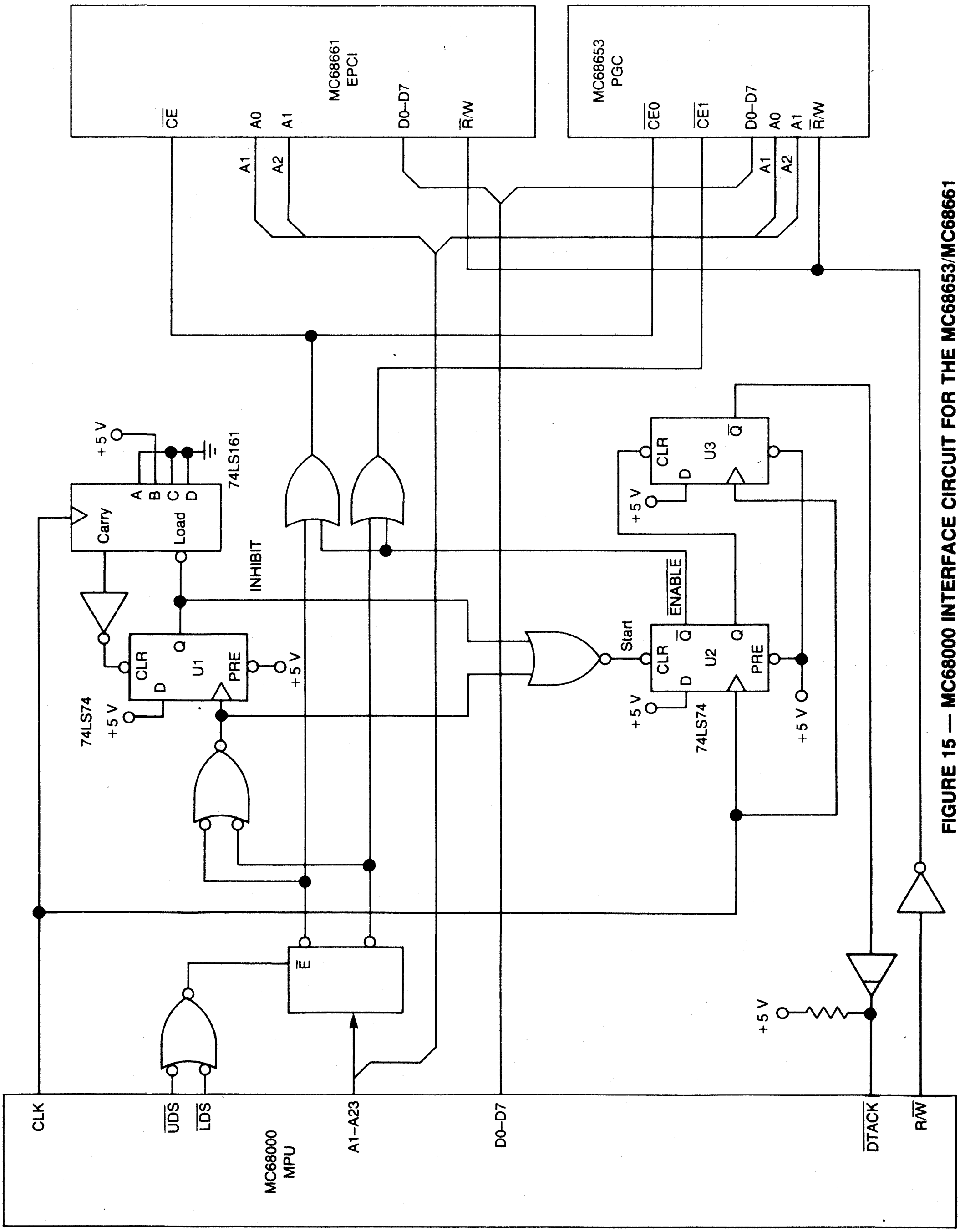
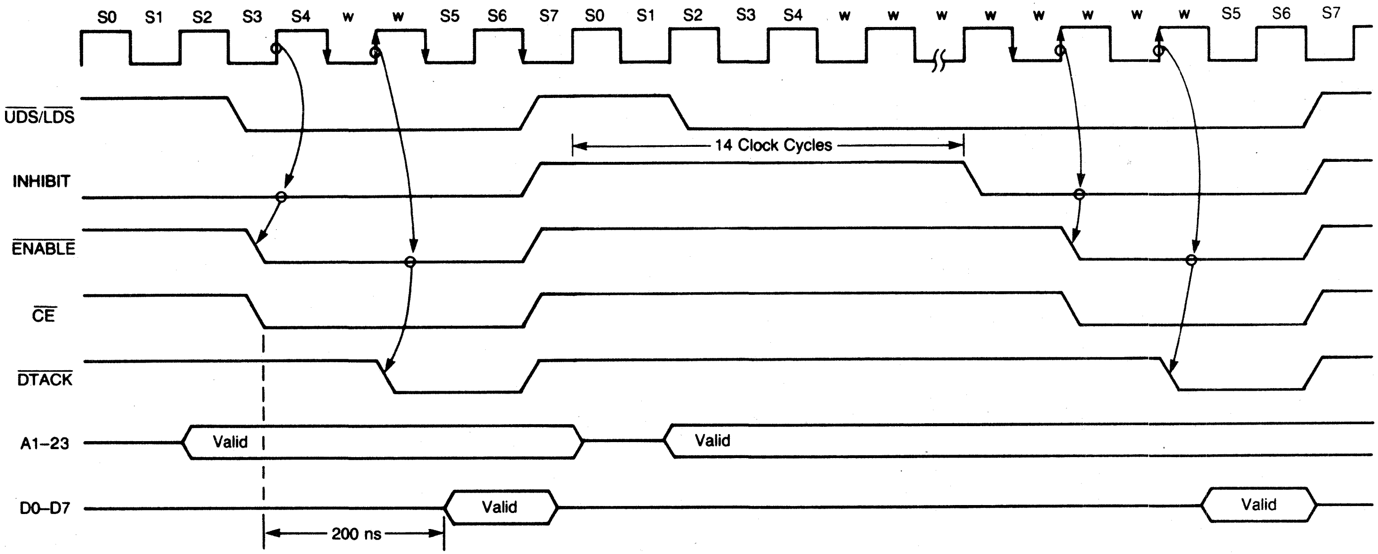


FIGURE 15 — MC68000 INTERFACE CIRCUIT FOR THE MC68653/MC68661



NOTE: THE SECOND READ CYCLE IS DELAYED 14 CLOCK CYCLES BY THE INSERTION OF WAIT CYCLE

FIGURE 16 — TIMING FOR TWO CONSECUTIVE READ CYCLES

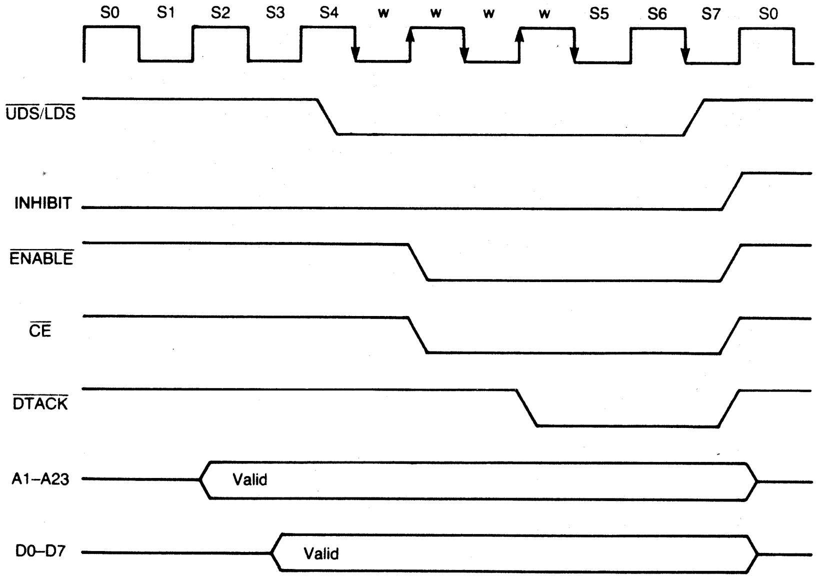


FIGURE 17 — WRITE CYCLE TIMING

MOTOROLA

MC68701 MC68A701

(1.0 MHz) (1.5 MHz)

MC68701-1 MC68B701

(1.25 MHz) (2.0 MHz)

Advance Information

MOS

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

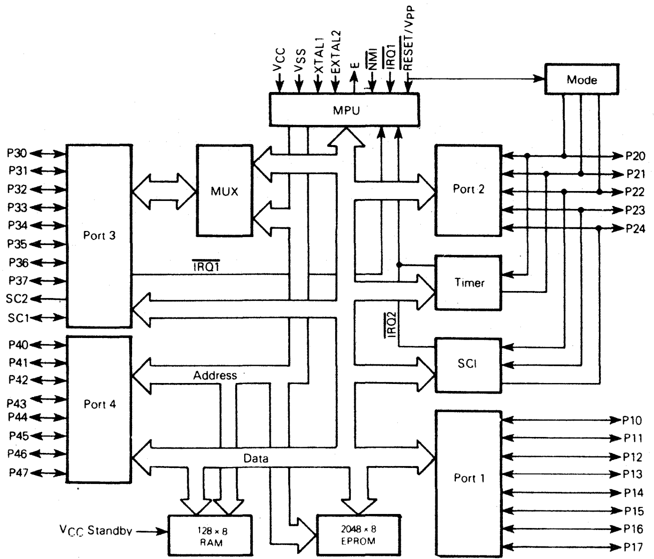
MICROCOMPUTER WITH EPROM

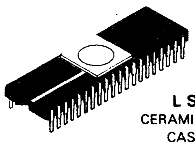
MC68701 MICROCOMPUTER UNIT (MCU)

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

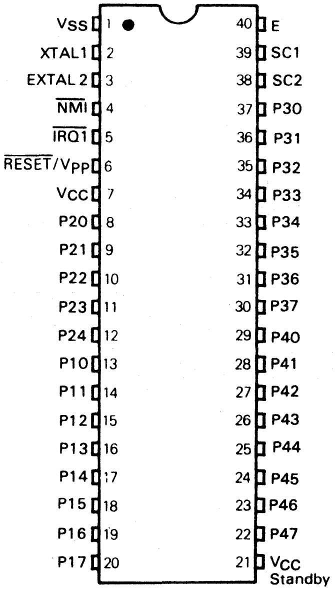
FIGURE 2 — MC68701 MICROCOMPUTER BLOCK DIAGRAM





L SUFFIX
CERAMIC PACKAGE
CASE 715

FIGURE 1 — PIN ASSIGNMENT



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}	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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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}	3.579	4.0	3.579	5.0	3.579	6.0	3.579	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%, $V_{SS}=0$, $T_A=0$ to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{\text{RESET}}$	V_{IH}	$V_{SS}+4.0$	–	V_{CC}	V
	Other Inputs*		$V_{SS}+2.0$	–	V_{CC}	
Input Low Voltage	$\overline{\text{RESET}}$	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) See Note	Port 4	I_{in}	–	–	0.6	mA
	SC1		–	–	1.0	
Input Current ($V_{in}=0$ to 5.25 V)	NMI, IRQ1	I_{in}	–	1.5	2.5	μA
Input Current See Note	$\overline{\text{RESET}}/V_{PP}$					
($V_{in}=0$ to 0.4 V)		I_{in}	–	−2.0	–	mA
($V_{in}=4.0$ V to V_{CC})			–	–	8.0	
Three-State (Off State) Input Current ($V_{in}=0.5$ to 2.4 V)	P10-P17, P30-P37	I_{TSI}	–	2.	10	μA
	P20-P24		–	10.0	100	
Output High Voltage ($I_{load}=-205$ μA, $V_{CC}=\text{min}$)	P30-P37	V_{OH}	$V_{SS}+2.4$	–	–	V
($I_{load}=-145$ μA, $V_{CC}=\text{min}$)	P40-P47, E, SC1, SC2		$V_{SS}+2.4$	–	–	
($I_{load}=-100$ μA, $V_{CC}=\text{min}$)	Other Outputs		$V_{SS}+2.4$	–	–	
Output Low Voltage ($I_{load}=2.0$ mA, $V_{CC}=\text{min}$)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Darlington Drive Current ($V_O=1.5$ V)	P10-P17	I_{OH}	1.0	2.5	10.0	mA
Internal Power Dissipation (measured at $T_A=0$°C in Steady-State Operation)		P_{INT}	–	–	1500	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	Powerdown	V_{SBB}	4.0	–	5.25	V
	Powerup	V_{SB}	4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	–	–	6.0	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

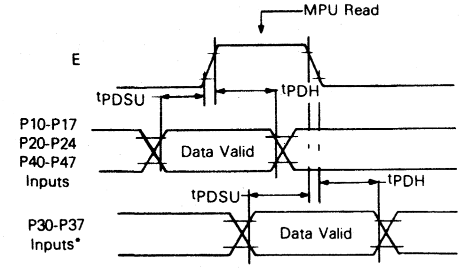
*Except Mode Programming Levels; See Figure 17.

NOTE: Port 4, SC1, and $\overline{\text{RESET}}/V_{PP}$ I_{in} differ from MC6801/03/03 NR Values.

PERIPHERAL PORT TIMING (Refer to Figures 3-6)

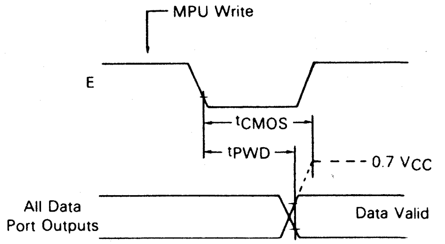
Characteristics	Symbol	Min	Typ	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	–	ns
Delay Time, Enable Positive Transition to OS3 Negative Transition	t_{OSD1}	–	–	350	ns
Delay Time, Enable Positive Transition to OS3 Positive Transition	t_{OSD2}	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid					
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 3 — DATA SETUP AND HOLD TIMES (MPU READ)



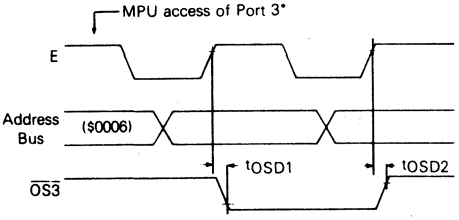
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

FIGURE 4 — DATA SETUP AND HOLD TIMES (MPU WRITE)



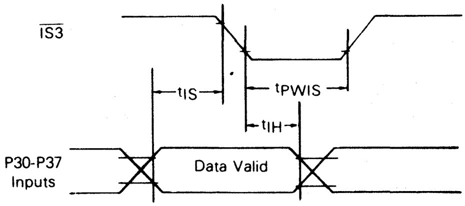
NOTES:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

FIGURE 5 — PORT 3 OUTPUT STROBE TIMING (SINGLE-CHIP MODE)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

FIGURE 6 — 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 7 — CMOS LOAD

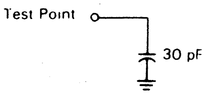
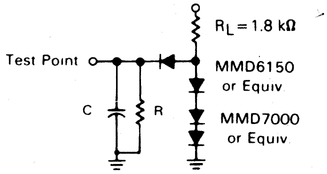


FIGURE 8 — TIMING TEST LOAD PORTS 1, 2, 3, 4



C = 90 pF for P30-P37, P40-P47, E, SC1, SC2
= 30 pF for P10-P17, P20-P24
R = 16.5 kΩ for P40-P47, E, SC1, SC2
= 24 kΩ for P10-P17, P20-P24
= 12 kΩ for P30-P37

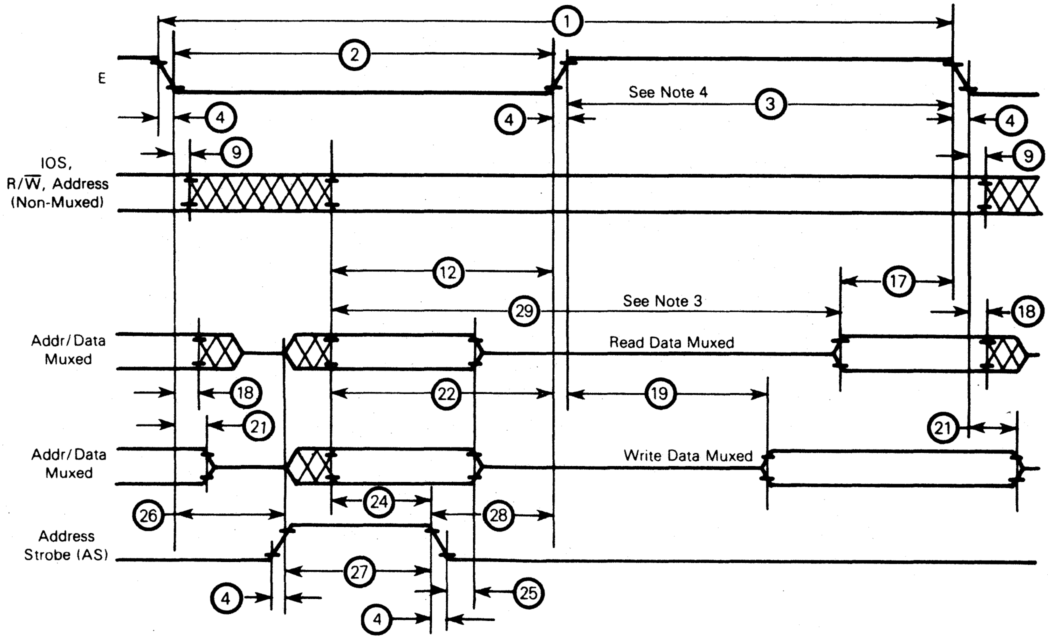
BUS TIMING (See Notes 2 and 3)

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 ns min (1.0 HMz devices), 80 ns min (1.25 MHz devices), 65 ns min (1.5 MHz devices), 50 ns min (2.0 MHz devices).

FIGURE 9 — 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 10 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 $\overline{\text{RESET}}$/V_{pp} 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.

In addition, MC6801 modes 1R and 6R, available as a mask option, are not available in the MC68701.

FIGURE 10 — MC68701/6801/6803 PROGRAMMING MODEL

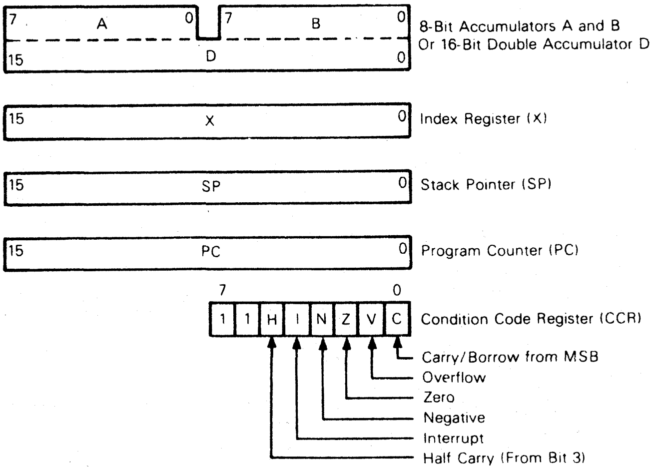


TABLE 1 — NEW INSTRUCTIONS

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

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 11. 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 12.

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 11 — SINGLE-CHIP MODE

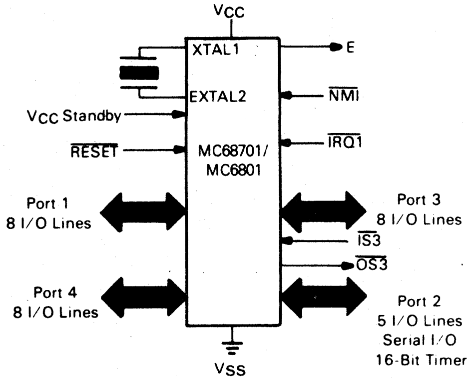


FIGURE 12 — SINGLE-CHIP DUAL PROCESSOR CONFIGURATION

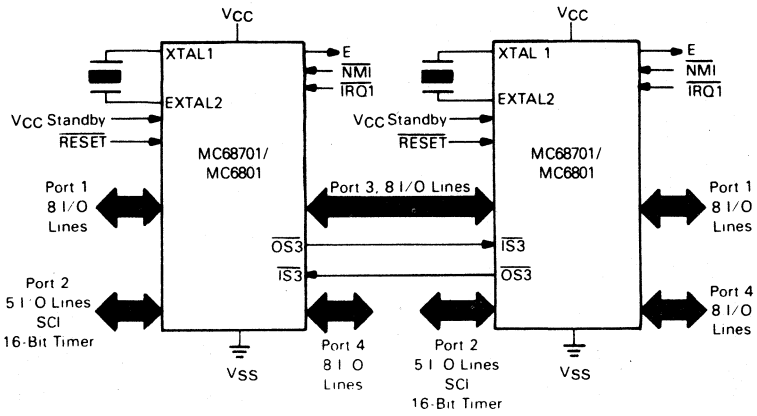
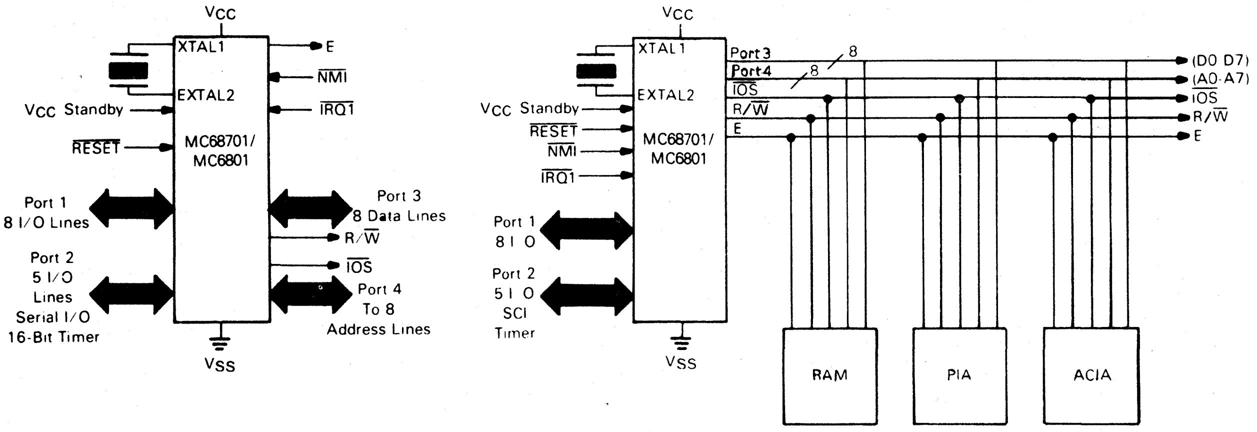


FIGURE 13 — EXPANDED NON-MULTIPLEXED CONFIGURATION



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

Figure 14 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 15. 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 $\overline{\text{RESET}}$ 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 16. 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 17 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 18. The first 32 locations of each map are reserved for the MCU internal registers as shown in Table 4, with exceptions as indicated.

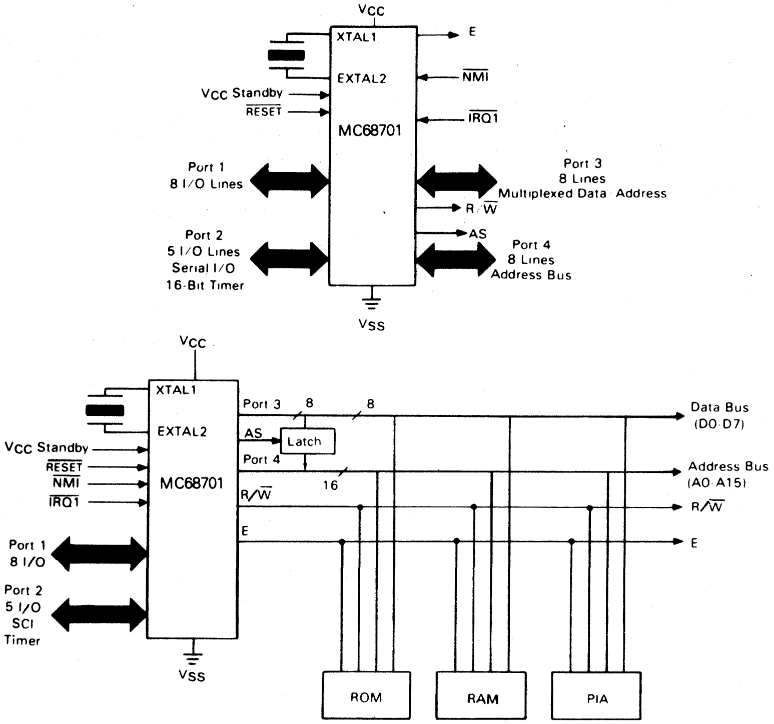
TABLE 3 — MODE SELECTION SUMMARY

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 14 — 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 15 — TYPICAL LATCH ARRANGEMENT

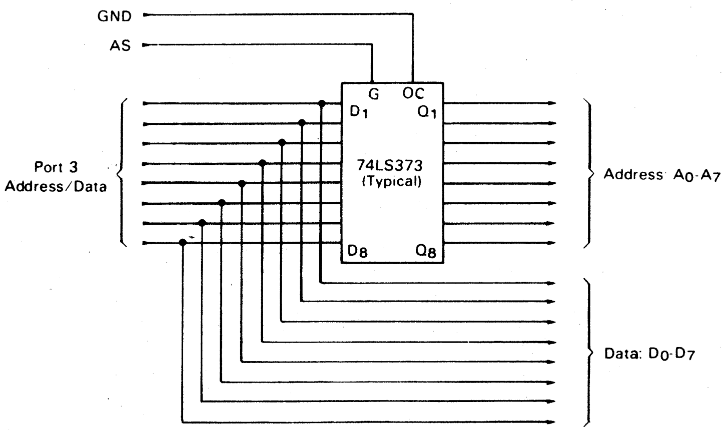
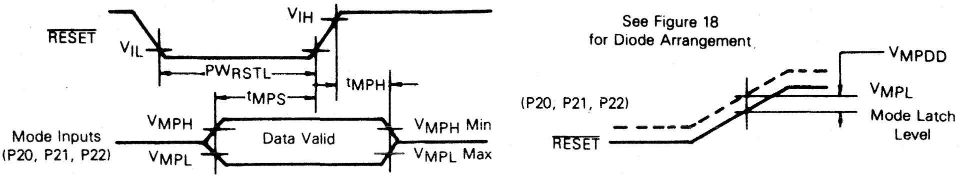


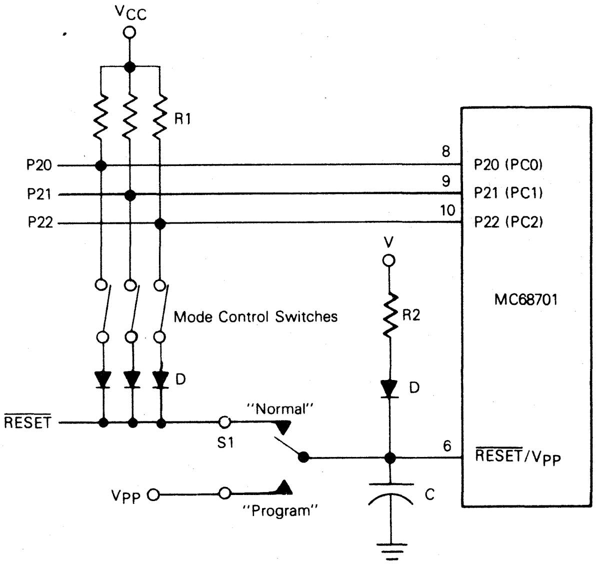
FIGURE 16 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 16)

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

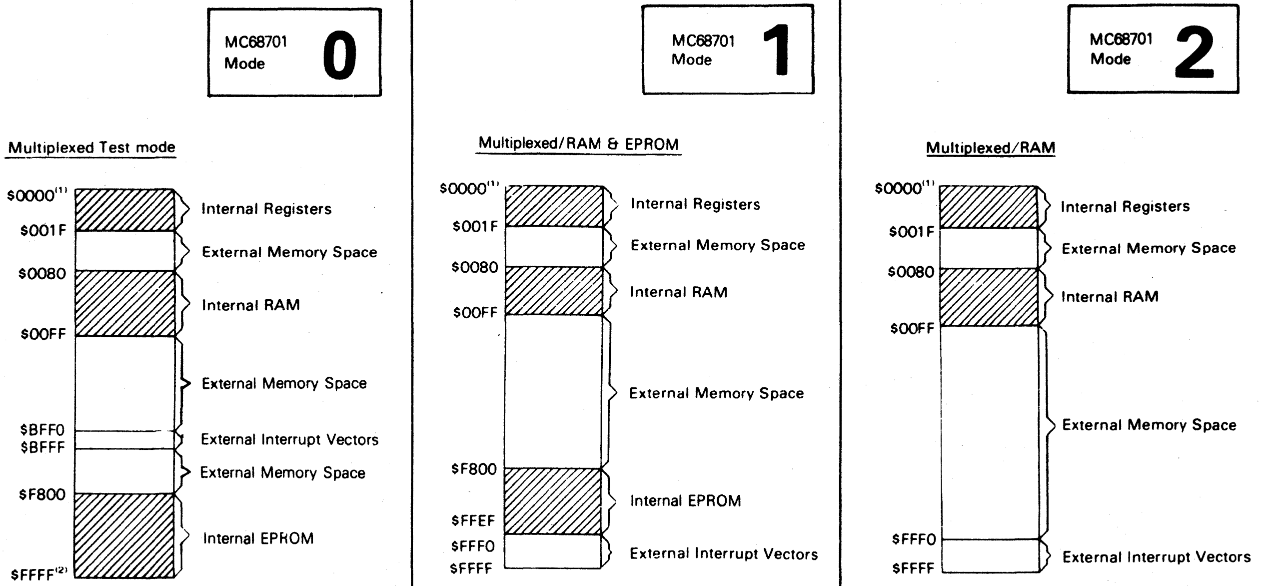
FIGURE 17 — TYPICAL MODE PROGRAMMING CIRCUIT



Notes:

1. Mode 0 as shown (switches closed).
2. R1 = 10k ohms (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 MC6801. 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 18 — MC68701 MEMORY MAPS



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.

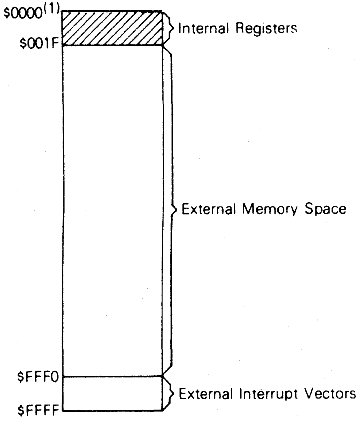
Notes:
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Internal EPROM addresses $FFF0 to $FFFF are not usable.

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

FIGURE 18 — MC68701 MEMORY MAPS (CONTINUED)

MC68701 Mode 3

Multiplexed/No RAM or EPROM

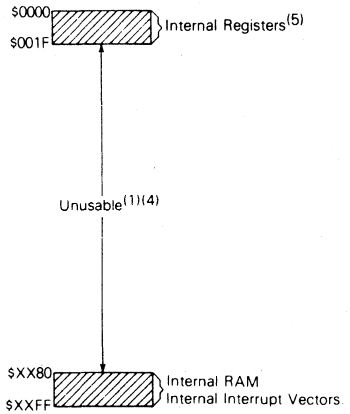


Notes:

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

MC68701 Mode 4

Single Chip Test

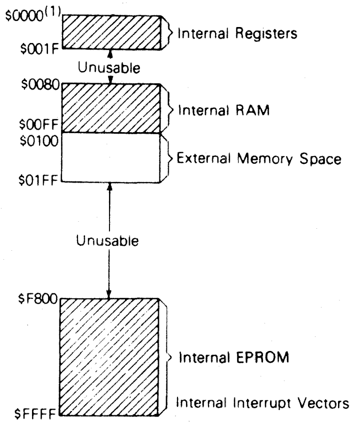


Notes:

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

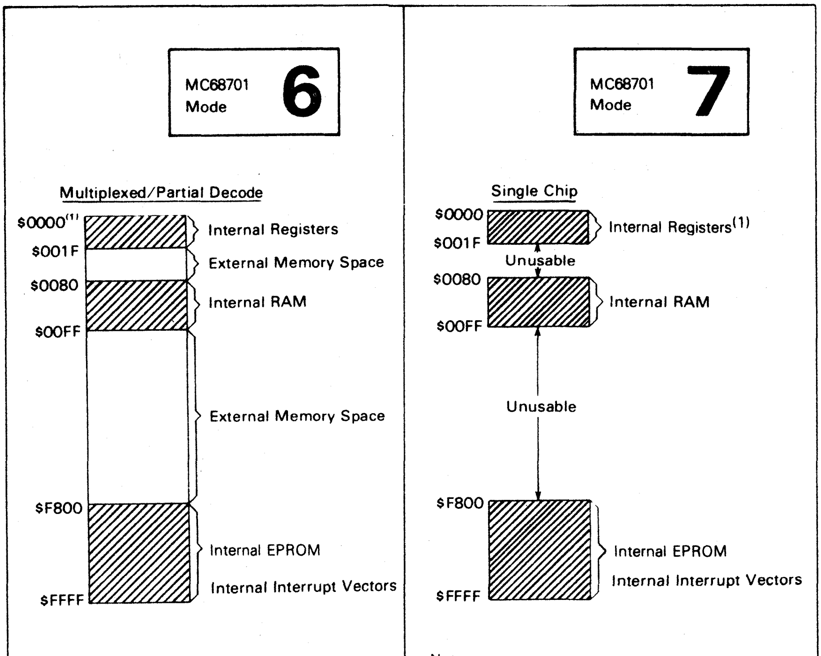
Non-Multiplexed/Partial Decode



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 18 — MC68701 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.

Note:

1) MCU read of the Port 3 Data Direction Register will access the Port 3 Data Register.

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

MC68701 INTERRUPTS

The MCU supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt (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: IRQ1 and IRQ2. The Programmable Timer and Serial Communications Interface use an internal IRQ2 interrupt line, as shown in Figure 2. External devices (and IS3) use IRQ1. An IRQ1 interrupt is serviced before IRQ2 if both are pending.

All 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.

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

Mode 0		Modes 1-7		
MSB	**LSB**	**MSB**	**LSB**	**Interrupt**
BFFE	BFFF	FFFE	FFFF	RESET
BFFC	BFFD	FFFC	FFFD	NMI
BFFA	BFFB	FFFA	FFFB	Software Interrupt (SWI)
BFF8	BFF9	FFF8	FFF9	IRQ1 (or 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 19 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 RESET timing are illustrated in Figures 20 and 21.

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 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 22.

RESET/V_{PP}

This input is used to reset the MCU internal state and provide an orderly startup procedure. During powerup, 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. 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.

NMI (NON-MASKABLE INTERRUPT)

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

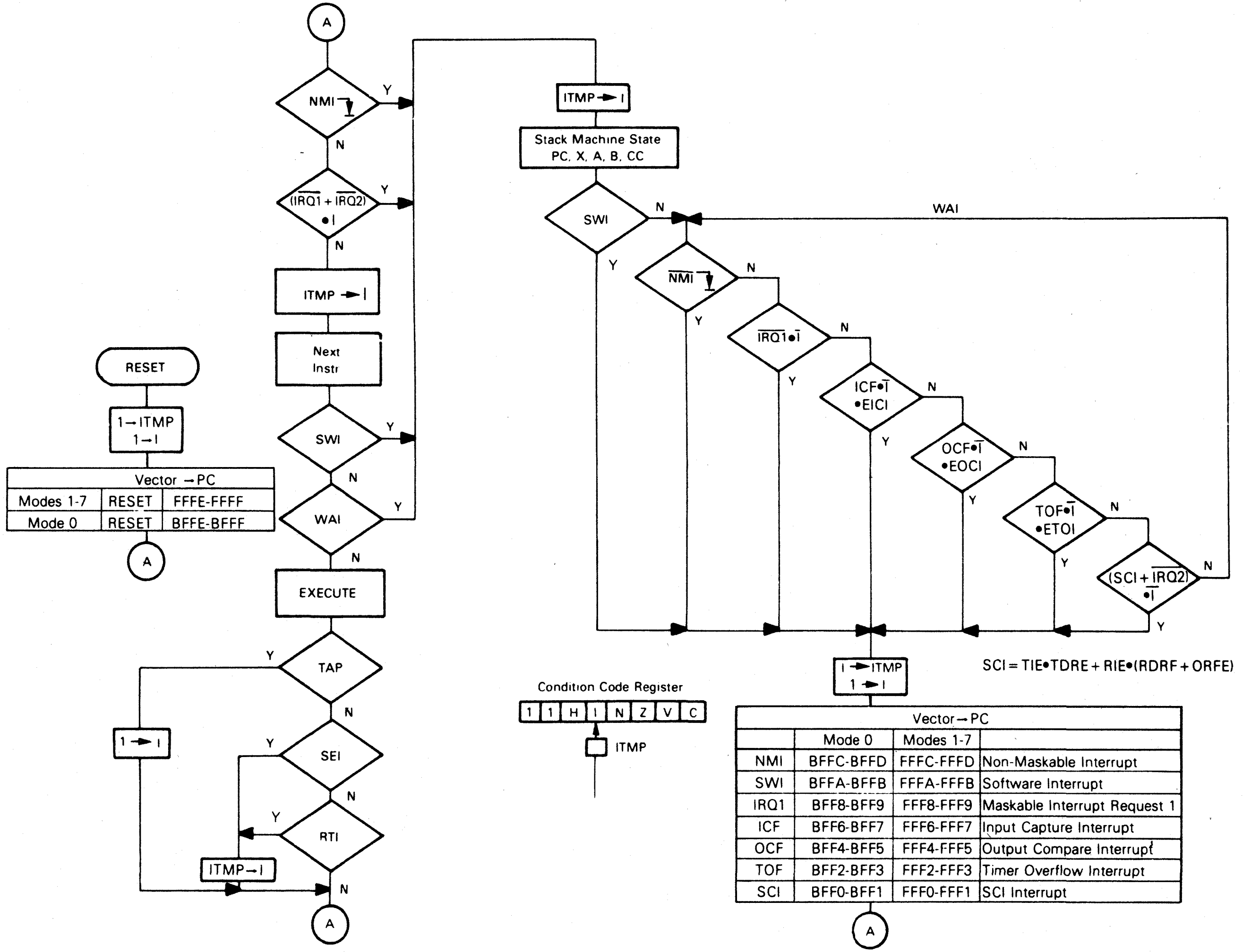
IRQ1 (MASKABLE INTERRUPT REQUEST 1)

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 characteristics.

FIGURE 19 — INTERRUPT FLOWCHART



Vector → PC		
Modes 1-7	RESET	FFFE-FFFF
Mode 0	RESET	BFFE-BFFF

SCI = TIE•TDRE + RIE•(RDRF + ORFE)

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

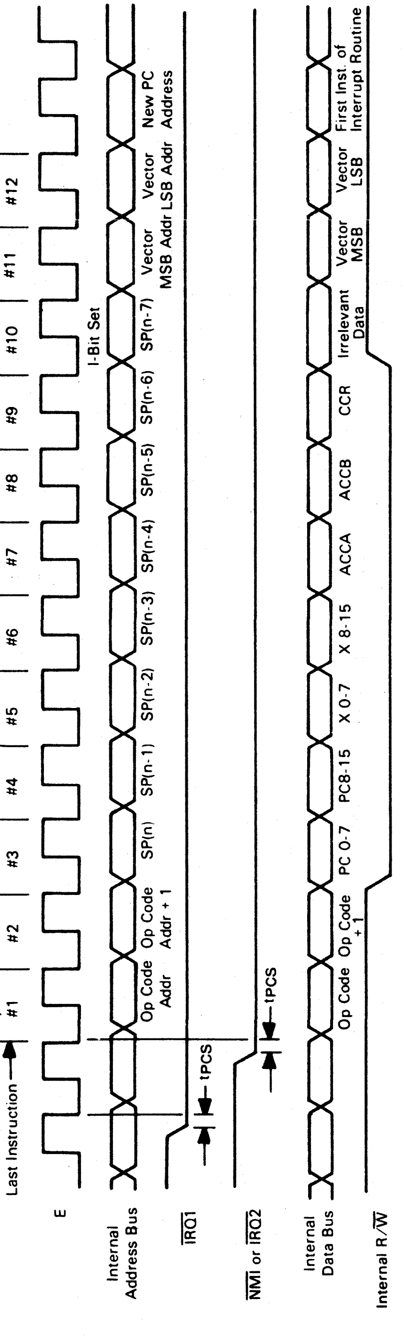


FIGURE 20 — INTERRUPT SEQUENCE

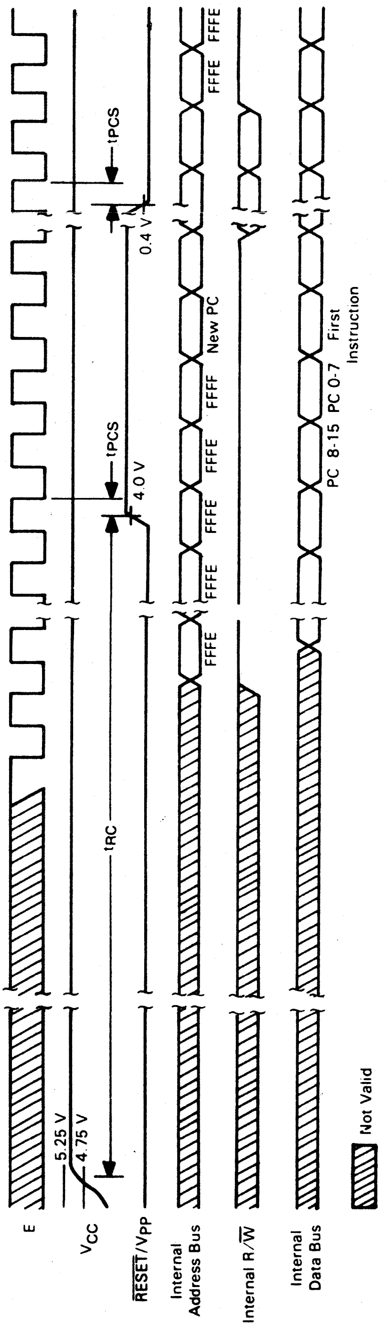


FIGURE 21 — RESET TIMING

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.

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

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.

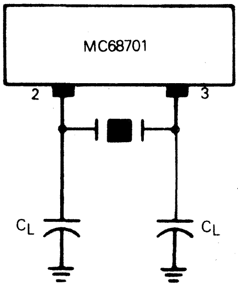
SC1 and SC2 In Single Chip Mode

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 $\overline{\text{IS3}}$ 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 the Port 3 Control and Status Register and are discussed in the Port 3 description. 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 5.

FIGURE 22 — MC68701 OSCILLATOR CHARACTERISTICS

(a) Nominal Recommended Crystal Parameters



C_L = 20 pF (typical)

NOTE

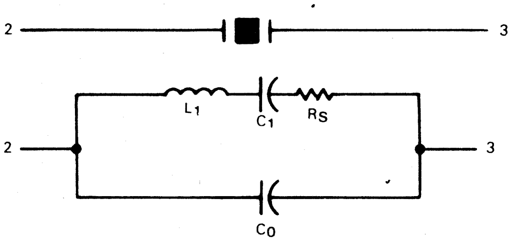
TTL-compatible oscillators may be obtained from:

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

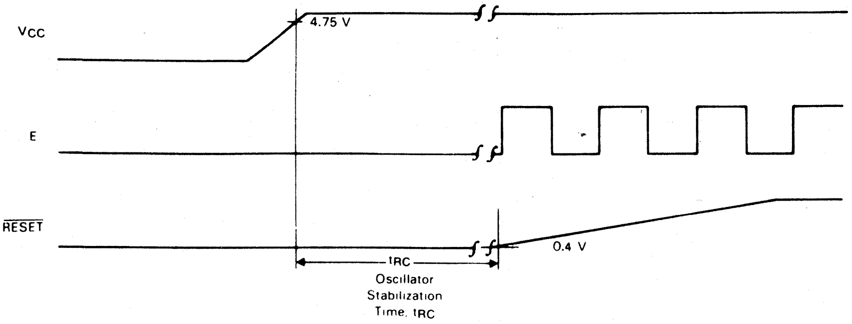
	3.58 MHz	4.00 MHz	5.0 MHz	6.0 MHz	8.0 MHz
RS	60 Ω	50 Ω	30-50 Ω	30-50 Ω	20-40 Ω
C_0	3.5 pF	6.5 pF	4.6 pF	4-6 pF	4.6 pF
C_1	0.015 pF	0.025 pF	0.01-0.02 pF	0.01-0.02 pF	0.01-0.02 pF
Q	>40 k	>30 k	>20 k	>20 k	>20 k

*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 ($\overline{\text{IOS}}$) 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 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 $\overline{\text{RESET}}$. 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 $\overline{\text{RESET}}$ 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.

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 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	\$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, $\overline{\text{IS3}}$ and $\overline{\text{OS3}}$, 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) $\overline{\text{OS3}}$ 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 $\overline{\text{IS3}}$ negative edge. Port 3 latch timing is shown in Figure 6.

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 $\overline{\text{IS3}}$ 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 $\overline{\text{OS3}}$ 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 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 $\overline{\text{RESET}}/V_{PP}$ 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 is cleared whenever V_{CC} Standby decreases below V_{SBB} (min). It can be set only by software and is not affected during reset.

Note that if PPC and PLC are set, they cannot be simultaneously cleared with a single MPU write. The PLC bit must be cleared prior to attempting to clear PPC. If both PPC and PLC are clear, setting PLC will also set PPC. In addition, it is assumed that Vpp is applied to the RESET/Vpp 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 (Vpp) to the $\overline{\text{RESET}}$/Vpp 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 (Vpp) from the $\overline{\text{RESET}}$/Vpp 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 (Vpp) from the $\overline{\text{RESET}}$/Vpp 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 IMBEG, 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:

$$\begin{aligned}\text{WAIT (MPU E-cycles)} &= t_{PP}\bullet(\text{MCU INPUT FREQ})/4\bullet10^6\\ &= 50000(4\bullet10^6)/4\bullet10^6\\ &= 50000\end{aligned}$$

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 23.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during $\overline{\text{reset}}$ 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.

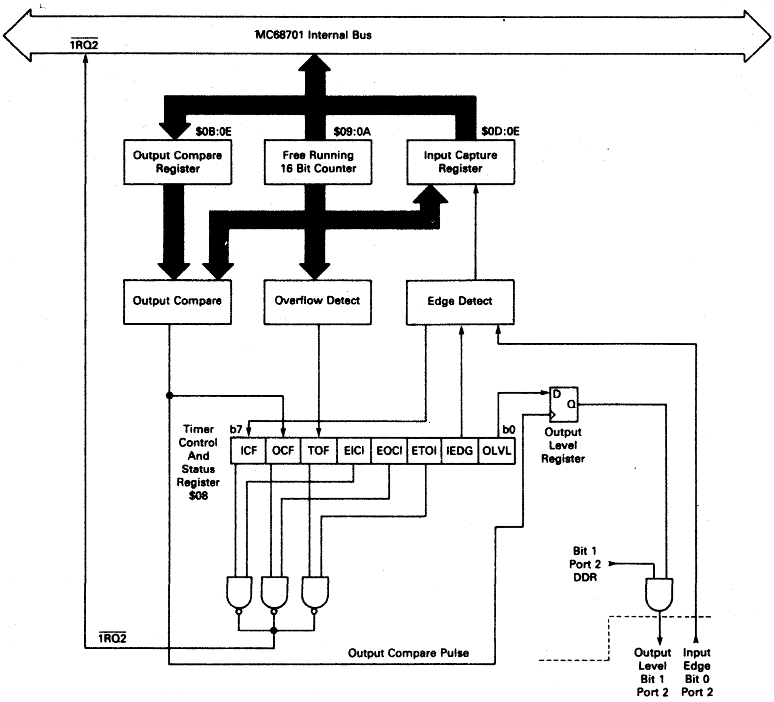
OUTPUT COMPARE REGISTER ($0B:0C)

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 23 — 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 by 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 24. 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 by 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 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 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

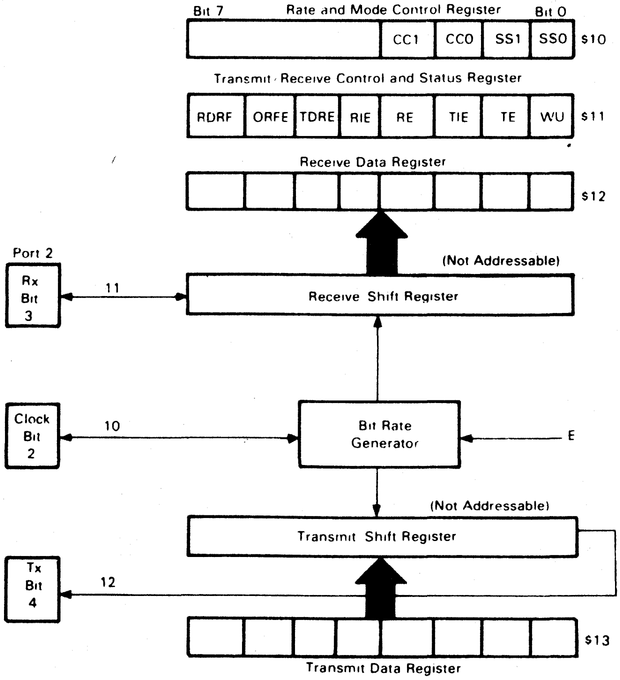
TABLE 6 — SCI BIT TIMES AND RATES

SS1:SS0	$4f_o \rightarrow$	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)		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 24 — 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.

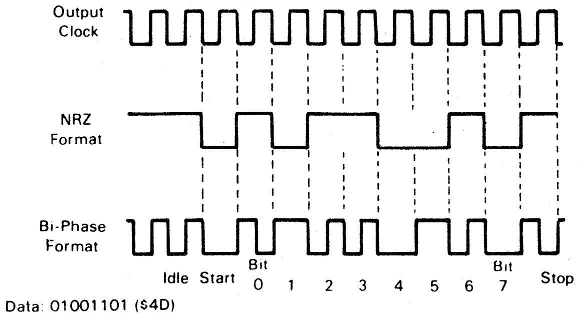
At this point one of two situations exist: 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit-Data Register (TDRE = 0), it will be transferred to the output serial shift register (synchronized 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

---

*Devices made with mask numbers T7A and CB4 do not transfer unframed data to the Receive Data Register.

FIGURE 25 — SCI DATA FORMATS



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 25.

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.

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 11. 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 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 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 26.

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	40	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		2	1	4D	TSTA		2	1	81	CMPA	↑	2	2	B5	BITA		4	3	E9	ADCB		4	2
1A	•	↓			4E	•				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	•	↓			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
   REL ≡ Relative
   INDXD ≡ Indexed
   EXTND ≡ Extended
   IMMED ≡ Immediate
   Dir ≡ Direct
2. Unassigned op codes 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	Mnemonic	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5	4	3	2	1	0
																		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X – M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X – 1 → X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	3	1	SP – 1 → SP	●	●	●	●	●	●
Increment Index Reg	INX													08	3	1	X + 1 → X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	3	1	1 SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	●	●	↕	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	●	●	↕	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	●	●	↕	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	●	●	↕	↕	R	●
Index Reg → Stack Pntr	TXS													35	3	1	X – 1 → SP	●	●	●	●	●	●
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	●	●	●	●	●	●
Add	ABX													3A	3	1	B + X → X	●	●	●	●	●	●
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP – 1 → SP X_H → M_{SP}, SP – 1 → SP	●	●	●	●	●	●
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	●	●	●	●	●	●

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTIONS

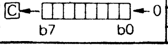
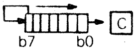
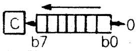
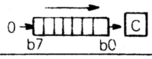
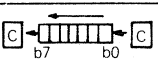
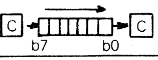
Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00 B + X → X	●	●	●	●	●	●
Add with Carry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

TABLE 10 — ACCUMULATOR AND MEMORY INSTRUCTION (CONTINUED)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	✓↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The Condition Code Register symbol explanations are listed after Table 11.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

		Direct			Relative			Index			Extnd			Inherent				Cond. Code Reg.					
																		5	4	3	2	1	0
Operations	**Mnemonic**	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	**Branch Test**	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										See Special Operations Figure 27	●	●	●	●	●	●
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 27	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

	Inherent					Cond. Code Reg.					
						5	4	3	2	1	0
Operations	**Mnemonic**	**OP**	**~**	**#**	**Boolean Operation**	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
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
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

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

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E-CYCLES

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

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

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

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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONTINUED)

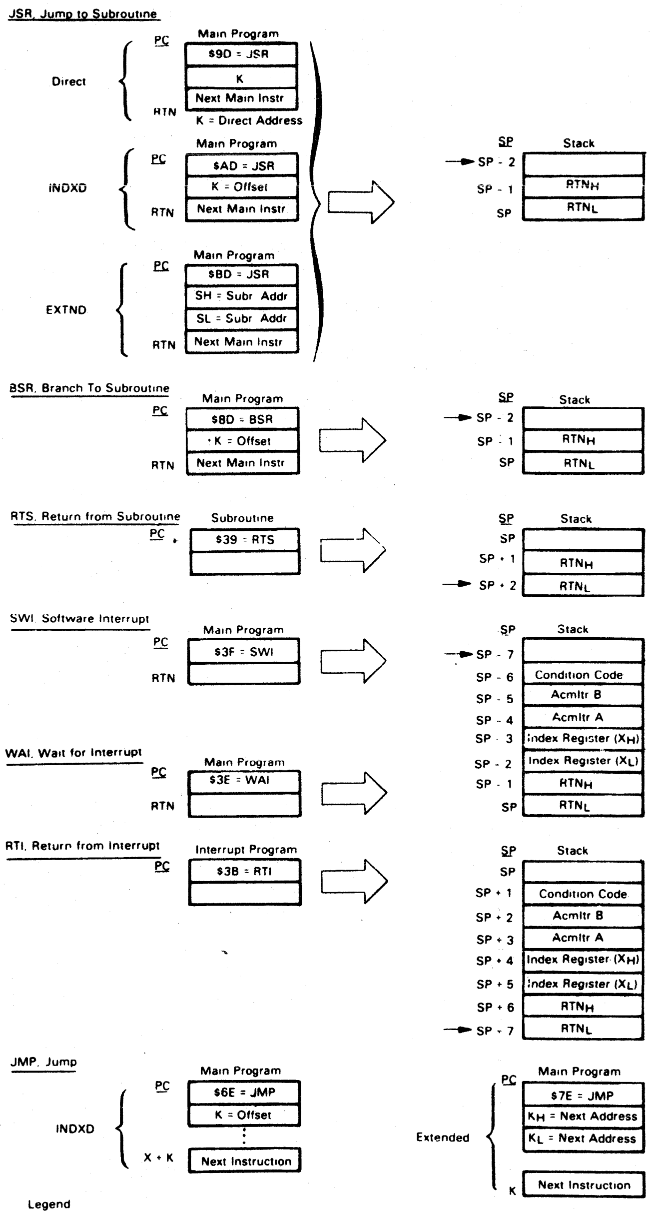
Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Op Code of Next Instruction
ABX	3	1	Op Code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

— Continued —

TABLE 14 — CYCLE-BY-CYCLE OPERATION (CONCLUDED)

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)
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 26 — 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 ***

                                  NAM    EPROM
                                  OPT    Z01,LLEN=80
                                  TTL    *** ROUTINE TO PROGRAM THE MC68701 EPROM **

                           *****************************************************
                           *
                           *  E P R O M -- A  NON-REENTRANT  ROUTINE  TO  PROGRAM
                           *               THE MC68701 EPROM.
                           *
                           *               THE ROUTINE PROGRAMS THE MC68701 EPROM
                           *               STARTING  AT  ADDRESS  "PNTR"  FROM  A
                           *               BLOCK  OF  MEMORY  STARTING  AT  "IMBEG"
                           *               AND ENDING AT "IMEND".
                           *
                           *  CALLING CONVENTION:
                           *
                           *     JSR  EPROM
                           *
                           *  NOTES:
                           *
                           *     1.  THE ROUTINE EXPECTS FOUR DOUBLE BYTE  VALUES
                           *         TO  BE INITIALIZED PRIOR  TO  BEING  CALLED.
                           *         THESE VALUES ARE:
                           *
                           *         IMBEG = A DOUBLE BYTE ADDRESS  WHICH  POINTS
                           *                 TO THE FIRST BYTE TO  BE  PROGRAMMED
                           *                 INTO THE EPROM.
                           *
                           *         IMEND = A DOUBLE BYTE ADDRESS  WHICH  POINTS
                           *                 TO THE LAST BYTE TO BE PROGRAMED IN-
                           *                 INTO THE EPROM.
                           *
                           *         PNTR  = A DOUBLE BYTE ADDRESS  WHICH  POINTS
                           *                 TO THE FIRST BYTE IN THE EPROM TO BE
                           *                 PROGRAMMED.
                           *
                           *         WAIT  = A DOUBLE BYTE COUNTER VALUE WHICH IS
                           *                 A FUNCTION OF THE MCU INPUT FREQUEN-
                           *                 CY AND IS USED WITH THE OUTPUT  COM-
                           *                 PARE FUNCTION TO  GENERATE A 50 MSEC
                           *                 TIMEOUT.  IT IS EQUIVALENT TO
                           *
                           *                 50000 * (MCU INPUT FREQ) / 4 * 10**6
                           *
                           *                 VALUES FOR TYPICAL INPUT FREQS  ARE:
                           *
                           *                   WAIT              MCU INPUT FREQ
                           *               -------------         --------------
                           *               30615 ($7797)            2.45 MHZ
                           *               50000 ($C350)            4.00 MHZ
                           *               61375 ($EFBF)            4.91 MHZ
                           *
                           *     2.  IT IS ASSUMED THAT POWER (VPP) IS  AVAILABLE
                           *         TO THE RESET PIN FOR PROGRAMMING.
                           *
                           *     3.  THIS ROUTINE  PERFORMS  NO  ERROR  CHECKING.
                           *
                           *
```

Routine parameter initialization, such as stack pointer, etc., must be done prior to entry. (Use of PRObug® will ensure all needed intitialization.)

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
```

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 turnaround 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.

A comparison table of key features for several members of the M6805 Family is shown on the last page of this data sheet.

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 $\overline{\text{INT}}$ 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

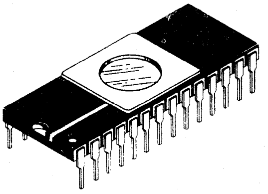
SOFTWARE FEATURES

- 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

FIGURE 1 — PIN ASSIGNMENTS

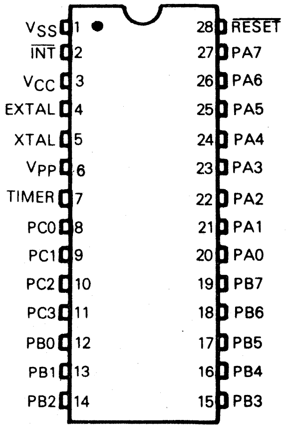
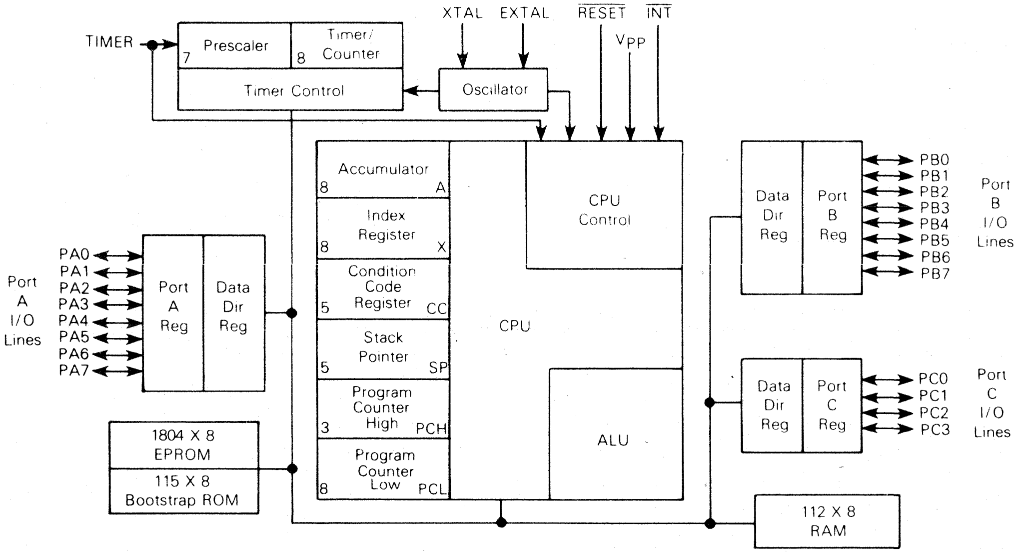


FIGURE 2 — MC68705P3 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage			
EPROM Programming Voltage (V_{PP} Pin)	V_{PP}	−0.3 to +22.0	V
TIMER Pin			
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 +50	°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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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:

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

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:

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

Solving equations 1 and 2 for K gives:

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

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 Oscillator Frequency	V_{PP}	20.0	21.0	22.0	V
Vpp 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 50°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	f_{osc}	0.4	–	4.4	MHz
Instruction Cycle Time ($4/f_{osc}$)	t_{cyc}	0.950	–	10	μs
$\overline{\text{INT}}$ or Timer Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	$t_{cyc}+250$	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Cap = 1.0 μF)	t_{RHL}	100	–	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (for ±5° Accuracy)	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 50°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage $\overline{\text{RESET}}$ ($4.75 \le V_{CC} \le 5.75$) ($V_{CC} < 4.75$) $\overline{\text{INT}}$ ($4.75 \le V_{CC} \le 5.75$) ($V_{CC} < 4.75$) 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 $\overline{\text{RESET}}$ $\overline{\text{INT}}$ 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}	–	500	TBD	mW
Input Capacitance XTAL All Other	C_{in}	 – –	 25 10	 – –	pF
$\overline{\text{INT}}$ Zero-Crossing Voltage, through a Capacitor	V_{INT}	2.0	–	4.0	V_{acp-p}
$\overline{\text{RESET}}$ 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 (Vpp 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) $\overline{\text{INT}}$ (V_{in} = 0.4 V) EXTAL (V_{in} = 2.4 V to V_{CC} Crystal Option) (V_{in} = 0.4 V Crystal Option) $\overline{\text{RESET}}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)	I_{in}	 – – – – −4.0	 – 20 – – –	 20 50 10 −1600 −50	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 50°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}	3.5	–	–	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}	−0.3	–	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}	−0.3	–	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}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	µA

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

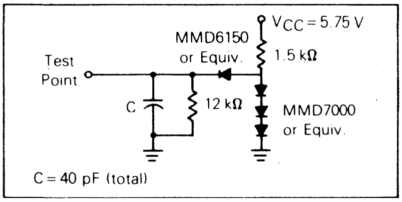


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

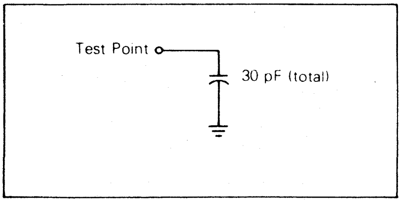
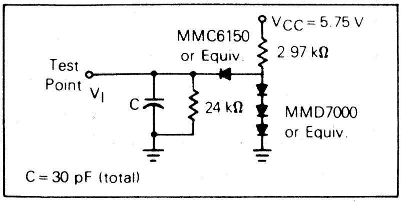


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$ — 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.

$\overline{\text{RESET}}$ — This pin has a Schmitt Trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low. Refer to RESETS for additional information.

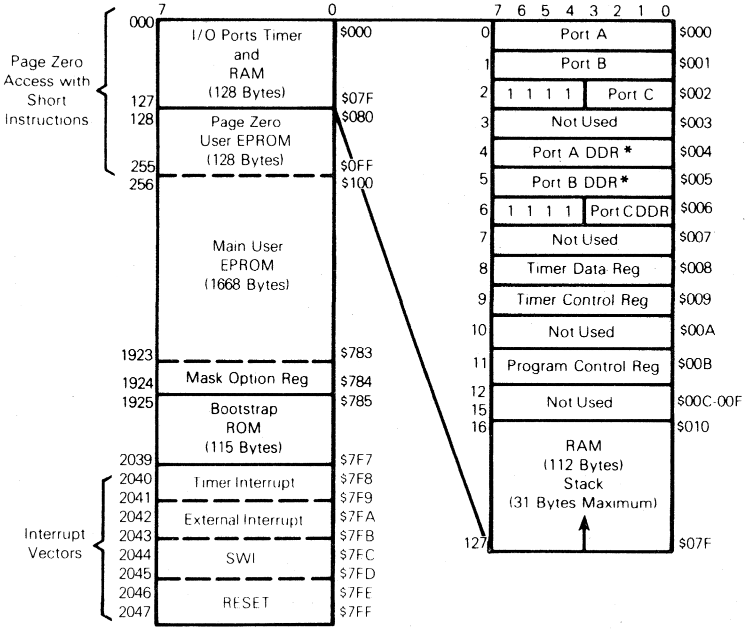
V_{PP} — This pin is used when programming the EPROM. By applying the programming voltage to this pin, one of the requirements is met for programming the EPROM. In normal operation, this pin is connected to V_{CC}. Refer to PROGRAMMING FIRMWARE and ELECTRICAL CHARACTERISTICS.

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 6, 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.

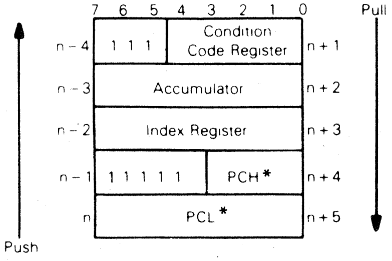
FIGURE 6 — MC68705P3 MCU MEMORY CONFIGURATION



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 7. 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 7 — INTERRUPT STACKING ORDER



* For subroutine calls, only PCH and PCL are stacked

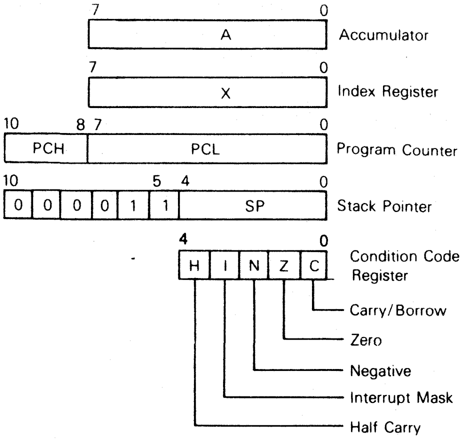
CENTRAL PROCESSING UNIT

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 8 and are explained in the following paragraphs.

FIGURE 8 — 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 ($\overline{\text{INT}}$) 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.

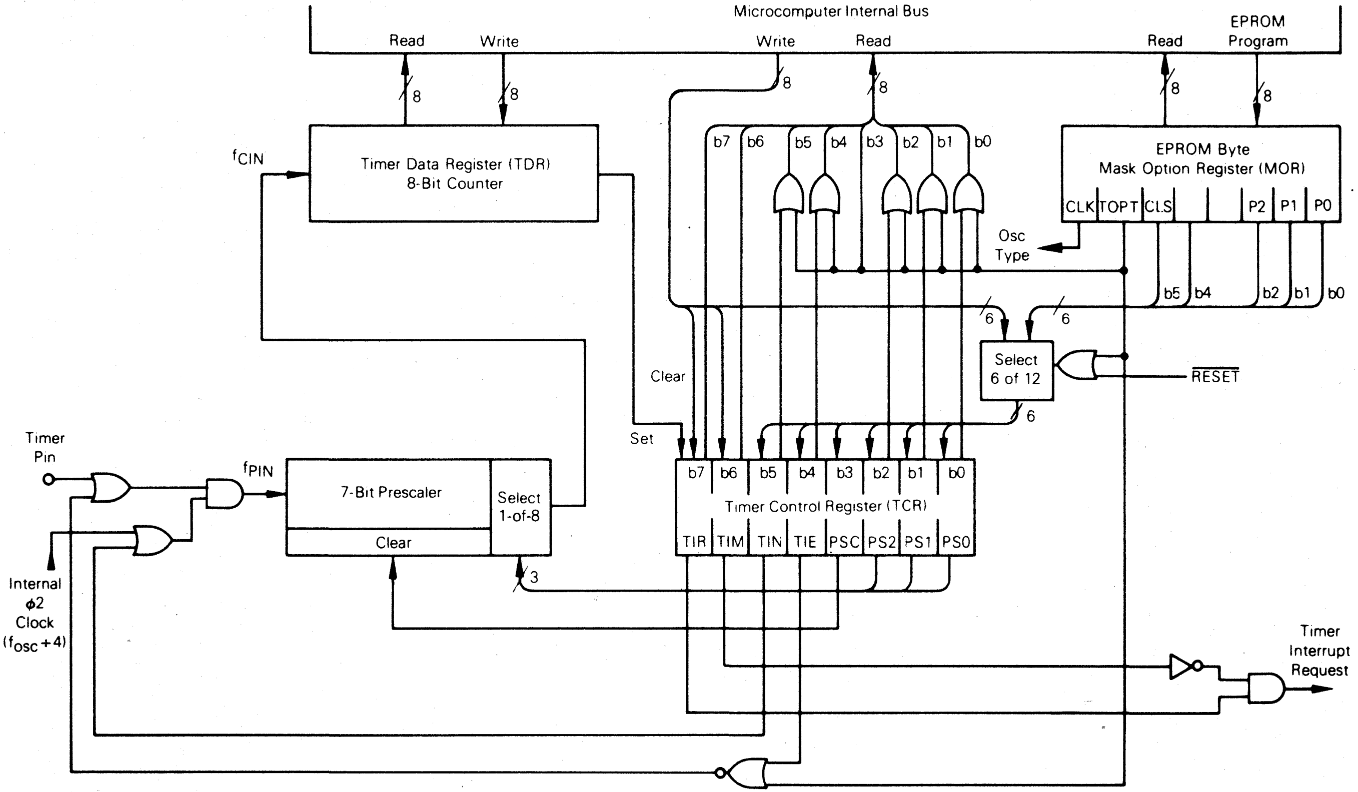
TIMER

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 9. 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 9 —MC68705P3 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 9 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.

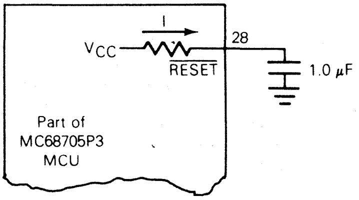
The MOR controlled mode of the timer is selected when the TOPT (Timer Option) bit (b6) in the MOR is programmed to a logical "1" to emulate the mask-programmable prescaler of the MC6805P2 and MC6805P4. The timer circuits are the same as described above; however, the Timer Control Register (TCR) is configured 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 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 10, typically provides sufficient delay.

FIGURE 10 — POWER-UP RESET DELAY CIRCUIT



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 when it senses logical "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. See Figure 15 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 12.

FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

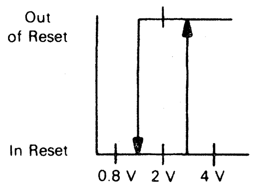
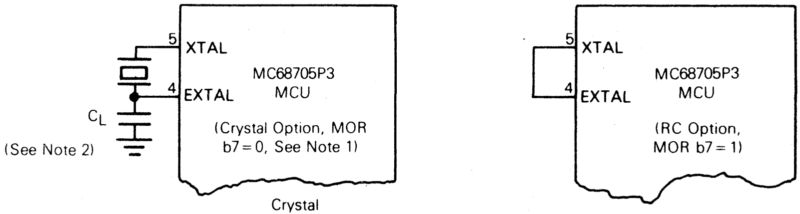
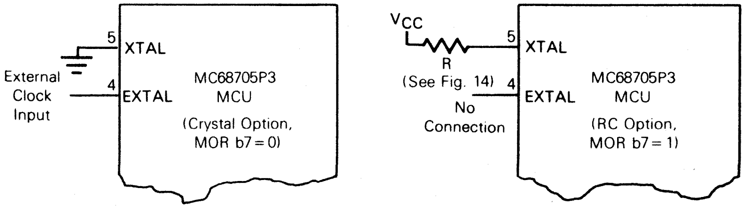


FIGURE 12 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% to 50% Accuracy
Typical t_{cyc} = 1.25 μs
External Jumper



External Clock

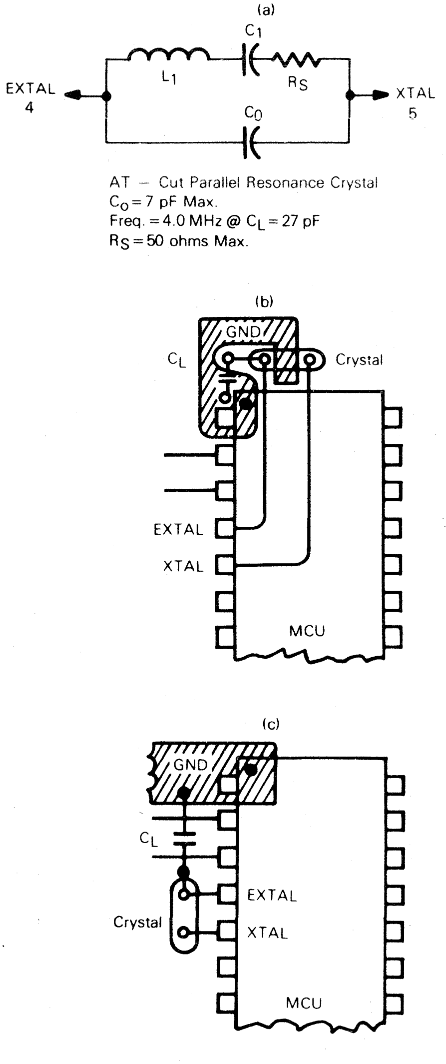
Approximately 10% to 25% Accuracy
(Excludes Resistor Tolerance)
External Resistor

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 13. A resistor selection graph is given in Figure 14.

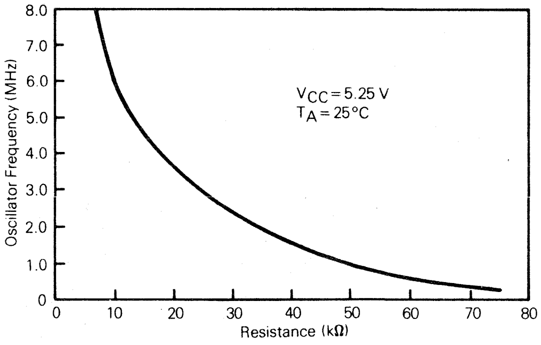
FIGURE 13 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



AT — Cut Parallel Resonance Crystal
$C_0 = 7$ 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 14 — 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

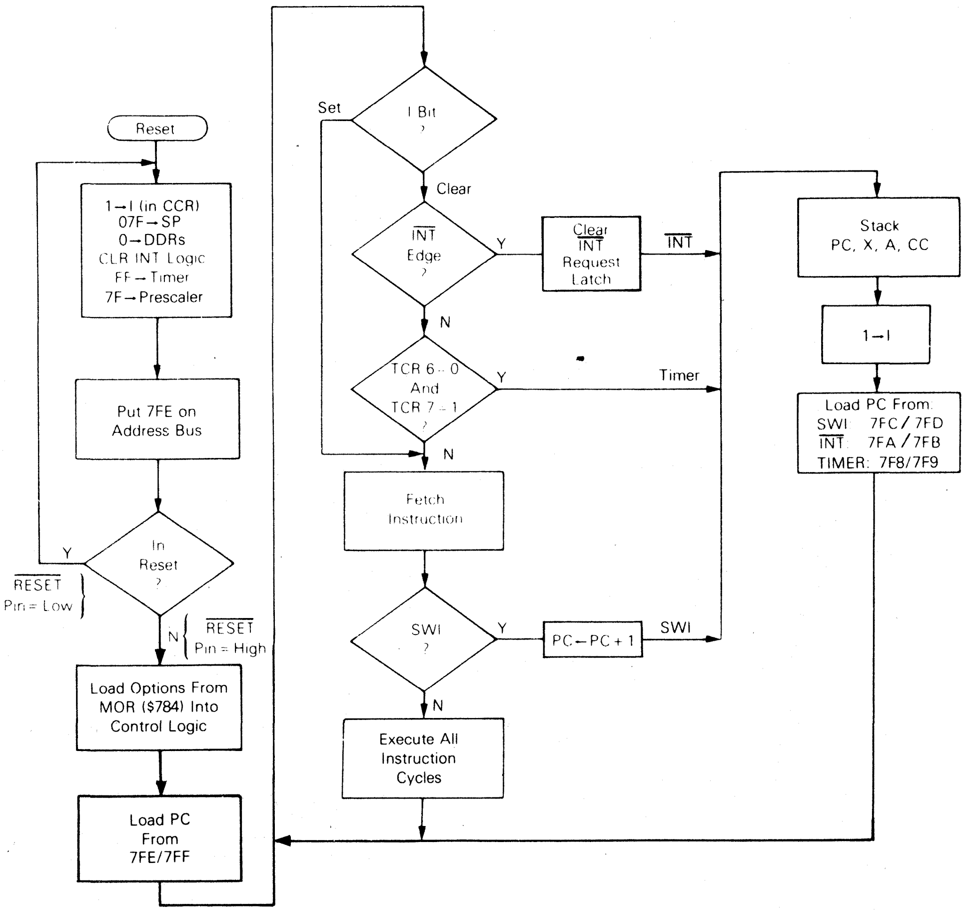
Figure 15. The interrupt service routine must end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt (by unstacking the previous CPU state). Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction 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 15 — RESET AND INTERRUPT PROCESSING FLOWCHART



(f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 16a, 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 $\overline{\text{INT}}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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)

$$t_{cyc} \times 2 + 250\text{ ns} = \text{period} = \frac{1}{\text{freq}}$$

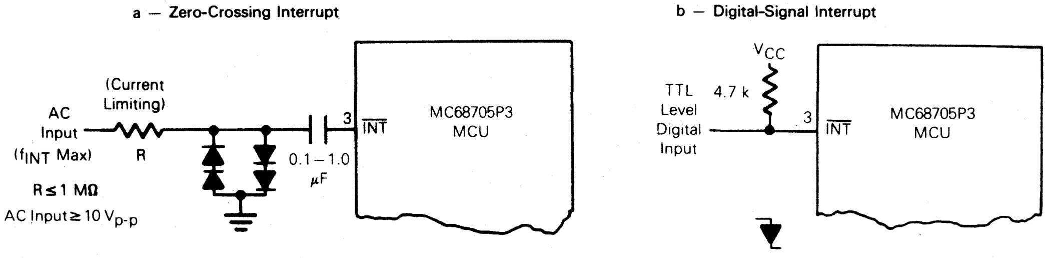
The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See figure 16(b). For the $\overline{\text{INT}}$ function, the maximum allowable frequency is also determined by the software response of the $\overline{\text{INT}}$ 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

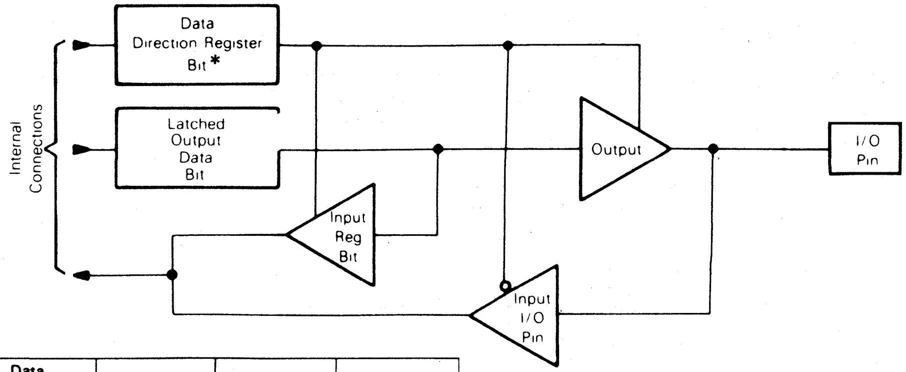
There are 20 input/output pins. The $\overline{\text{INT}}$ pin may be polled with branch instructions to provide an additional input pin. All pins on Ports A, B, and C are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). The port I/O 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 17. When Port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

FIGURE 16 — TYPICAL INTERRUPT CIRCUITS


a — Zero-Crossing Interrupt

b — Digital-Signal Interrupt

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	High-Z**	Pin

*DDR is 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.

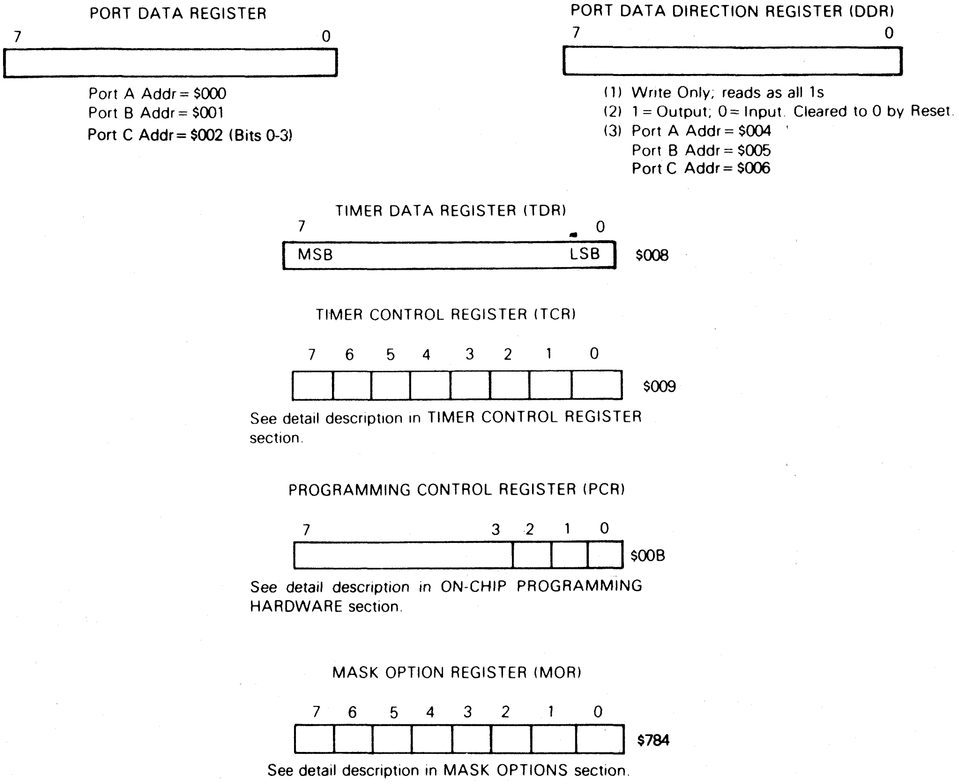
All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs while port B and C lines are CMOS compatible as inputs. The memory map in Figure 6 gives the address of data registers and DDRs. The Register configuration is provided in Figure 18. Figure 19 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 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").

FIGURE 18 — MCU REGISTER CONFIGURATION



Port A Addr = $000
Port B Addr = $001
Port C Addr = $002 (Bits 0-3)

(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

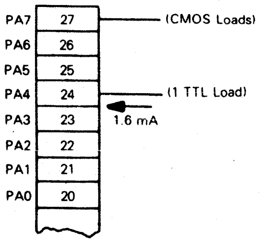
See detail description in TIMER CONTROL REGISTER section.

See detail description in ON-CHIP PROGRAMMING HARDWARE section.

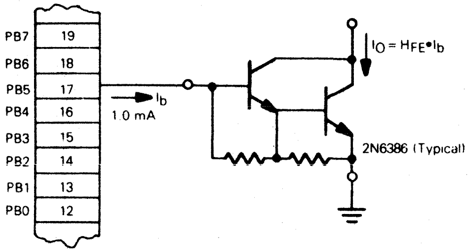
See detail description in MASK OPTIONS section.

FIGURE 19 — TYPICAL PORT CONNECTIONS

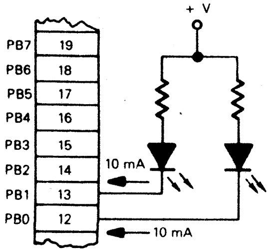
(a) Output Modes



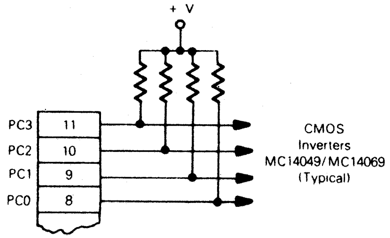
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 Driving one TTL Load Directly (using CMOS output option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

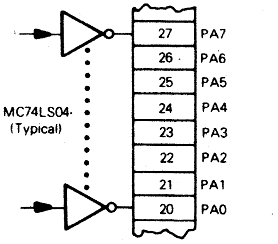


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

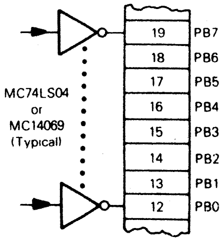


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors.

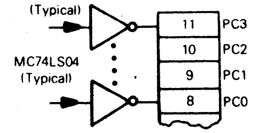
(b) Input Modes



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 9 and TIMER section).

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	$\overline{\text{VPON}}$	$\overline{\text{PGE}}$	$\overline{\text{PLE}}$	Program Control Register $00B

b0, $\overline{\text{PLE}}$ 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)
$\overline{\text{PLE}}$ is set during a Reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{\text{VPON}}$ is a logic "1".

b1, $\overline{\text{PGE}}$ Program Enable—When cleared, $\overline{\text{PGE}}$ enables programming of the EPROM. $\overline{\text{PGE}}$ can only be cleared if $\overline{\text{PLE}}$ is cleared. $\overline{\text{PGE}}$ 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 $\overline{\text{PLE}}$ is low)
$\overline{\text{PGE}}$ is set during a Reset; however, it has no effect on EPROM circuits if $\overline{\text{VPON}}$ is a logic "1".

b2, $\overline{\text{VPON}}$ (Vpp ON) — $\overline{\text{VPON}}$ 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

$\overline{\text{VPON}}$ 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:

$\overline{\text{VPON}}$	$\overline{\text{PGE}}$	$\overline{\text{PLE}}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	0	1	Invalid state; $\overline{\text{PGE}}=0$ iff $\overline{\text{PLE}}=0$
1	0	1	Invalid state; $\overline{\text{PGE}}=0$ iff $\overline{\text{PLE}}=0$
0	1	1	"High voltage" on Vpp
1	1	1	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ 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 15 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.

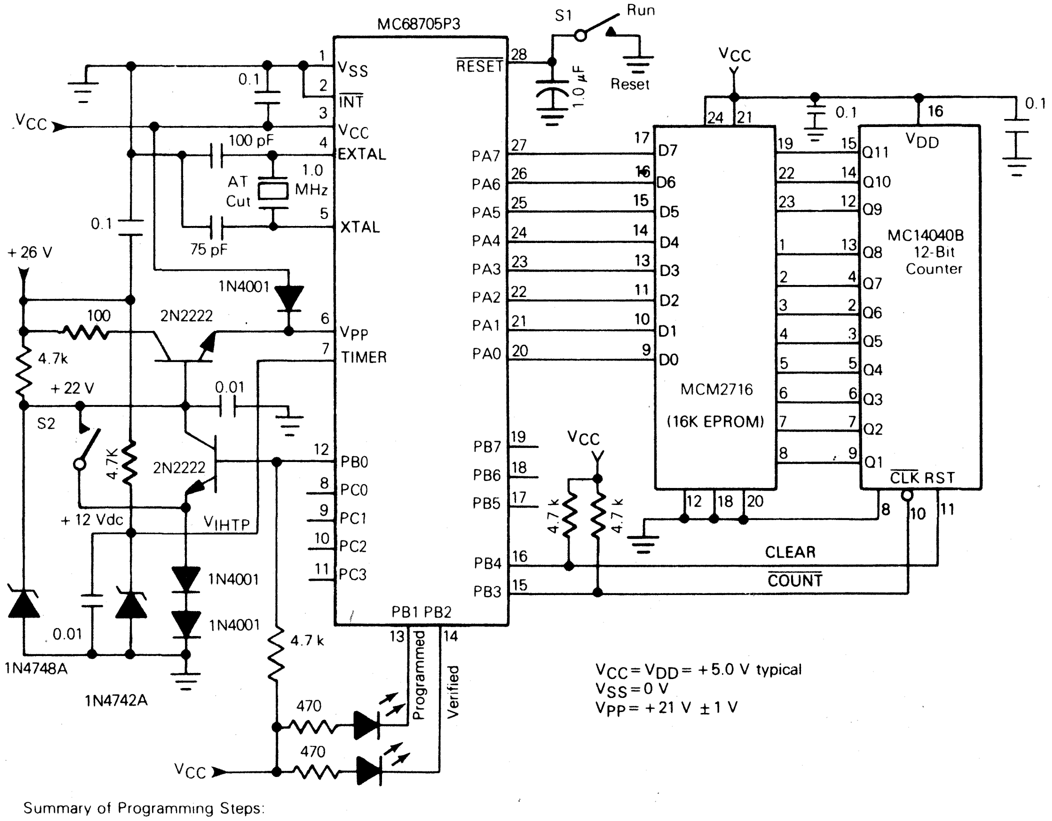
Caution

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 $\overline{\text{RESET}}$ pin is allowed to rise above V_{IRES+}. Figure 20 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 20 — 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 20):

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 $\overline{\text{RESET}}$ 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 ($\overline{\text{COUNT}}$). 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, $\overline{\text{COUNT}}$ 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.

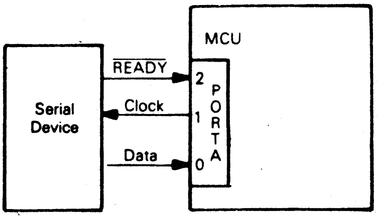
The operation of all other circuitry has been exactly duplicated or designed to function exactly the same in both devices including Interrupts, Timer, Data Ports, and Data Direction Registers (DDRs). A stated design goal has been to provide the user with a safe inexpensive way to verify his program and 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 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

FIGURE 21 — BIT MANIPULATION EXAMPLE



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

		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

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

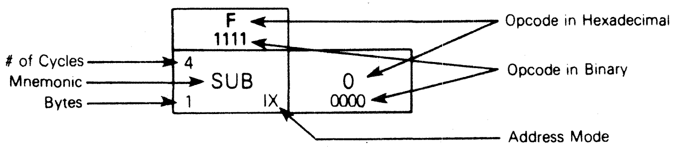
TABLE 7 — M6805 HMOS FAMILY INSTRUCTION SET OPCODE MAP

	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	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



M6805 FAMILY

	MC6805P2	MC6805P4	MC6805R2	MC6805T2	MC6805U2
Technology	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	28	40
On-Chip RAM (Bytes)	64	112	64	64	64
On-Chip User ROM (Bytes)	1.1K	1.1K	2K	2.5K	2K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	19	24
Unidirectional I/O Lines	None	None	6 Inputs	None	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, PLL	Timer
External Interrupt Inputs	1	1	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705R3	None	MC68705U3
STOP and WAIT	No	No	No	No	No

	MC68075P3	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2
Technology	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS
Number of Pins	28	40	40	40	28	40
On-Chip RAM (Bytes)	112	112	112	112	64	112
On-Chip User ROM (Bytes)	1.8K EPROM	3.8K EPROM	3.8K EPROM	None	1K	2K
External Bus	None	None	None	Yes	None	None
Bidirectional I/O Lines	20	24	24	16	16	32
Unidirectional I/O Lines	None	6 Inputs	8 Inputs	None	4 Inputs	None
Other I/O Features	Timer	Timer, A/D	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	2	2	1	1	1
EPROM Version	–	–	–	None	None	None
STOP and WAIT	No	No	No	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE

M6805
A FAMILY OF MICROCOMPUTERS
MC6805P2P
A PART NUMBER
A PARTICULAR M6805 FAMILY VERSION
A PLASTIC PACKAGE
MC68705P3
AN EPROM VERSION
MC146805G2
A CMOS M6805 FAMILY MEMBER
MC6805P2P2
AN EVALUATION PROGRAM STORED IN ROM
(PLASTIC PACKAGE)

MOTOROLA

MC68705R3

Advance Information

8-BIT EPROM MICROCOMPUTER UNIT WITH A/D

The MC68705R3 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, A/D Converter, and a TIMER.

Because of these features, the MC68705R3 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.

A comparison table of the key features for several members of the M6805 Family is shown on the last page of this data sheet.

HARDWARE FEATURES

- 8-Bit Architecture
- 112 Bytes of RAM
- Memory Mapped I/O
- 3776 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
- 4 Vectored Interrupts — External (2), Timer (1), and Software (1)
- Zero-Cross Detection on $\overline{\text{INT}}$ Input
- 24 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- 8 Digital Input Lines
- A/D Converter
  - 8-Bit Conversion, Monotonic
  - 4 Multiplexed Analog Inputs
  - ± ½ LSB Quantitizing Error
  - ± ½ LSB All Other Errors
  - ± 1 LSB Total Error (Max)
  - Ratiometric Conversion
- On-Chip Clock Generator
- Master Reset
- Complete Development System Support on EXORciser
- 5 V Single Supply
- Emulates the MC6805R2
- Bootstrap Program in ROM Simplifies EPROM Programming

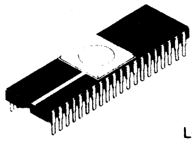
SOFTWARE FEATURES

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Registers
- 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

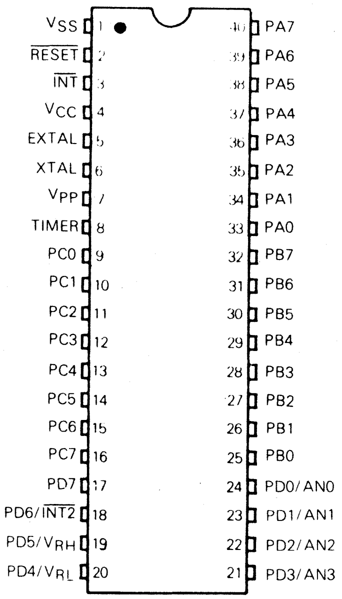
(HIGH-DENSITY, N-CHANNEL
DEPLETION LOAD,
5 V EPROM PROCESS)

8-BIT EPROM
MICROCOMPUTER WITH A/D



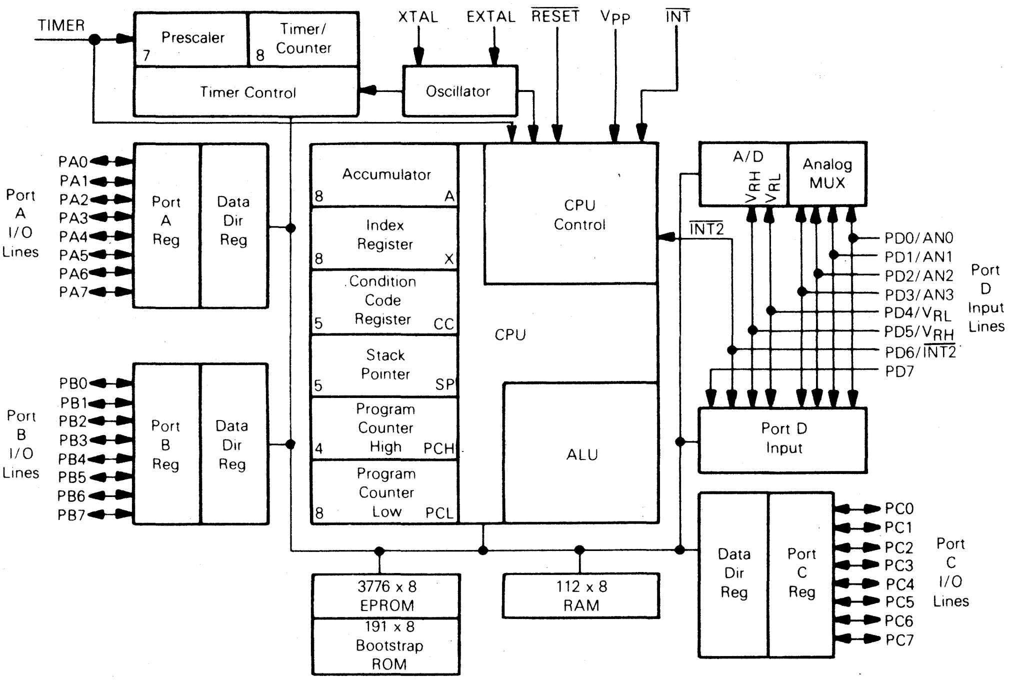
L SUFFIX
CERAMIC PACKAGE
CASE 715

FIGURE 1 — PIN ASSIGNMENTS



ADI-858-R1

FIGURE 2 — MC68705R3 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 (VPP Pin)	V_{PP}	−0.3 to +22.0	V
TIMER Pin			
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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. 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:

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

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:

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

Solving equations 1 and 2 for K gives:

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

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}	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

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}	–	–	5.0	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	FF	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.

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
$\overline{\text{INT}}$, $\overline{\text{INT2}}$, or Timer Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	t_{cyc} + 250	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	t_{cyc} + 250	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Cap = 1.0 μF)	t_{RHL}	100	–	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (for ±5° Accuracy)	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	V_{IH}				
$\overline{\text{RESET}}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	–	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} − 0.5	–	V_{CC}	
$\overline{\text{INT}}$ (4.75 ≤ V_{CC} ≤ 5.75)		4.0	**	V_{CC}	V
(V_{CC} < 4.75)		V_{CC} − 0.5	**	V_{CC}	
All Other		2.0	–	V_{CC}	V
Input High Voltage (TIMER Pin)	V_{IH}				
Timer Mode		2.0	–	V_{CC}	V
Bootstrap Programming Mode		9.0	12.0	15.0	V
Input Low Voltage	V_{IL}				
$\overline{\text{RESET}}$		−0.3	–	0.8	V
$\overline{\text{INT}}$		−0.3	**	1.5	V
All Other		−0.3	–	0.8	V
$\overline{\text{INT}}$ 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)	P_{INT}	–	600	TBD	mW
Input Capacitance	C_{in}				
EXTAL		–	25	–	pF
All Other (See Note)		–	10	–	pF
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figure 11)					
Out of Reset Voltage	V_{IRES+}	2.1	–	4.0	V
Into Reset Voltage	V_{IRES-}	0.8	–	2.0	V
Programming Voltage (V_{PP} Pin)	V_{PP}*				
Programming EPROM		20.0	21.0	22.0	V
Operating Mode		4.0	V_{CC}	5.75	V
Input Current	I_{in}				μA
TIMER (V_{in} = 0.4 V)		–	–	20	
$\overline{\text{INT}}$ (V_{in} = 0.4 V)		–	20	50	
EXTAL (V_{in} = 2.4 V to V_{CC})		–	–	10	
(V_{in} = 0.4 V)		–	–	−1600	
$\overline{\text{RESET}}$ (V_{in} = 0.8 V) (External Capacitor Changing Current)		−4.0	–	−50	

*V_{PP} is pin 7 on the MC68705R3 and is connected to V_{CC} in the normal operating mode. In the MC6805R2, 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 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.

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}	3.5	–	–	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}	−0.3	–	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}	−0.3	–	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}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	µA
Port D (Input Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input Current*	I_{in}	–	–	20	µA

*The A/D conversion resistor (15 kΩ typical) is connected internally between PD5/V_{RH} and PD4/V_{RL}.

FIGURE 3 — TTL EQUIVALENT TEST LOAD
(PORT B)

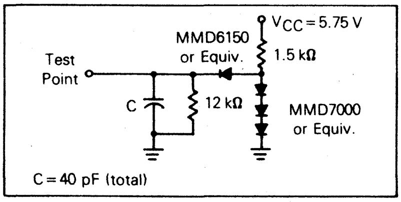


FIGURE 4 — CMOS EQUIVALENT TEST LOAD
(PORT A)

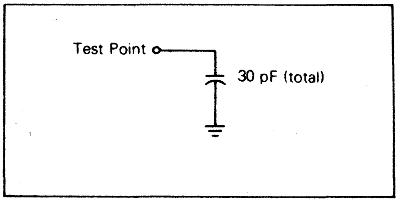
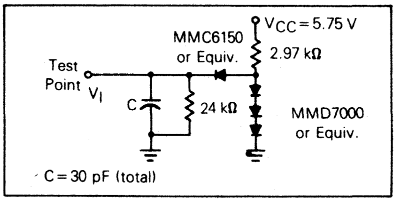


FIGURE 5 — TTL EQUIVALENT TEST LOAD
(PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$

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 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 CLK bit (see Mask Options section), 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 section for recommendations about these inputs.

TIMER

This pin is used as an external input to control the internal timer/counter circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program (see Programming Firmware section). Refer to Timer section for additional information about the timer circuitry.

$\overline{\text{RESET}}$

This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ low. Refer to Resets section 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 section and Electrical Characteristics section.

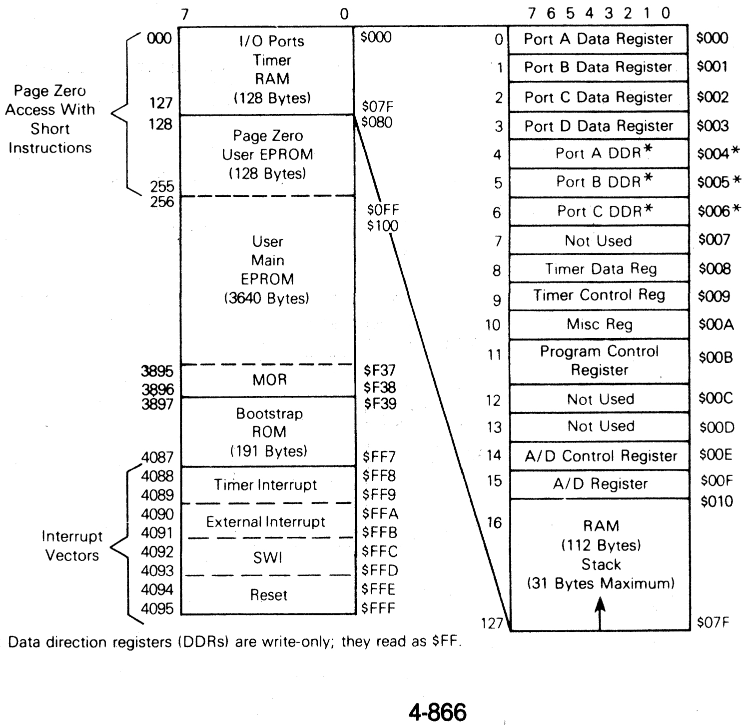
INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7)

These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs under software control of the data direction register (DDRs). Port D has up to four analog inputs, plus two voltage reference inputs when the A/D is used (PD5/V_{RH}, PD4/V_{RL}), an $\overline{\text{INT2}}$ input, and from one to eight digital inputs. All port D lines can be directly read 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 Input/Output, A/D Converter, and Interrupts sections for additional information.

MEMORY

As shown in Figure 6, the MCU is capable of addressing 4096 bytes of memory and I/O registers with its program

FIGURE 6 — MCU MEMORY MAP

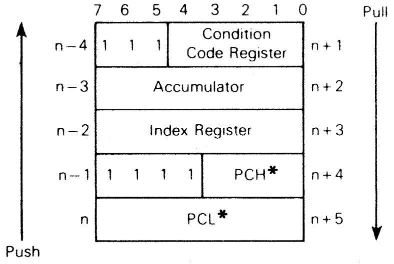


*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

counter. The MCU has implemented 4093 bytes of these locations. This consists of: 3776 bytes of user EPROM, 191 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM mask option register (MOR), a program control register (PCR), seven bytes of I/O, two timer registers, a miscellaneous register, and two A/D registers. The user EPROM is located in two areas. The main EPROM area is memory locations \$080 to \$F37. The second area is reserved for eight interrupt/reset vector bytes at memory locations \$FF8 to \$FFF. The MCU uses 13 of the lowest 16 memory locations for program control and I/O features such as ports, the port DDRs, the timer, and A/D registers. The mask option register at memory location \$F38 completes the total. The 112 bytes of user RAM include up to 31 bytes for the stack.

The shared stack area is used during the processing of interrupt and subroutine calls. The contents of the CPU registers are pushed onto the stack in the order shown in Figure 7. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the higher order four 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.

FIGURE 7 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH 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 CPU has five registers available to the programmer. They are shown in Figure 8 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 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 a 12-bit register that contains the address of the next instruction to be executed.

STACK POINTER (SP)

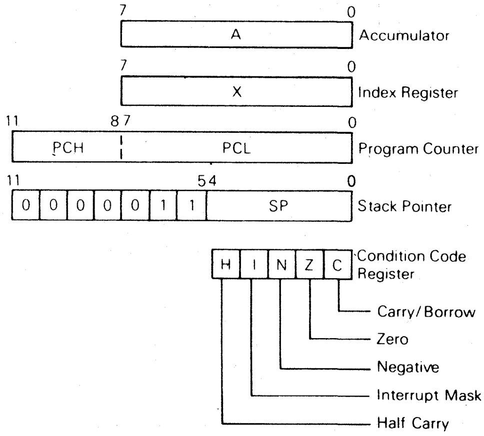
The stack pointer is a 12-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location \$07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is then pulled from the stack. The seven most significant bits of the stack point 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).

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.

FIGURE 8 — PROGRAMMING MODEL



INTERRUPT (I) — When this bit is set the timer and external interrupt ($\overline{\text{INT}}$) 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 MCU timer consists of an 8-bit software programmable counter which is driven by a 7-bit prescaler with 1-of-8 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. 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 9. 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 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 register.

The timer interrupt and $\overline{\text{INT2}}$ 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) 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.

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 or $\overline{\text{INT}}$ 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 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 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 b3 of the TCR is written to a logic "1"; however, TCR bit b3 always reads as a logic "0" 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.

Note that the timer block diagram in Figure 9 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, bit b3 is write-only (reads as logic "0"), and the other five 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 MC6805R2 mask ROM version. In the following paragraphs refer to Figure 9 as well as the Timer Control Register and Mask Options sections.

The TOPT (time 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). Refer to Timer Control Register section.

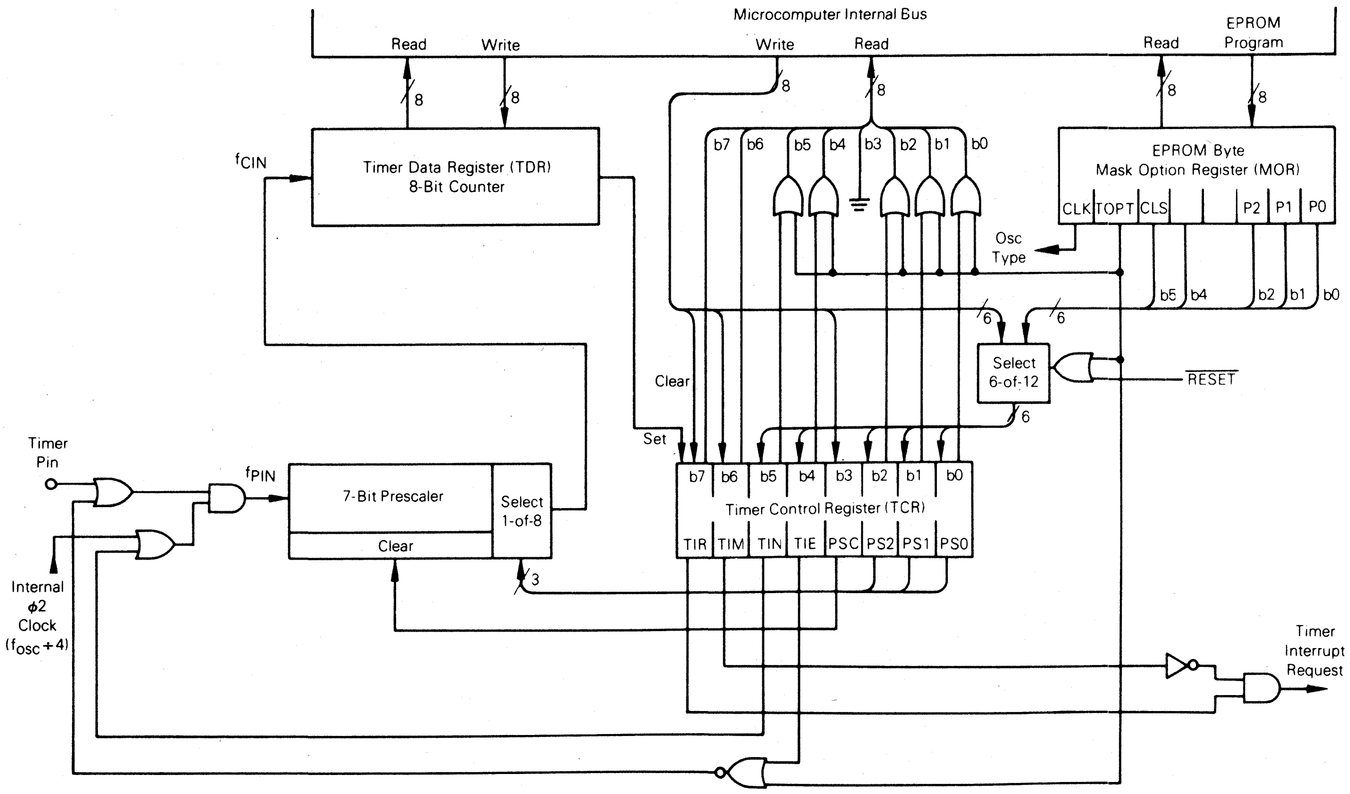
When the TIE and TIN bits are programmed to "0", the timer input is from the internal clock ($\phi 2$) and TIMER input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

When TIE = 1 and TIN = 0, the internal clock and the TIMER input pin signals are ANDed to form the timer input. 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.

FIGURE 9 — MC68705R3 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 EPROM byte.

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 a write operation to TCR, b3 (when bit 3 of the written data equals "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 MC6805R2 mask-programmable prescaler. 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, $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 logical "1" in the MOR controlled mode. (When read by software, these five 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. Bit b3 of the TCR, in the MOR controlled mode, always reads as a logical "0" and can be written to a logical "1" to clear the prescaler. 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.

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 10, typically provides sufficient delay.

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 11. See Figure 15 under Interrupts section for the complete reset sequence.

FIGURE 10 — POWER-UP RESET DELAY CIRCUIT

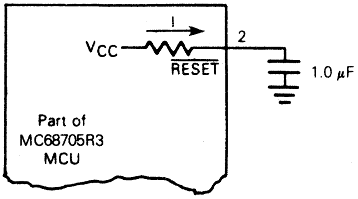
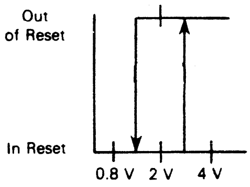


FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS



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 12. Crystal specifications and suggested PC board layouts are given in Figure 13. A resistor selection graph is given in Figure 14.

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.

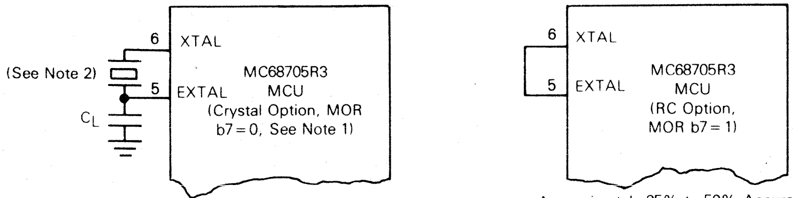
BOOTSTRAP ROM

The bootstrap ROM contains a factory program which allows the MCU to fetch data from an external device and transfer it into the MC68705R3 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)

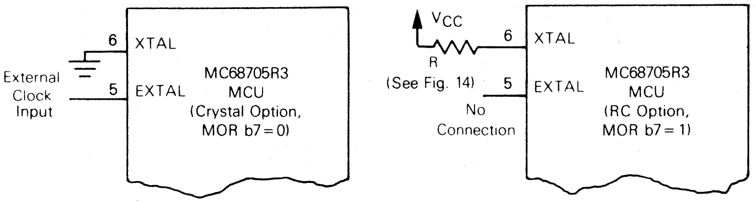
The mask option register is an 8-bit user programmed (EPROM) register in which six of the bits are used. Bits in this register are used to select the type of system clock, the timer option, the timer/prescaler clock source, and the prescaler option. It is fully described in the Mask Options section.

FIGURE 12 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% to 50% Accuracy
Typical $t_{cyc} = 1.25\ \mu s$
External Jumper



External Clock

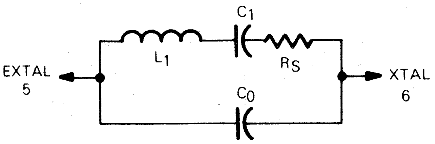
Approximately 10% to 25% Accuracy
External Resistor
(Excludes Resistor Tolerance)

NOTE:
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.

FIGURE 13 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT

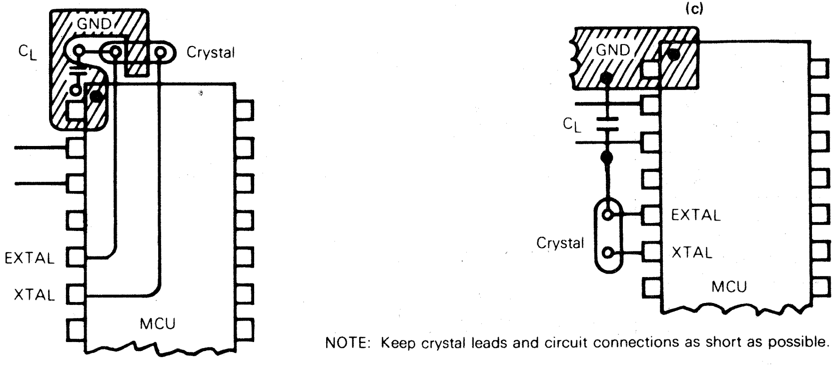
(a)

AT — Cut Parallel Resonance Crystal
$C_0 = 7$ pF Max.
Freq. = 4.0 MHz @ $C_L = 27$ pF
$R_S = 50$ ohms Max.



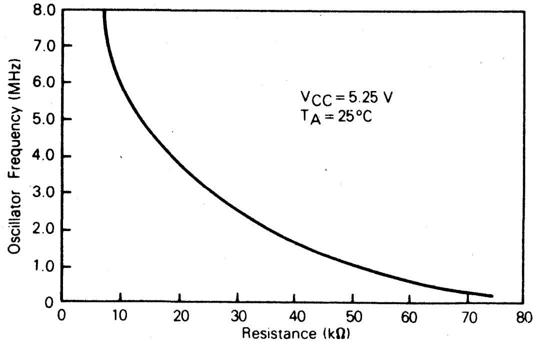
(b)

(c)



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

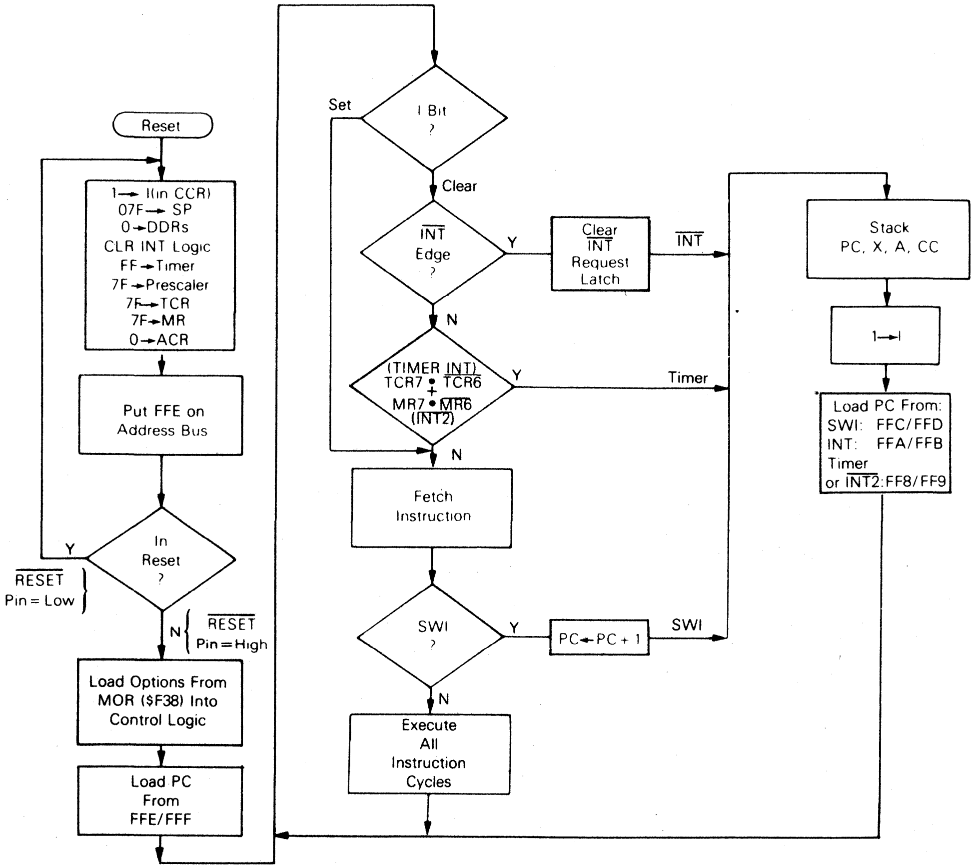
FIGURE 14 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MCU 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 15. 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

FIGURE 15 — RESET AND INTERRUPT PROCESSING FLOWCHART



$\overline{\text{RESET}}$, 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 "0" 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); refer to Figure 18. 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.

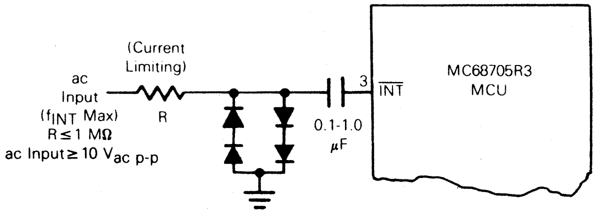
A sinusoidal input signal (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 16(a), for use as a zero-crossing detector. This allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip full wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provide a 2f clock. For digital applications the $\overline{\text{INT}}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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)

$$t_{cyc} \times 2 + 250 \text{ ns} = \frac{1}{\text{freq}}$$

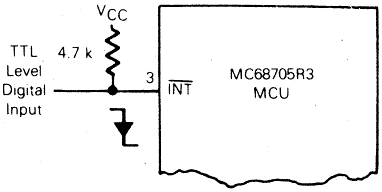
The period is not simply $t_{WL} + t_{WH}$. This computation is allowable, but it does reduce the maximum allowable frequency by defining an unnecessarily longer period (250 ns twice). See Figure 16(b).

FIGURE 16 — TYPICAL INTERRUPT CIRCUITS

(a) Zero-Crossing Interrupt



(b) Digital-Signal Interrupt



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 32 input/output pins. The $\overline{\text{INT}}$ 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 and should be initialized by software before changing the DDRs from input to output. When programmed as outputs, all output ports read latched output data, regardless of the logic levels at the output pin, due to output loading; refer to Figure 17.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs 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 multiplexed analog inputs, reference voltage, and $\overline{\text{INT2}}$. 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. 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} to V_{RH} recommended input voltage range.

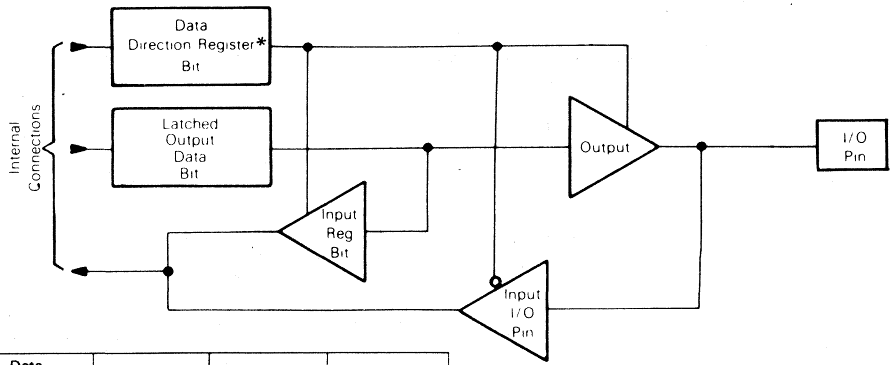
The memory map in Figure 6 gives the addresses of data registers and DDRs. The register configuration is provided in Figure 18. Figure 19 provides some example of port connections.

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 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.

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 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 ("0") and corresponds to the latched output data when the DDR is an output ("1").

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	Hi-Z **	Pin

* DDR is a write-only register and reads as all "1s."

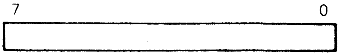
** Ports A (with CMOS drive disabled), B, and C are three-state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristics tables for complete information.

FIGURE 18 — MCU REGISTER CONFIGURATION

PORT DATA REGISTER

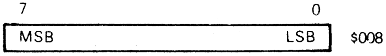
Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

PORT DATA DIRECTION REGISTER (DDR)

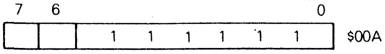


(1) Write Only; reads as all "1s"
(2) 1 = Output; 0 = Input. Cleared to 0 by reset.
(3) Port A Addr = $004
Port B Addr = $005
Port C Addr = $006

TIMER DATA REGISTER (TDR)

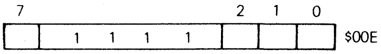


MISCELLANEOUS REGISTER (MR)



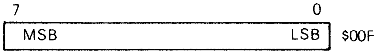
Bit 7—$\overline{\text{INT2}}$ Interrupt Request Bit: Set when falling edge detected on $\overline{\text{INT2}}$ pin; must be cleared by software. Cleared by 0 to reset.
Bit 6—$\overline{\text{INT2}}$ Interrupt Mask Bit: 1 = $\overline{\text{INT2}}$ interrupt masked (disabled). Set to 1 by reset.
Bits 5, 4, 3, 2, 1, 0—Read as "1s"—unused bits.

A/D CONTROL REGISTER (ACR)

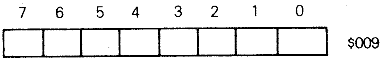


Bit 7—Conversion Complete Status Flag Set when conversion is complete; cleared only on a write to ACR, readable, not writable.
Bit 2, 1, 0—A/D Input Mux Selection (see Table 2)
Bits 6, 5, 4, 3—read as "1s"—unused bits

A/D RESULT REGISTER (ARR)

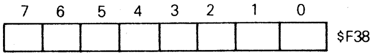


TIMER CONTROL REGISTER (TCR)



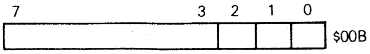
See detail description in Timer Control Register section.

MASK OPTION REGISTER (MOR)



See detail description in Mask Options section.

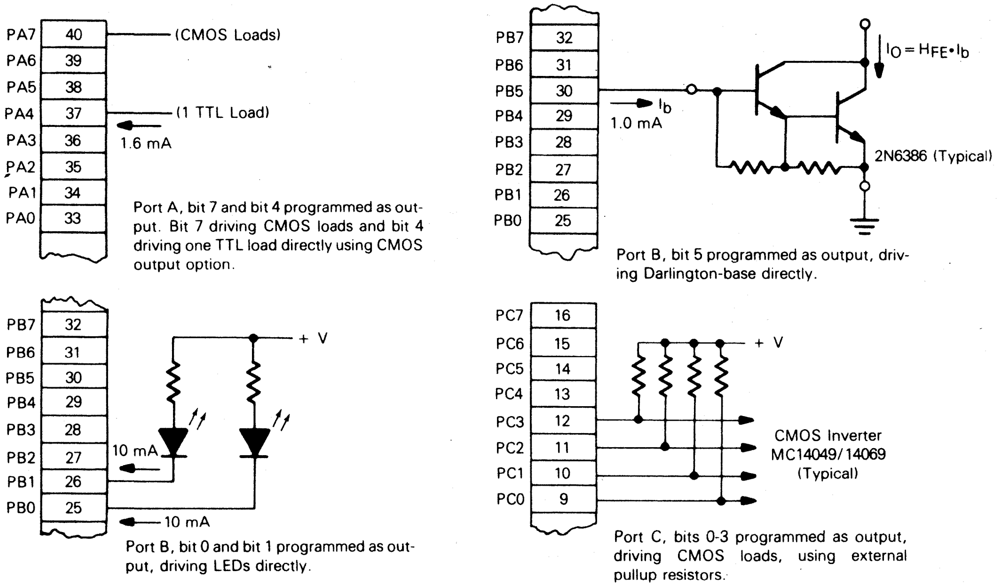
PROGRAMMING CONTROL REGISTER (PCR)



See detail description in On-Chip Programming Hardware section.

FIGURE 19 — TYPICAL PORT CONNECTIONS

(a) Output Modes



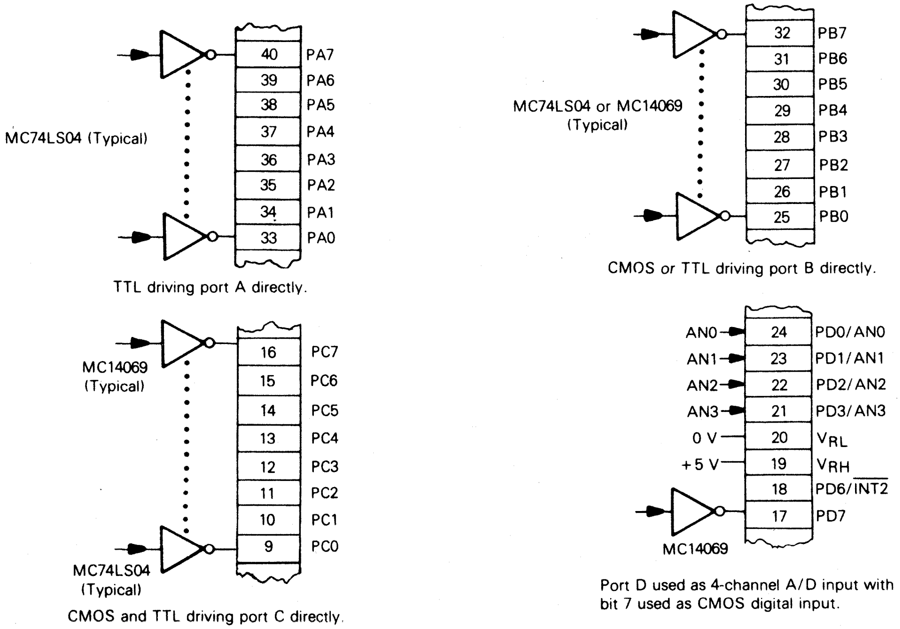
Port A, bit 7 and bit 4 programmed as output. Bit 7 driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.

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.

(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 with bit 7 used as CMOS digital input.

ANALOG-TO-DIGITAL CONVERTER (A/D)

The MCU has an 8-bit A/D converter implemented on the chip using a successive approximation technique, as shown in Figure 20. 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}, $V_{RH}/2$, $V_{RH}/4$, 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 1. This register is cleared during any reset condition. Refer to Figure 18 for register configuration.

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 and held internally.

The converter operates continuously using 30 machine cycles to complete a conversion of the sampled analog input. When conversion is complete, the digitized sample or 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. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

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,

FIGURE 20 — A/D BLOCK DIAGRAM

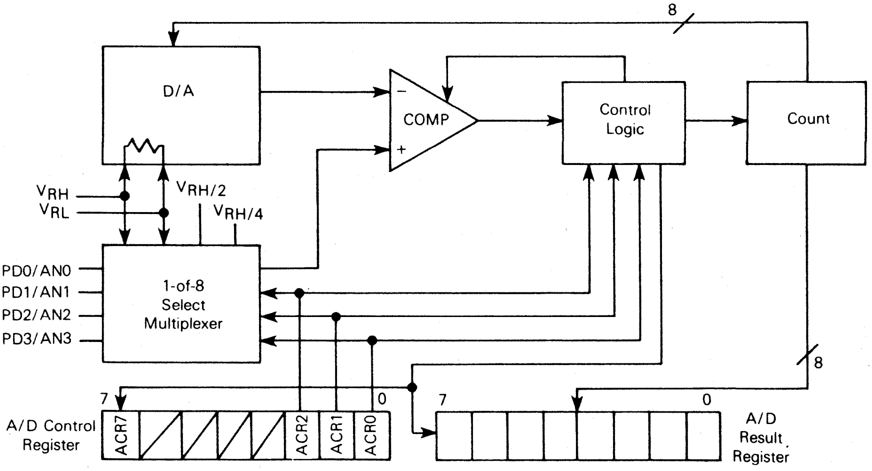


TABLE 1 — A/D INPUT MUX SELECTION

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	00
1	1	0	$V_{RH}/4$*	3F	40	41
1	1	1	$V_{RH}/2$*	7F	80	81

*Internal (Calibration) levels

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 9 and Timer section).

b7	b6	b5	b4	b3	b2	b1	b0
TIR	TIM	1	1	PSC*	1	1	1

Timer Control Register $009

TCR with MOR TOPT = 1 (MC6805R2 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 (8) or the internal $\phi2$.

1 = Selects the external clock source.

0 = Selects the internal $\phi2$ ($f_{OSC} \div 4$) 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 logical "1".

1 = Enables external TIMER pin.

0 = Disables external TIMER pin.

TIN-TIE Modes

TIN	TIE	Clock
0	0	Internal Clock ($\phi2$)
0	1	Gated (AND) of External and Internal Clocks
1	0	No Clock
1	1	External Clock

b3, PSC Prescaler Clear — This a write-only bit. It reads as a logical zero so the BSET and BCLR on the TCR function correctly. Writing a "1" into PSC generates a pulse which clears 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 PS2-0 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 MC68705R3 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 MC6805R2, 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 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 MC6805R2 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 $F38

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 type timer/prescaler. All bits, except 3, 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 MC6805R2 mask options.

0 = All TCR bits are implemented as a software programmable timer. The state of bits 5, 4, 2, 1, and 0 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 = Internal $\phi 2$

b4 Not used if MOR TOPT = 1 (MC6805R2 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 MC6805R2 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 MC68705R3 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 MC68705R3 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	$\overline{\text{VPON}}$	$\overline{\text{PGE}}$	$\overline{\text{PLE}}$	Programming Control Register \$00B

b0, $\overline{\text{PLE}}$ 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)

$\overline{\text{PLE}}$ is set during a reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{\text{VPON}}$ is a logic "1".

b1, $\overline{\text{PGE}}$ Program Enable – When cleared, $\overline{\text{PGE}}$ enables programming of the EPROM. $\overline{\text{PGE}}$ can only be cleared if $\overline{\text{PLE}}$ is cleared. $\overline{\text{PGE}}$ 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 $\overline{\text{PLE}}$ is low)

$\overline{\text{PGE}}$ is set during a reset; however, it has no effect on EPROM circuits if $\overline{\text{VPON}}$ is a logic "1".

b2, $\overline{\text{VPON}}$ (V_{PP} ON) – $\overline{\text{VPON}}$ 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

$\overline{\text{VPON}}$ being "1" "disconnects" $\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ from the rest of the chip, preventing accidental clearing of these bits from affecting the normal operating mode.

Note

$\overline{\text{VPON}}$ 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.

$\overline{\text{VPON}}$	$\overline{\text{PGE}}$	$\overline{\text{PLE}}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	0	1	Invalid state; $\overline{\text{PGE}}$ = 0 iff $\overline{\text{PLE}}$ = 0
1	0	1	Invalid state; $\overline{\text{PGE}}$ = 0 iff $\overline{\text{PLE}}$ = 0
0	1	1	"High voltage" on V_{PP}
1	1	1	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system (operating mode)

ERASING THE EPROM

The MC68705R3 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 15 Ws/cm^2. The lamps should be used without shortwave filters and the MC68705R3 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC68705R3 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 MC68705R3 has 191 bytes of mask ROM containing a bootstrap program which can be used to program the MC68705R3 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) of the MC68705R3 and the $\overline{\text{RESET}}$ pin is allowed to rise above V_{IRES+}. Figure 21 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM in the MC68705R3.

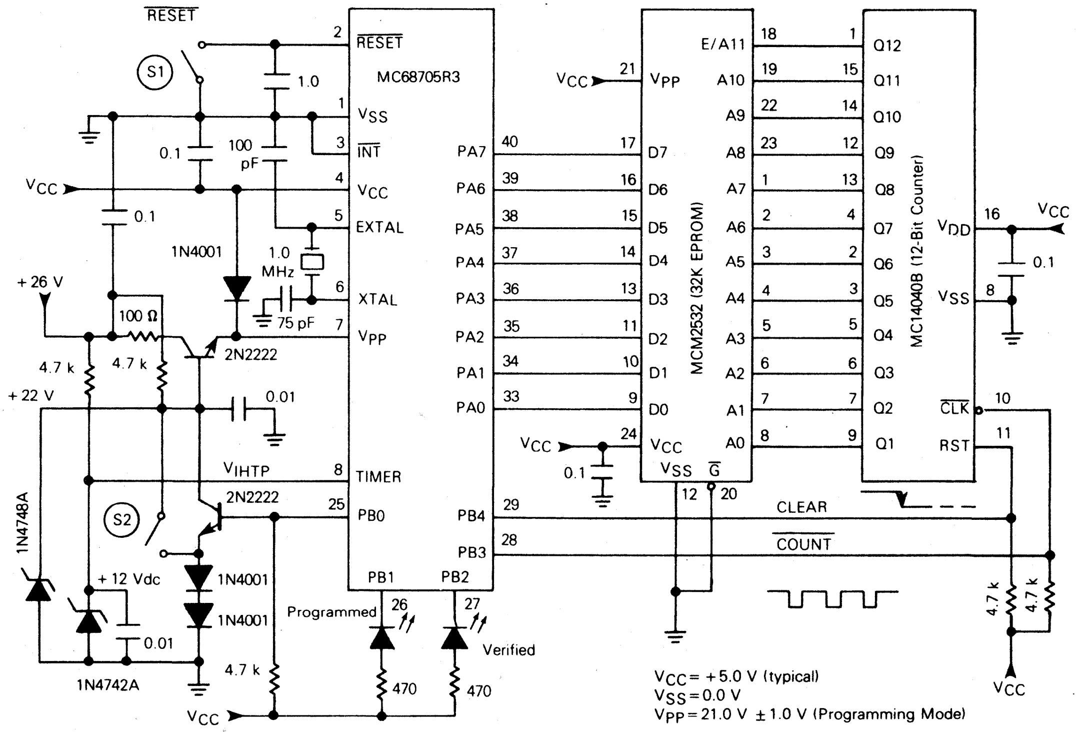
PROGRAMMING STEPS

The MCM2532 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705R3. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705R3 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 21):

Caution

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705R3 and MCM2532 into their respective sockets. This ensures that $\overline{\text{RESET}}$ is held low while inserting the devices.

FIGURE 21 — PROGRAMMING CONNECTIONS SCHEMATIC DIAGRAM



Summary of Programming Steps:

1. When plugging in the MC68705R3 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 +26 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705R3, first close S2 and then close S1. Disconnect V_{CC} and +26 V then remove the MC68705R3.

When ready to program the MC68705R3 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 12-bit counter (MC14040B) is clocked by the PB3 output ($\overline{\text{COUNT}}$). The counter selects the MCM2532 EPROM byte which is to load the equivalent MC68705R3 EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, $\overline{\text{COUNT}}$ clocks the counter to the next EPROM location. This continues until the MC68705R3 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 MC68705R3 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 MC68705R3 from its socket.

MC6805R2 EMULATION

The MC68705R3 emulates the MC6805R2 "exactly." MC6805R2 mask features are implemented in the mask option register (MOR) EPROM byte on the MC68705R3. There are a few minor exceptions to the exactness of emulation which are listed below.

1. The MC6805R2 "future ROM" area is implemented in the MC68705R3 and these 1728 bytes must be left unprogrammed to accurately simulate the MC6805R2. (The MC6805R2 reads all zeros from this area.)
2. The reserved ROM areas in the MC6805R2 and MC68705R3 have different data stored in them and this data is subject to change without notice. The MC6805R2 uses the reserved ROM for the self-check feature and the MC68705R3 uses this area for the bootstrap program.
3. The MC6805R2 reads all ones in its 48 byte "future RAM" area. This RAM is not implemented in the MC6805R2 mask ROM version, but is implemented in the MC68705R3.
4. The V_{PP} line (pin 7) in the MC68705R3 must be tied to V_{CC} for normal operation. In the MC6805R2, pin 7 is the NUM pin and is grounded in normal operation.
5. The LVI feature is not available in the MC68705R3. 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 MC6805R2 version. Therefore, the MC68705R3 does not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805R2 version, 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 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 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 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 22 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.

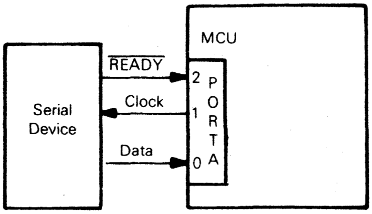
ADDRESSING MODES

The 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 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).

FIGURE 22 — BIT MANIPULATION EXAMPLE



```
                .
                .
                .
SELF   BRSET    2, PORTA, SELF
                .
                .
                .
       BSET     1, PORTA
       BRCLR    0, PORTA, CONT
CONT   BCLR     1; PORTA
       ASR      RAMLOC
                .
                .
                .
```

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 section.

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 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 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.

REGISTERS/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 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 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 3.

BRANCH INSTRUCTIONS – The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

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 and branch operations. Refer to Table 5.

CONTROL INSTRUCTIONS – The control instructions control the MCU operations during program execution. Refer to Table 6.

ALPHABETICAL LISTING – The complete instruction set is given in alphabetical order in Table 7.

OPCODE MAP SUMMARY – Table 8 is an opcode map for the instructions used on the MCU.

TABLE 2 — 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 3 — 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 4 — 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 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 5 — 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 6 — 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 7 — 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 7 — INSTRUCTION SET (CONTINUED)

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**
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

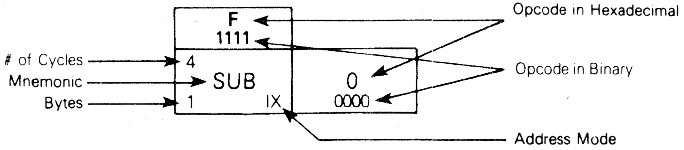
TABLE 8 — M68705 HMOS FAMILY INSTRUCTION SET OPCODE MAP

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	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

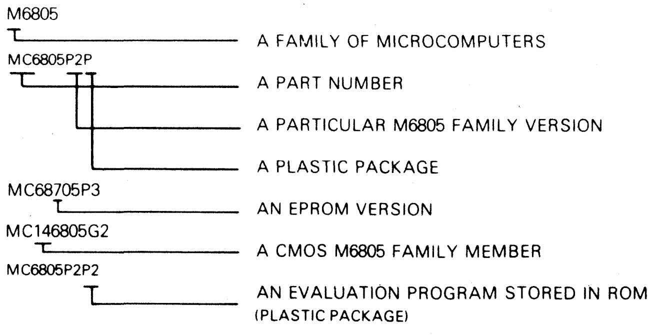


M6805 FAMILY

	MC6805P2	MC6805P4	MC6805R2	MC6805T2	MC6805U2
Technology	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	28	40
On-Chip RAM (Bytes)	64	112	64	64	64
On-Chip User ROM (Bytes)	1.1K	1.1K	2K	2.5K	2K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	19	24
Unidirectional I/O Lines	None	None	6 Inputs	None	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, PLL	Timer
External Interrupt Inputs	1	1	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705R3	None	MC68705U3
STOP and WAIT	No	No	No	No	No

	MC68075P3	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2
Technology	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS
Number of Pins	28	40	40	40	28	40
On-Chip RAM (Bytes)	112	112	112	112	64	112
On-Chip User ROM (Bytes)	1.8K EPROM	3.8K EPROM	3.8K EPROM	None	1K	2K
External Bus	None	None	None	Yes	None	None
Bidirectional I/O Lines	20	24	24	16	16	32
Unidirectional I/O Lines	None	6 Inputs	8 Inputs	None	4 Inputs	None
Other I/O Features	Timer	Timer, A/D	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	2	2	1	1	1
EPROM Version	–	–	–	None	None	None
STOP and WAIT	No	No	No	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE



MOTOROLA

MC68705U3

Advance Information

8-BIT EPROM MICROCOMPUTER UNIT

The MC68705U3 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 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 MC68705U3 offers the user an economical means of designing the M6805 Family MCU into his system, either as a prototype evaluation, as a low-volume production run, or a pilot production run.

A comparison of the key features for several members of the M6805 Family is shown on the last page of this data sheet.

HARDWARE FEATURES:

- 8-Bit Architecture
- 112 Bytes of RAM
- Memory Mapped I/O
- 3776 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
- 4 Vectored Interrupts — External (2), Timer (1), and Software (1)
- Zero-Cross Detection on $\overline{\text{INT}}$ Input
- 24 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- 8 Digital Input Lines
- On-Chip Clock Generator
- Master and Power-On Reset
- Complete Development System Support on EXORciser®
- 5 V Single Supply
- Emulates the MC6805U2
- Bootstrap Program in ROM Simplifies EPROM Programming

SOFTWARE

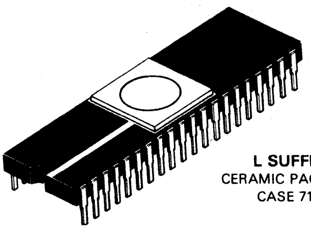
- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to EPROM, RAM, and I/O

® EXORciser is a registered trademark of Motorola Inc.

HMOS

(HIGH-DENSITY, N-CHANNEL
DEPLETION LOAD,
5 V EPROM PROCESS)

**8-BIT EPROM
MICROCOMPUTER**



L SUFFIX
CERAMIC PACKAGE
CASE 715

FIGURE 1 — PIN ASSIGNMENTS

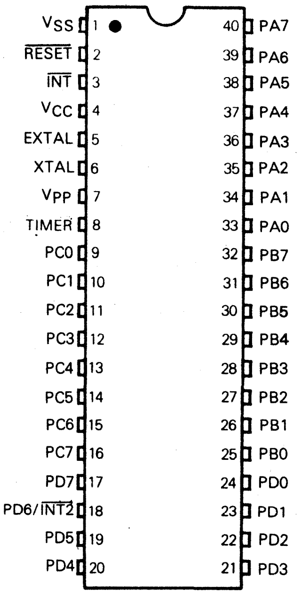
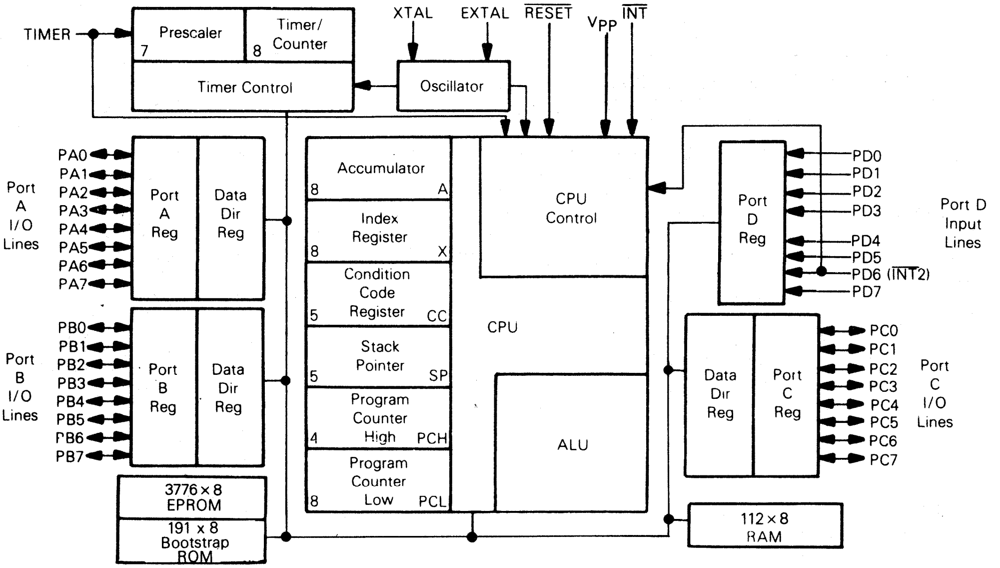


FIGURE 2 — MC68705U3 HMOS MICROCOMPUTER BLOCK DIAGRAM



MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage			
EPROM Programming Voltage (V_{PP} Pin)	V_{PP}	−0.3 to +22.0	V
TIMER Pin			
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 +50	°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 $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{CC}$. 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:

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

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:

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

Solving equations 1 and 2 for K gives:

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

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}=GND, 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	I_{PP}	–	–	8	mA
V_{PP}=21.0 V		–	–	30	
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

SWITCHING CHARACTERISTICS (V_{CC}= +5.25 Vdc ±0.5 V, V_{SS}=GND, T_A=0° to 50°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency					
Normal	f_{osc}	0.4	–	4.4	MHz
Instruction Cycle Time (4/f_{osc})	t_{cyc}	0.950	–	10	μs
$\overline{\text{INT}}$, $\overline{\text{INT2}}$, or Timer Pulse Width	t_{WL}, t_{WH}	t_{cyc}+250	–	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	t_{cyc}+250	–	–	ns
$\overline{\text{RESET}}$ Delay Time (External Cap=1.0 μF)	t_{RHL}	100	–	–	ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (for ±5° Accuracy)	f_{INT}	0.03	–	1.0	kHz
External Clock Duty Cycle (EXTAL) (See Figure 12)	–	40	50	60	%

DC ELECTRICAL CHARACTERISTICS (V_{CC}= +5.25 Vdc ±0.5 Vdc, V_{SS}=GND, T_A=0° to 50°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage					
$\overline{\text{RESET}}$ (4.99≤V_{CC}≤5.51)	V_{IH}	4.0	–	V_{CC}	V
(V_{CC}<4.75)		V_{CC}−0.5	–	V_{CC}	
$\overline{\text{INT}}$ (4.99≤V_{CC}≤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/BOOT Pin)					
Timer Mode	V_{IH}	2.0	–	V_{CC}	V
Bootstrap Programming Mode		9.0	12.0	15.0	
Input Low Voltage					
$\overline{\text{RESET}}$	V_{IL}	−0.3	–	0.8	V
$\overline{\text{INT}}$		−0.3	**	1.5	
All Other		−0.3	–	0.8	
Internal Power Dissipation (No Port Loading, V_{CC}=5.25 V, T_A=0°C)	P_{INT}	–	600	TBD	mW
Input Capacitance					
XTAL	C_{in}	–	25	–	pF
All Other		–	10	–	
INT Zero-Crossing Voltage, Through a Capacitor	V_{INT}	2.0		4.0	V_{ac} p-p
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figure 11)					
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		4.0	V_{CC}	5.75	
Input Current					
TIMER (V_{in}=0.4 V)	I_{in}	–	–	20	μA
$\overline{\text{INT}}$ (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	
$\overline{\text{RESET}}$ (V_{in}=0.8 V)		−4.0	–	−50	
(External Capacitor Changing Current)					

* V_{PP} is Pin 7 on the MC68705U3 and is connected to V_{CC} in the Normal Operating Mode. In the MC68705U2, 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 MC68705U2 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} = GND, T_A = 0° to 50°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}	3.5	–	–	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}	−0.3	–	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}	−0.3	–	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}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA
Port D (Input Only)					
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input Current	I_{in}	–	–	20	μA

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

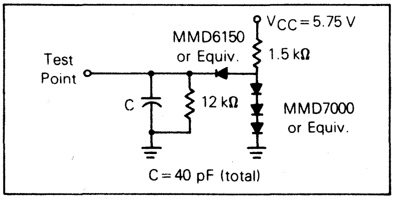


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

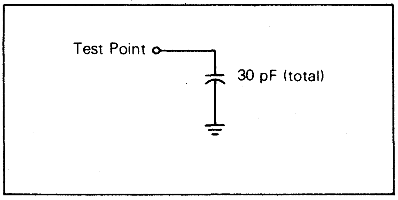
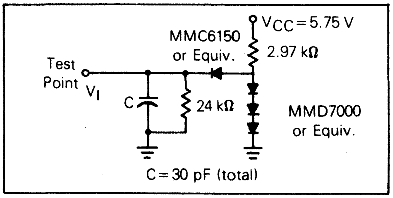


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$ — 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 pin is used as an external input to control the internal timer/counter circuitry. This pin also detects a higher voltage level used to initiate the bootstrap program (see PROGRAMMING FIRMWARE). Refer to TIMER under REGISTERS for additional information about the timer circuitry).

$\overline{\text{RESET}}$ — This pin has a Schmitt Trigger input and an on-chip pullup. The MCU can be reset by pulling $\overline{\text{RESET}}$ 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.

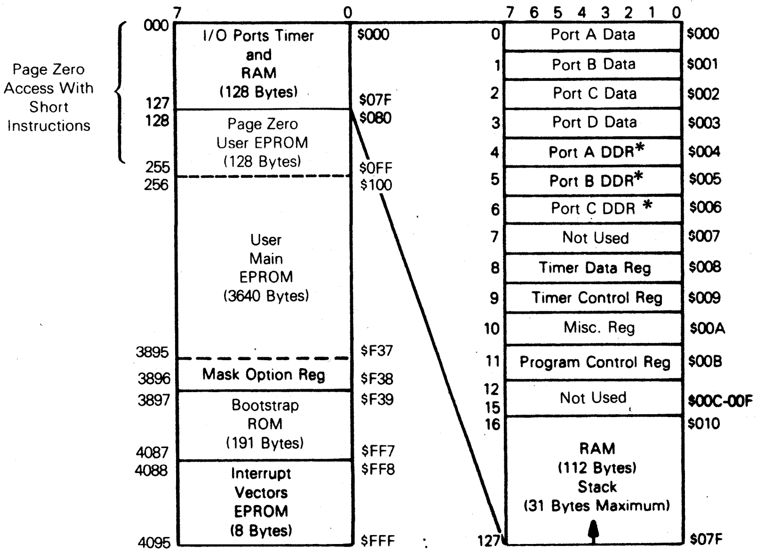
INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7) — These 32 lines are arranged into four 8-bit ports (A, B, C, and D). Ports A, B, and C are programmable as either inputs or outputs, under software control of the Data Direction Register (DDRs). Port D is for digital input only and PD6 may be used for a second interrupt, $\overline{\text{INT2}}$. Refer to INPUT/OUTPUT and INTERRUPTS paragraphs for additional information, being sure to observe the Caution.

MEMORY

As shown in Figure 6, the MCU is capable of addressing 4096 bytes of memory and I/O registers with its program counter. The MC68705U3 MCU has inplemented 4091 bytes of these locations. This consists of: 3776 bytes of user EPROM, 191 bytes of bootstrap ROM, 112 bytes of user RAM, an EPROM Mask Option Register (MOR), a Program Control Register (PCR), seven bytes of I/O, two Timer Registers, and a Miscellaneous Register. The user EPROM is located in two areas. The main EPROM area is memory locations $080 to $F37. The second area is reserved for eight interrupt/reset vector bytes at memory locations $FF8 to $FFF. The MCU uses 11 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 $F38 completes the total. The 112 bytes of user RAM include up to 31 bytes for the stack.

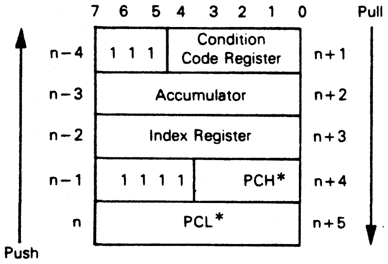
The stack area is used during the processing of interrupt and subroutine calls to save the processor state. The contents of the CPU registers are pushed onto the stack in the order shown in Figure 7. Since the Stack Pointer decrements during pushes, the low order byte (PCL) of the Program Counter is stacked first; then the higher order four 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.

FIGURE 6 — MC68705U3 MCU MEMORY CONFIGURATION



* CAUTION: Data Direction Registers (DDRs) are write-only; they read as $FF.

FIGURE 7 — 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 8 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 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 be used for data manipulations and 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. 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 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).

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.

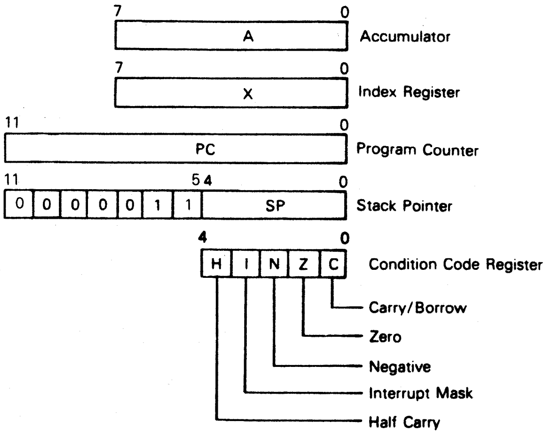
Interrupt (I) — When this bit is set the timer and external interrupt ($\overline{\text{INT}}$) are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

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 one).

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.

FIGURE 8 — PROGRAMMING MODEL



TIMER

The MC68705U3 MCU timer consists of an 8-bit software programmable counter which is driven by a 7-bit prescaler with an 1-of-8 selectable output. 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 9. 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 $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 register.

The timer interrupt and $\overline{\text{INT}}2$ 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) 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.

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. 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. TCR bit b3 always reads as a logic 0 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 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 9 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, bit b3 is write-only, (reads as a logic "0"), and the other 5 have no effect on a write and read as logic "1s". The 2 configurations provide the user with the capability to freely select timer options as well as accurately emulate the MC6805R2 mask ROM version. In the following paragraphs refer to Figure 9 as well as the 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). Refer to TIMER CONTROL REGISTER section.

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. This mode can be used to measure external pulse widths. The external pulse simply gates to 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.

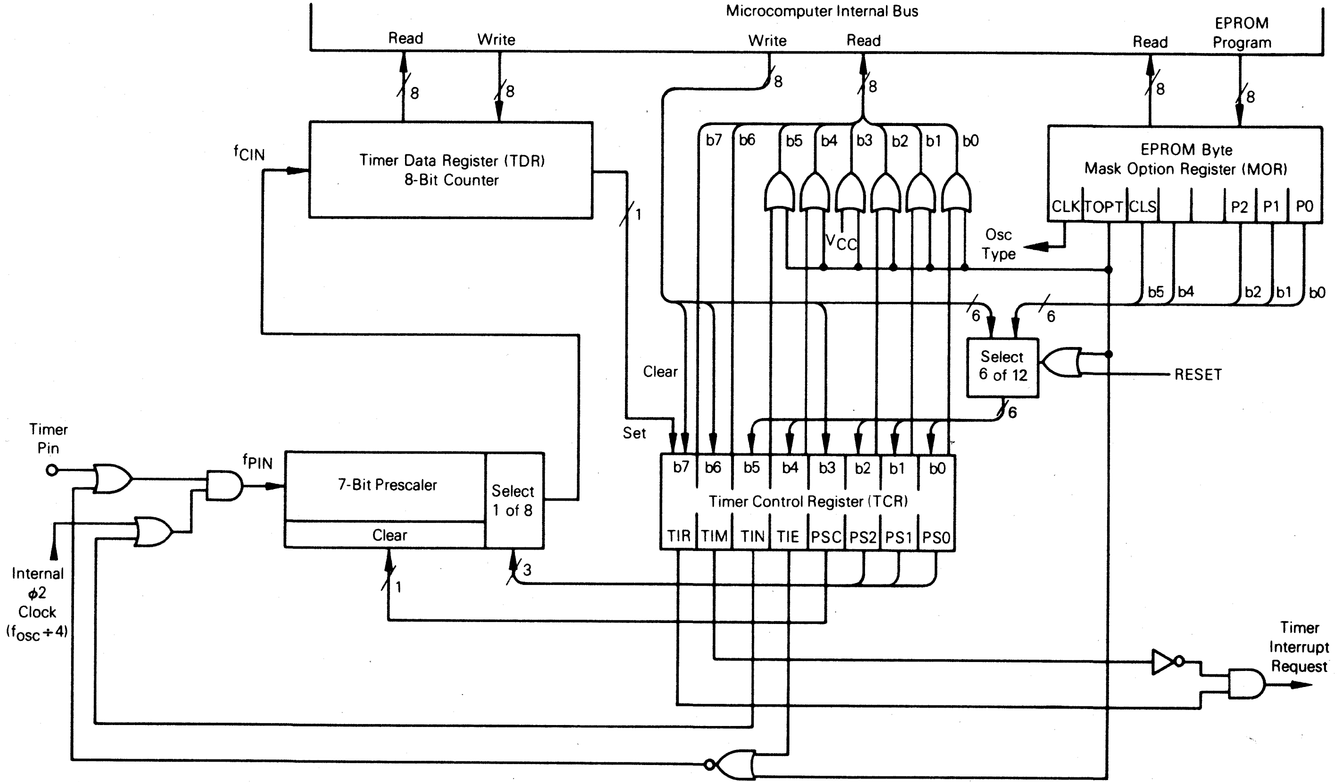
When TIE and TIN are both programmed to a "1", the timer is from the external clock. The external clock can be used to count external events as well as provide an external frequency for generating periodic interrupts.

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 a write operation to TCR, b3 (when bit 3 of the written data equals "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 MC6805R2 mask-programmable prescaler. 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, $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 logical "1" in the MOR controlled mode. (When read by software, these five 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. Bit b3 of the TCR, in the MOR controlled mode, always reads as a logical "0" and can be written to a logical "1" to clear the prescaler. 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.

FIGURE 9 — MC6805U3 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 ports via the MOR PROM byte.

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 10, typically provides sufficient delay.

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 11. See Figure 15 under INTERRUPTS for the complete reset sequence.

FIGURE 10 — POWER-UP RESET DELAY CIRCUIT

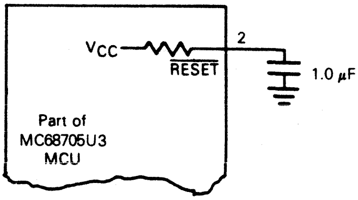
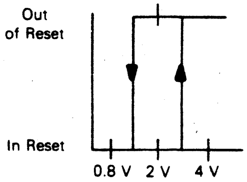


FIGURE 11 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

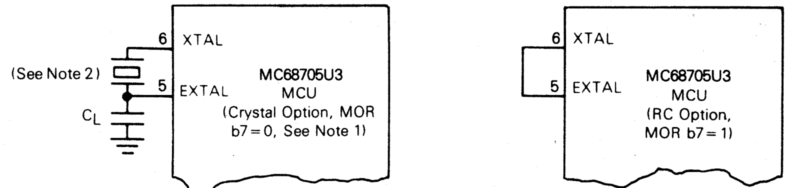


INTERNAL CLOCK GENERATOR OPTIONS

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.

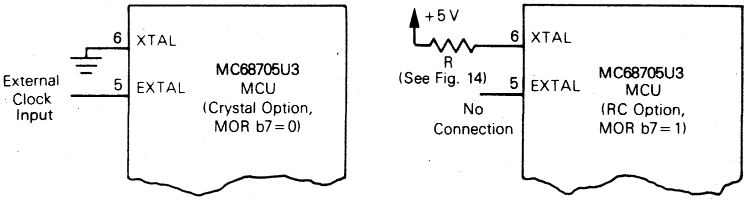
The different connection methods are shown in Figure 12.

FIGURE 12 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% Accuracy
Typical t_{CYC} = 1.25 µs
External Jumper



External Clock

Approximately 10% Accuracy
(Excludes Resistor Tolerance)
External Resistor

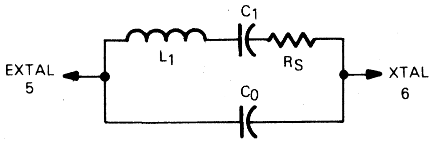
NOTE: 1. When the TIMER input pin is in the V_{IHTP} range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V_{CC}, the clock generator option is determined by bit 7 of the Mask Option Register (CLK).

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.

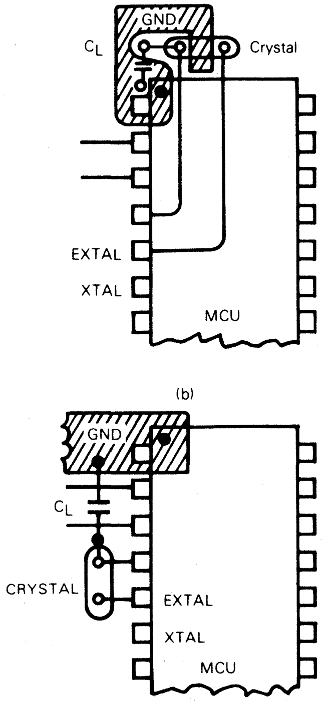
Crystal specifications are given in Figure 13. A resistor selection graph is given in Figure 14.

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.

FIGURE 13 — CRYSTAL MOTIONAL-ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



AT — Cut Parallel Resonance Crystal
$C_0 = 7$ pF Max.
FREQ = 4.0 MHz @ $C_L = 27$ pF
$R_S = 100$ ohms Max.

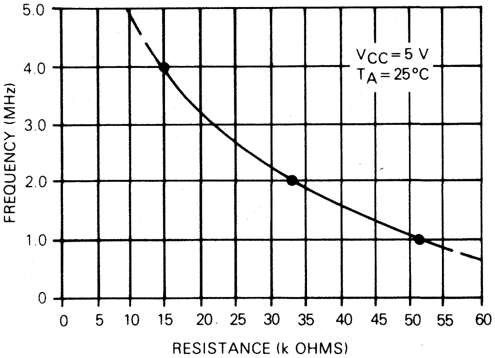


(a)

(b)

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

FIGURE 14 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



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 MC68705U3 EPROM. The bootstrap program provides: timing of programming pulses, timing of V_{PP} input, and verification after programming. See PROGRAMMING FIRMWARE section.

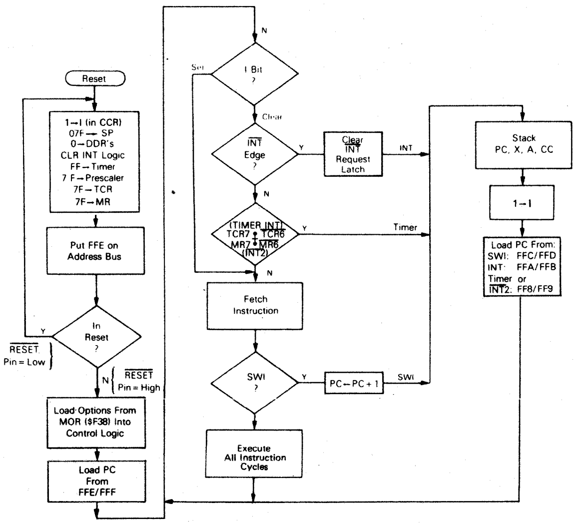
MASK OPTION REGISTER (MOR)

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 MC68705U3 MCU can be interrupted four different ways: through the external interrupt ($\overline{\text{INT}}$) input pin, the internal timer interrupt request, the external Port D bit 6 ($\overline{\text{INT2}}$) input pin, or the software interrupt instruction (SWI). When any interrupt occurs: the current instruction (including SWI) is completed, processing is suspensed, 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 15. 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). Figure 6 provides a map of the interrupts and the address of the vector which contains the starting address of the appropriate interrupt service routine. When the interrupt mask bit in the Condition Code Register is set, the interrupt is latched for later interrupt execution.

FIGURE 15 — RESET AND INTERRUPT PROCESSING FLOWCHART



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 signl (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 16a, 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 provide a 2f clock. For digital application the $\overline{\text{INT}}$ pin can be driven by a digital signal at a maximum period of t_{WL} as shown in Figure 16b.

A software interrupt (SWI) is an executable instruction which is executed regardless of the state of the I-bit in the Condition Code Register. SWI's are usually used as breakpoints for debugging or as system calls.

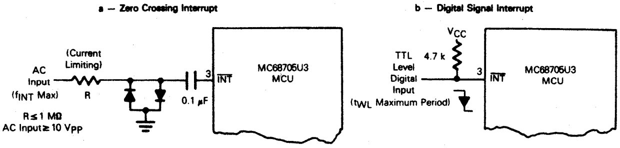
INPUT/OUTPUT

There are 32 input/output pins. The $\overline{\text{INT}}$ 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 and should be initialized by software before changing the DDRs from input to output. When programmed as outputs, all output ports read latched output data, regardless of the logic levels at the output pin due to output loading; refer to Figure 17.

All input/output lines are TTL compatible as both inputs and outputs. Port A lines are CMOS compatible as outputs while port B, C, and D lines are CMOS compatible as inputs. Port D lines are input only; thus, there is no corresponding DDR. The second external interrupt input ($\overline{\text{INT2}}$) is on bit 6 of Port D. When programmed as outputs, port B is capable of sinking 10 milliamperes and sourcing 1.0 milliampere on each pin.

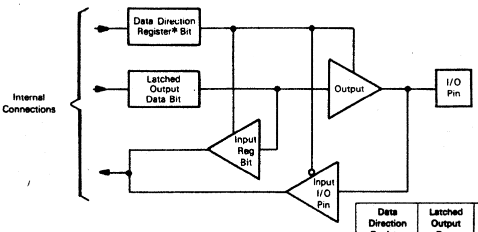
The memory map in Figure 6 gives the addresses of data registers and DDRs. The register configuration is provided in Figure 18. Figure 19 provides some example of port connections.

FIGURE 16 — TYPICAL INTERRUPT CIRCUITS


a — Zero Crossing Interrupt

b — Digital Signal Interrupt

FIGURE 17 — TYPICAL PORT I/O CIRCUITRY



Data Direction Register Bit	Latched 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 1's.

** Ports B and C are three-state ports. Port A has internal pullup devices to provide CMOS drive capability. See Electrical Characteristics Table for complete information.

FIGURE 18 — MCU REGISTER CONFIGURATION

PORT DATA DIRECTION REGISTER (DDR)

7 … 0

(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

PORT DATA REGISTER

7 … 0

Port A Addr = $000
Port B Addr = $001
Port C Addr = $002
Port D Addr = $003

TIMER DATA REGISTER (TDR)

7	0	
MSB	LSB	$008

MISCELLANEOUS REGISTER (MR)

7	6						0	
		1	1	1	1	1	1	$00A

Bit 7 — $\overline{\text{INT2}}$ Interrupt Request Bit: Set when falling edge detected on $\overline{\text{INT2}}$ pin, must be cleared by software. Cleared by 0 to Reset.
Bit 6 — $\overline{\text{INT2}}$ Interrupt Mask Bit: 1 = $\overline{\text{INT2}}$ Interrupt masked (disabled). Set to 1 by Reset.
Bits 5, 4, 3, 2, 1, 0 — Read as 1s — unused bits.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0	
								$009

See detail description in TIMER CONTROL REGISTER section.

MASK OPTION REGISTER (MOR)

7	6	5	4	3	2	1	0	
								$F38

See detail description in MASK OPTIONS section.

PROGRAMMING CONTROL REGISTER (PCR)

7	3	2	1	0	
					$00B

See detail description in ON-CHIP PROGRAMMING HARDWARE section.

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.

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 DDRis an input (0) and corresponds to the latched output data when the DDR is an output (1).

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 MC6805U2. 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 Figure 9 and TIMER section).

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	1	1	PSC*	1	1	1	Timer Control Register $009

TCR with MOR TOPT = 1 (MC6805R2 Emulation)

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	TIN	TIE	PSC*	PS2	PS1	PS0	Timer Control Register $009

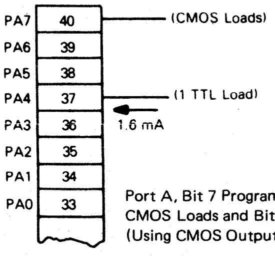
TCR with MOR TOPT = 0 (Software Programmable Timer)

* = write only, reads as a zero

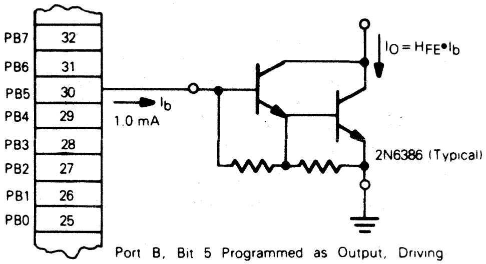
b7, TIR	Timer Interrupt Request — Used to initiate the timer interrupt or signal a Timer Data Register underflow when it is a logical "1". 1 = Set when the Timer Data Register changes to all zeros. 0 = Cleared by external reset or under program control.
b6, TIM	Timer Interrupt Mask — Used to inhibit 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.

FIGURE 19 — TYPICAL PORT CONNECTIONS

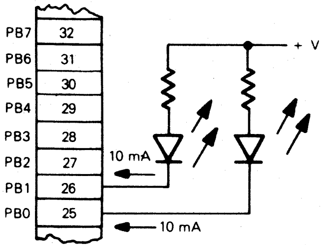
a. Output Modes



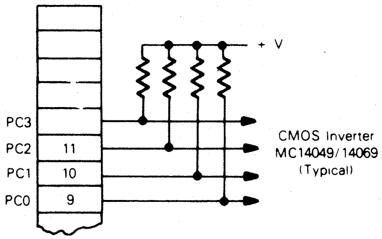
Port A, Bit 7 Programmed as Output, Driving CMOS Loads and Bit 4 one TTL Load Directly. (Using CMOS Output Option).



Port B, Bit 5 Programmed as Output, Driving Darlington-Base Directly.

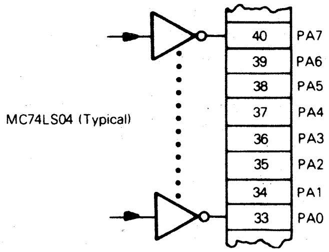


Port B, Bit 0 and Bit 1 Programmed as Output, Driving LEDs Directly.

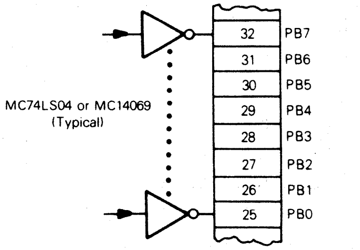


Port C, Bits 0-3 Programmed as Output, Driving CMOS Loads, Using External Pullup Resistors.

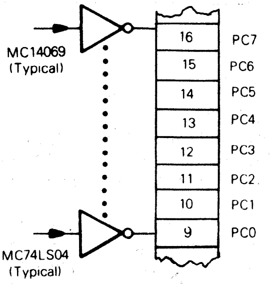
b. Input Modes



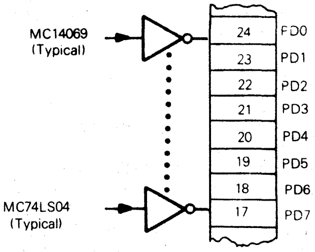
TTL Driving Port A Directly.



CMOS or TTL Driving Port B Directly.



CMOS and TTL Driving Port C Directly.



CMOS or LSTTL driving Port D directly

b5, TIN — External or Internal—Selects the input clock source to be either the external TIMER pin (8) or the internal $\phi 2$.

1 = Selects the external clock source

0 = Selects the internal $\phi 2$ ($f_{OSC} \div 4$)

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 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 zero so the BSET and BCLR on the TCR function correctly. Writing a "1" into PSC generates a pulse which clears the prescaler.

b2, PS2 — Prescaler Select—These bits are decoded

b1, PS1 — to select one of eight taps on the timer prescaler.

b0, PS0 — The table 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 PS2-0 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 MC68705U3 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 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 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 MC6805U2 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	Mask Option
CLK	TOPT	CLS			P2	P1	P0	Register $F38

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 = MC6805U2 type timer/prescaler. All bits, except 3, 6, and 7, of the Timer Control Register (TCT) are invisible to the user. Bits 5, 2, 1, and 0 of the Mask Option Register determine the equivalent MC6805U2 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 controlled after initialization).

b5, CLS — Timer/Clock Source

1 = External TIMER pin.

0 = Internal $\phi 2$

b4 — Not used if MOR TOPT = 1 (MC6805U2 emulation). Sets initial value of TCR TIE if MOR TOPT = 0.

b3 — Not used.

b2, P2 — Prescaler Option—the logical levels of these

b1, P1 — bits, when decoded, select one of eight out-

b0, P0 — puts 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 MC6805U2 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 MC68705U3 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 MC68705U3 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	VPON	PGE	PLE	Programming Control Register $00B

b0, $\overline{\text{PLE}}$ 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)

$\overline{\text{PLE}}$ is set during a Reset, but may be cleared any time. However, its effect on the EPROM is inhibited if $\overline{\text{VPON}}$ is a logic "1".

b1, $\overline{\text{PGE}}$ Program Enable—When cleared, $\overline{\text{PGE}}$ enables programming of the EPROM. $\overline{\text{PGE}}$ can only be cleared if $\overline{\text{PLE}}$ is cleared. $\overline{\text{PGE}}$ 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 $\overline{\text{PLE}}$ is low)

$\overline{\text{PGE}}$ is set during a Reset; however, it has not effect on EPROM circuits if $\overline{\text{VPON}}$ is a logic "1".

b2, $\overline{\text{VPON}}$ (V_{PP} ON) – $\overline{\text{VPON}}$ 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

$\overline{\text{VPON}}$ being "1" "disconnects" $\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ from the rest of the chip, preventing accidental clearing of these bits from effecting the normal operating mode.

Note

$\overline{\text{VPON}}$ 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.

$\overline{\text{VPON}}$	$\overline{\text{PGE}}$	$\overline{\text{PLE}}$	Programming Conditions
0	0	0	Programming mode (program EPROM byte)
1	0	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	1	0	Programming disabled (latch address and data in EPROM)
1	1	0	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system
0	0	1	Invalid state; $\overline{\text{PGE}}$ = 0 iff $\overline{\text{PLE}}$ = 0
1	0	1	Invalid state; $\overline{\text{PGE}}$ = 0 iff $\overline{\text{PLE}}$ = 0
0	1	1	"High voltage" on V_{PP}
1	1	1	$\overline{\text{PGE}}$ and $\overline{\text{PLE}}$ disabled from system (Operating Mode)

ERASING THE EPROM

The MC68705U3 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 15 Ws/cm^2. The lamps should be used without shortwave filters and the MC68705U3 should be positioned about one inch from the UV tubes. Ultraviolet erasure clears all bits of the MC68705U3 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 MC68705U3 has 191 bytes of mask ROM containing a bootstrap program which can be used to program the MC68705U3 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) of the MC68705U3 and the $\overline{\text{RESET}}$ pin is allowed to rise above V_{IRES+}. Figure 20 provides a schematic diagram of a circuit and a summary of programming steps which can be used to program the EPROM in the MC68705U3.

PROGRAMMING STEPS

The MCM2532 UV EPROM must first be programmed with an exact duplicate of the information that is to be transferred to the MC68705U3. Non-EPROM addresses are ignored by the bootstrap. Since the MC68705U3 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 20):

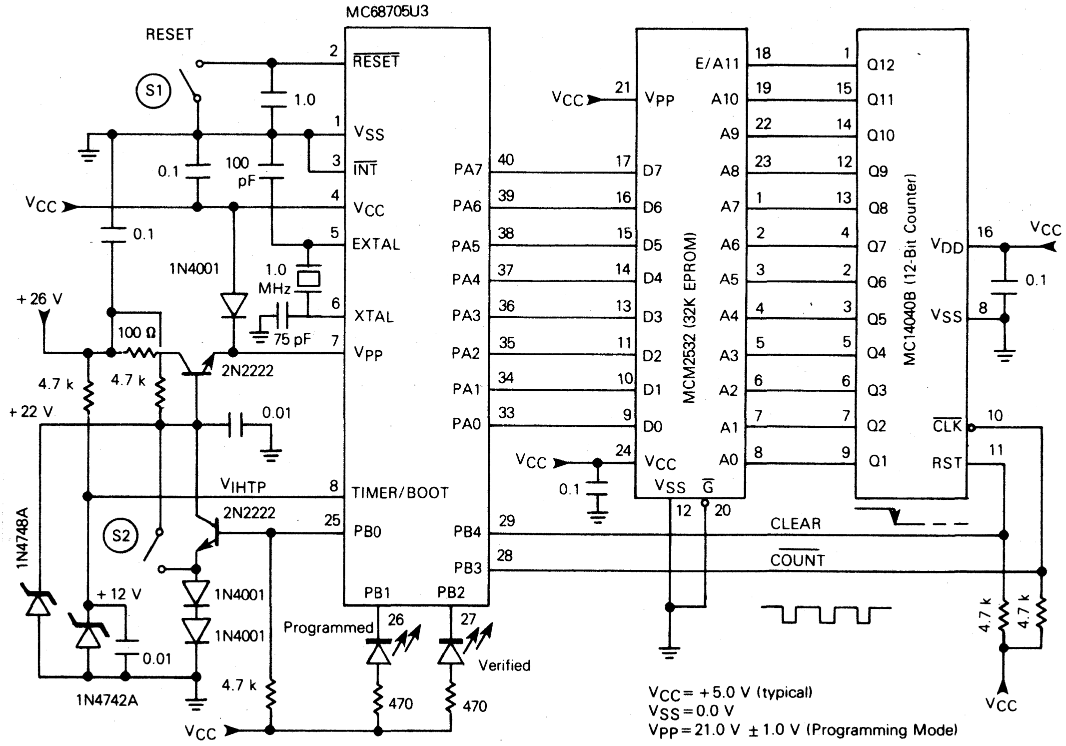
Caution

Be sure S1 and S2 are closed and V_{CC} and +26 V are not applied when inserting the MC68705U3 and MCM2532 into their respective sockets. This ensures that $\overline{\text{RESET}}$ is held low while inserting the devices.

When ready to program the MC68705U3 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 12-bit counter (MC14040B) is clocked by the PB3 output ($\overline{\text{COUNT}}$). The counter selects the MCM2532 EPROM byte which is to load the equivalent MC68705U3 EPROM byte selected by the bootstrap program. Once the EPROM location is loaded, $\overline{\text{COUNT}}$ clocks the counter to the next EPROM location. This continues until the MC68705U3 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 MC68705U3 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 MC68705U3 from its socket.

FIGURE 20 — PROGRAMMING CONNECTIONS SCHEMATIC DIAGRAM



Summary of Programming Steps:

1. When plugging in the MC68705U3 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 +26 V are applied. Then open S2, followed by S1.
3. Before removing the MC68705U3, first close S2 and then close S1. Disconnect V_{CC} and +26 V then remove the MC68705U3.

MC6805U2 EMULATION

The MC68705U3 emulates the MC6805U2 "exactly." MC6805U2 mask features are implemented in the Mask Option Register (MOR) EPROM byte on the MC68705U3. There are a few minor exceptions to the exactness of emulation which are listed below.

1. The MC6805U2 "future ROM" area is implemented in the MC68705U3 and these 1728 bytes must be left unprogrammed to accurately simulate the MC6805U2. (The MC6805U2 reads all zeros from this area.)
2. The reserved ROM areas in the MC6805U2 and MC68705U3 have different data stored in them and this data is subject to change without notice. The MC6805U2 uses the reserved ROM for the Self-Check feature and the MC68705U3 uses this area for the bootstrap program.
3. The MC6805U2 reads all ones in its 48 byte "future RAM" area. This RAM is not implemented in the MC6805U2 mask ROM version, but is implemented in the MC68705U3.
4. The V_{PP} line (pin 7) in the MC68705U3 must be tied to V_{CC} for normal operation. In the MC6805U2, pin 7 is the NUM pin and is grounded in normal operation.
5. The LVI feature is not available in the MC68705U3. 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 MC6805U2 version. Therefore, the MC68705U3 does not function in the MEX6805 Support System. In normal operation, all pin functions are the same as on the MC6805U2 version, 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 both devics 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 MC68705U3 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 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 flage (C-bit), clears the clock line, and finally accumulates the data bit in a RAM location.

ADDRESSING MODES

The MCU has 10 addressing modes which are explained briefly in the following paragraphs. For additional details and graphical illustration, 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.

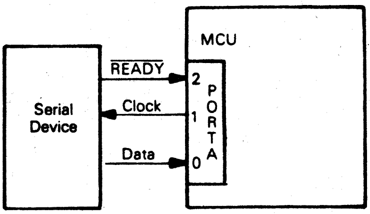
RELATIVE — The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode (the offset) is added to the PC, if and only if, the branch condition is true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −126 to +129 from the opcode address. The programmer need not worry about calculting 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

FIGURE 21 — BIT MANIPULATION EXAMPLE



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

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 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.

REGISTERS/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 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 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 3.

BRANCH INSTRUCTIONS — The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

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 test and branch operations. Refer to Table 5.

CONTROL INSTRUCTIONS — The control instructions control the MCU operations during program execution. Refer to Table 6.

ALPHABETICAL LISTING — The complete instruction set is given in alphabetical order in Table 7.

OPCODE MAP SUMMARY — Table 8 is an opcode map for the instructions used on the MCU.

TABLE 2 — 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 3 — 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 4 — 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 5 — 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 6 — 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 7 — 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 7 — INSTRUCTION SET (CONTINUED)

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
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 8 — M6805 FAMILY INSTRUCTION SET OPCODE MAP

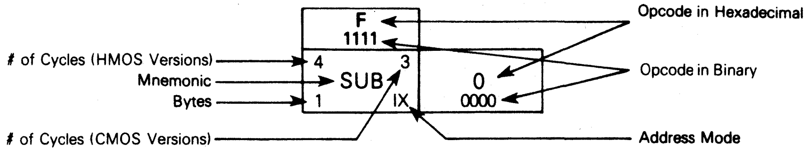
	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH(A)	INH(X)	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Hi / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	Hi / Low
0 0000	10 5 BRSET0 3 BTB	7 5 BSET0 2 BSC	4 3 BRA 2 REL	6 5 NEG 2 DIR	4 3 NEGA 1 INH	4 3 NEGX 1 INH	7 6 NEG 2 IX1	6 5 NEG 1 IX	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
1 0001	10 5 BRCLR0 3 BTB	7 5 BCLR0 2 BSC	4 3 BRN 2 REL						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 0010	10 5 BRSET1 3 BTB	7 5 BSET1 2 BSC	4 3 BHI 2 REL								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
3 0011	10 5 BRCLR1 3 BTB	7 5 BCLR1 2 BSC	4 3 BLS 2 REL	6 5 COM 2 DIR	4 3 COMA 1 INH	4 3 COMX 1 INH	7 6 COM 2 IX1	6 5 COM 1 IX	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
4 0100	10 5 BRSET2 3 BTB	7 5 BSET2 2 BSC	4 3 BCC 2 REL	6 5 LSR 2 DIR	4 3 LSRA 1 INH	4 3 LSRX 1 INH	7 6 LSR 2 IX1	6 5 LSR 1 IX			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
5 0101	10 5 BRCLR2 3 BTB	7 5 BCLR2 2 BSC	4 3 BCS 2 REL								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
6 0110	10 5 BRSET3 3 BTB	7 5 BSET3 2 BSC	4 3 BNE 2 REL	6 5 ROR 2 DIR	4 3 RORA 1 INH	4 3 RORX 1 INH	7 6 ROR 2 IX1	6 5 ROR 1 IX			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
7 0111	10 5 BRCLR3 3 BTB	7 5 BCLR3 2 BSC	4 3 BEQ 2 REL	6 5 ASR 2 DIR	4 3 ASRA 1 INH	4 3 ASRX 1 INH	7 6 ASR 2 IX1	6 5 ASR 1 IX		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
8 1000	10 5 BRSET4 3 BTB	7 5 BSET4 2 BSC	4 3 BHCC 2 REL	6 5 LSL 2 DIR	4 3 LSLA 1 INH	4 3 LSLX 1 INH	7 6 LSL 2 IX1	6 5 LSL 1 IX		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
9 1001	10 5 BRCLR4 3 BTB	7 5 BCLR4 2 BSC	4 3 BHCS 2 REL	6 5 ROL 2 DIR	4 3 ROLA 1 INH	4 3 ROLX 1 INH	7 6 ROL 2 IX1	6 5 ROL 1 IX		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
A 1010	10 5 BRSET5 3 BTB	7 5 BSET5 2 BSC	4 3 BPL 2 REL	6 5 DEC 2 DIR	4 3 DECA 1 INH	4 3 DECX 1 INH	7 6 DEC 2 IX1	6 5 DEC 1 IX		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
B 1011	10 5 BRCLR5 3 BTB	7 5 BCLR5 2 BSC	4 3 BMI 2 REL							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
C 1100	10 5 BRSET6 3 BTB	7 5 BSET6 2 BSC	4 3 BMC 2 REL	6 5 INC 2 DIR	4 3 INCA 1 INH	4 3 INCX 1 INH	7 6 INC 2 IX1	6 5 INC 1 IX		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
D 1101	10 5 BRCLR6 3 BTB	7 5 BCLR6 2 BSC	4 3 BMS 2 REL	6 4 TST 2 DIR	4 3 TSTA 1 INH	4 3 TSTX 1 INH	7 5 TST 2 IX1	6 4 TST 1 IX		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
E 1110	10 5 BRSET7 3 BTB	7 5 BSET7 2 BSC	4 3 BIL 2 REL						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
F 1111	10 5 BRCLR7 3 BTB	7 5 BCLR7 2 BSC	4 3 BIH 2 REL	6 5 CLR 2 DIR	4 3 CLRA 1 INH	4 3 CLRX 1 INH	7 6 CLR 2 IX1	6 5 CLR 1 IX	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

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

* CMOS Versions Only

LEGEND

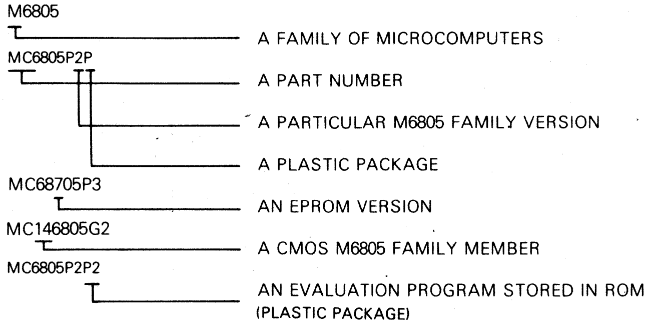


M6805 FAMILY

	Available Now					
	MC6805P2	**MC6805U2**	**MC6805R2**	**MC68705P3**	**MC146805E2**	**MC146805G2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS	CMOS
Number of Pins	28	40	40	28	40	40
On-Chip RAM (Bytes)	64	64	64	112	112	112
On-Chip User ROM (Bytes)	1.1K	2K	2K	1.8K EPROM	None	2K
External Bus	None	None	None	None	Yes	None
Bidirectional I/O Lines	20	24	24	20	16	32
Unidirectional I/O Lines	None	8 Inputs	6 Inputs	None	None	None
Other I/O Features	Timer	Timer	Timer, A/D	Timer	Timer	Timer
EPROM Version	MC68705P3	MC68705U3	MC68705R3	–	None	None
STOP and WAIT	No	No	No	No	Yes	Yes

	Available				
	3Q81	**3Q81**	**3Q81**	**4Q81**	**1Q82**
	MC68705U3	**MC68705R3**	**MC6805T2**	**MC6805P4**	**MC146805F2**
Technology	HMOS	HMOS	HMOS	HMOS	CMOS
Number of Pins	40	40	28	28	28
On-Chip RAM (Bytes)	112	112	64	112	64
On-Chip User ROM (Bytes)	3.8K EPROM	3.8K EPROM	2.5K	1.1K	1K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	24	24	19	20	20
Unidirectional I/O Lines	8 Inputs	6 Inputs	None	None	None
Other I/O Lines	Timer	Timer, A/D	Timer, PLL	Timer	Timer
EPROM Version	–	–	None	MC68705P3	None
STOP and WAIT	No	No	No	No	Yes

M6805 FAMILY NOMENCLATURE



M6805 — A FAMILY OF MICROCOMPUTERS

MC6805P2P — A PART NUMBER

A PARTICULAR M6805 FAMILY VERSION

A PLASTIC PACKAGE

MC68705P3 — AN EPROM VERSION

MC146805G2 — A CMOS M6805 FAMILY MEMBER

MC6805P2P2 — AN EVALUATION PROGRAM STORED IN ROM (PLASTIC PACKAGE)

MOTOROLA

MC146805E2

CMOS

(HIGH PERFORMANCE SILICON GATE)

8-BIT MICROPROCESSOR

Advance Information

8-BIT MICROPROCESSOR UNIT

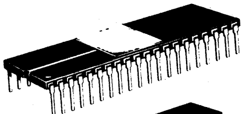
The MC146805E2 Microprocessor Unit (MPU) belongs to the M6805 Family of Microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and TIMER. It is a low-power, low-cost processor designed for low-end to mid-range applications in the consumer, automotive, industrial, and communications markets where very low power consumption constitutes an important factor. The following are the major features of the MC146805E2 MPU:

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

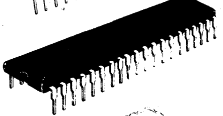
SOFTWARE FEATURES

- 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

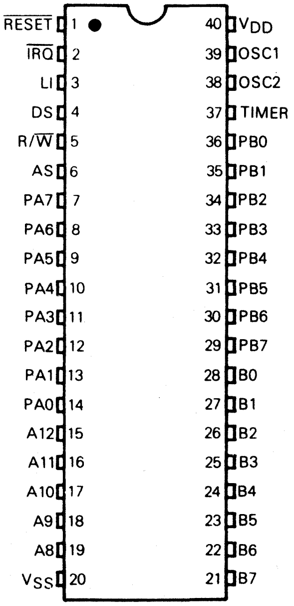
L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS



ADI-850-R2

MAXIMUM RATINGS (voltages referenced to V_{SS})

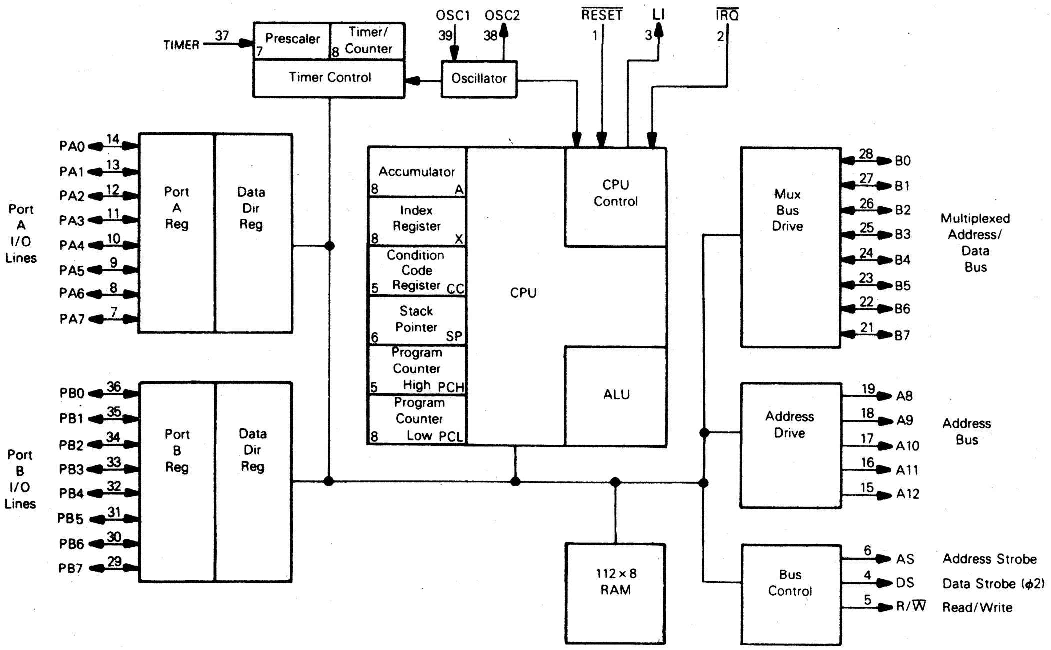
Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146805E2 MC146805E2C	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

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 ($I_{Load} \le 10.0\ \mu A$)	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/$\overline{\text{W}}$	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/$\overline{\text{W}}$, PA0-PA7	V_{OL}	–	0.3	V
Input High Voltage				
PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
TIMER, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$	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				
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1	I_{in}	–	± 1	μA
Hi-Z Output Leakage				
PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	–	± 10	μA
Capacitance				
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER	C_{in}	–	8.0	pF
Capacitance				
DS, AS, R/$\overline{\text{W}}$, 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.

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
Wait (Test Conditions – See Note Below)	I_{DD}	–	1.5	mA
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
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:

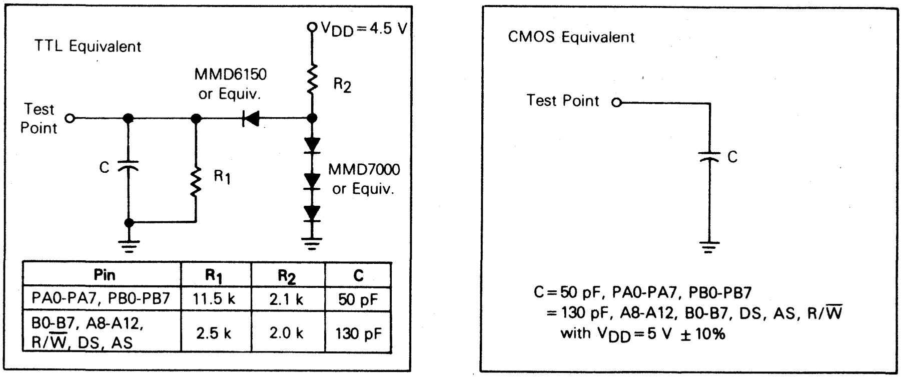
Port A and B programmed as inputs.
V_{IL}=0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
$V_{IH}=V_{DD}-0.2$ V for $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, and TIMER.
OSC1 input is a squarewave from $V_{SS}+0.2$ V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode (I_{DD}) 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.

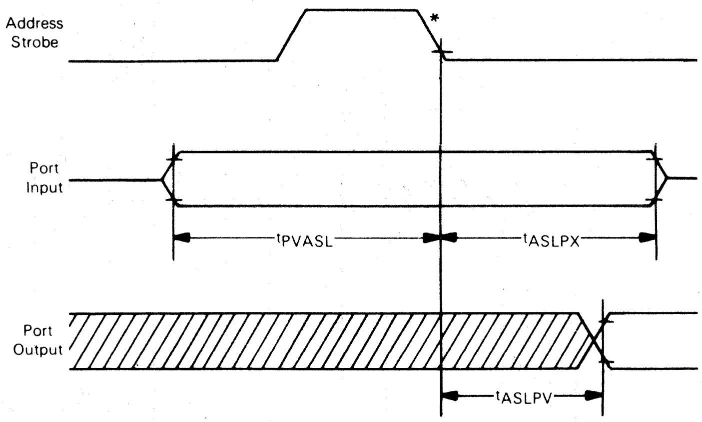
FIGURE 2 — EQUIVALENT TEST LOADS



Pin	R_1	R_2	C
PA0-PA7, PB0-PB7	11.5 k	2.1 k	50 pF
B0-B7, A8-A12, R/$\overline{W}$, DS, AS	2.5 k	2.0 k	130 pF

C = 50 pF, PA0-PA7, PB0-PB7
= 130 pF, A8-A12, B0-B7, DS, AS, R/$\overline{W}$
with $V_{DD}=5$ V ±10%

FIGURE 3 — I/O PORT TIMING
($V_{Low} = 0.8$ V, $V_{High} = V_{DD} - 2.0$ V, $V_{DD} = 5.0 \pm 10\%$
$T_A = T_L$ to T_H, C_L on Port = 50 pF, $f_{osc} = 5$ MHz)

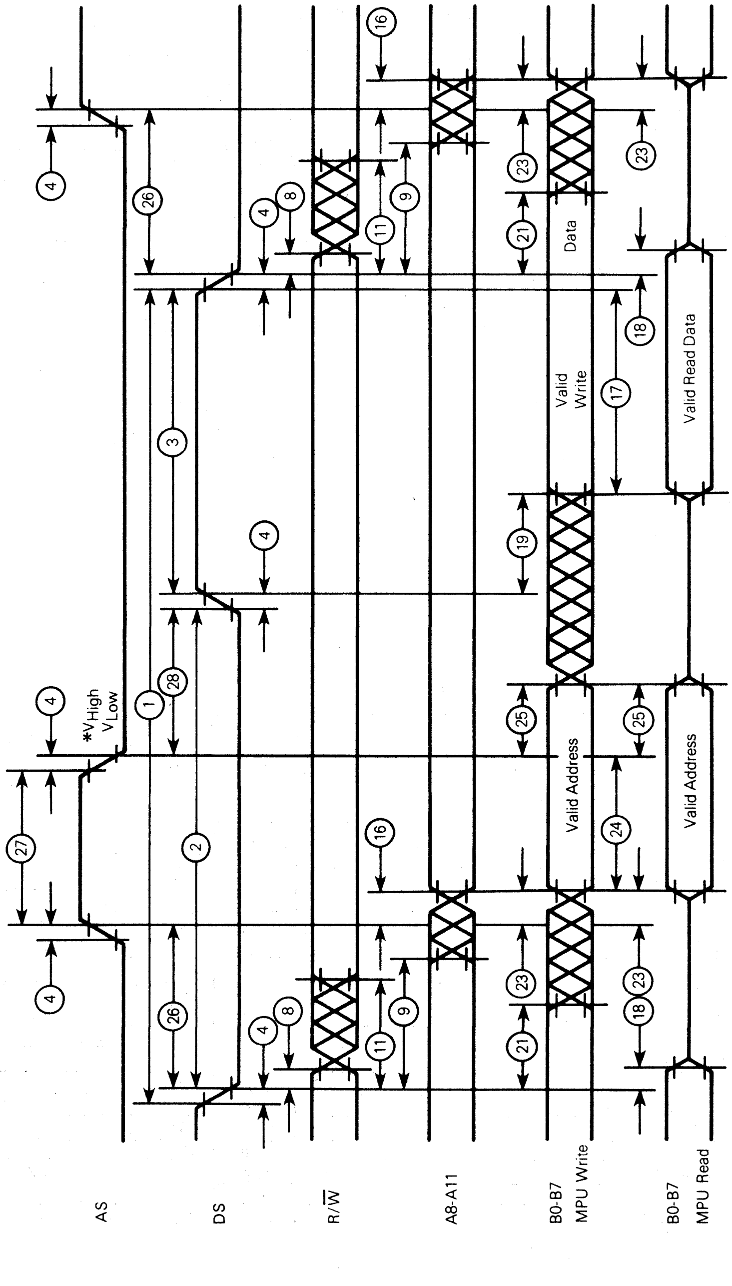


* The address strobe of the first cycle of the next instruction.

TABLE 2 — BUS TIMING ($T_A = T_L$ to T_H, $V_{SS} = 0$ V) See Figure 4

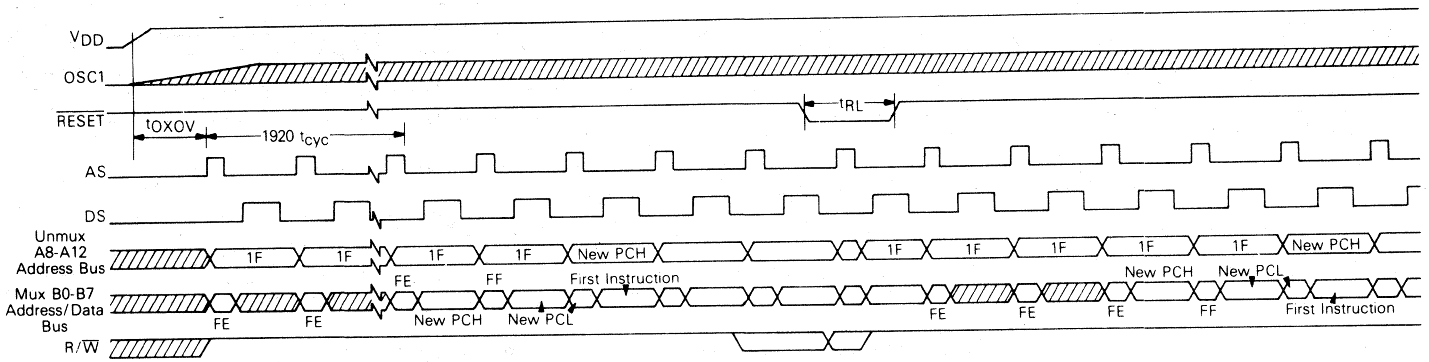
Num	Characteristics	Symbol	$f_{osc} = 1$ MHz $V_{DD} = 3.0$ V 50 pF Load		$f_{osc} = 5$ MHz $V_{DD} = 5.0$ V ± 10%, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	dc	1000	dc	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High	PW_{EH}	1800	–	375	–	ns
4	Clock 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}	5	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	1000	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

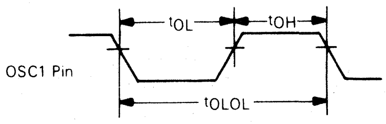


* $V_{High}=2.0$ V, $V_{Low}=0.5$ V for $V_{DD}=3$ V
$V_{High}=V_{DD}-2.0$ V, $V_{Low}=0.8$ V for $V_{DD}=5$ V ±10%

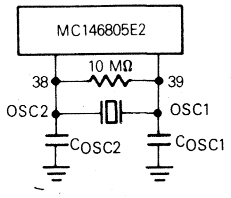
FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$ TIMING



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	15-25 pF	15-30 pF

Crystal Circuit

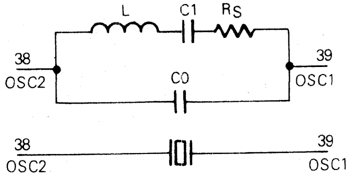
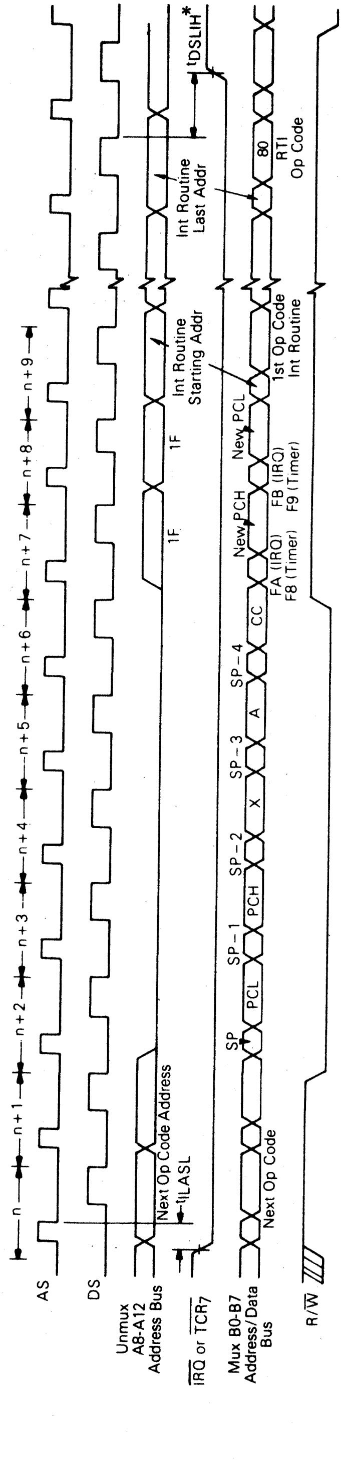


FIGURE 6 — $\overline{\text{IRQ}}$ AND $\overline{\text{TCR}}_7$ INTERRUPT TIMING



*t_{DSLIH} — The interrupting device must release the $\overline{\text{IRQ}}$ line within this time to prevent subsequent recognition of the same interrupt.

FIGURE 7 — TIMER INTERRUPT AFTER WAIT INSTRUCTION: TIMING

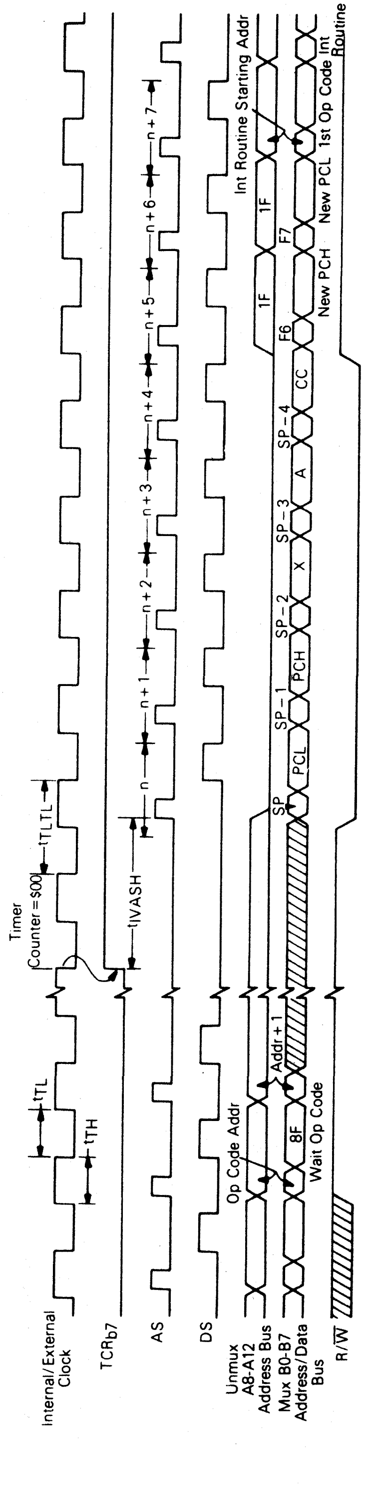
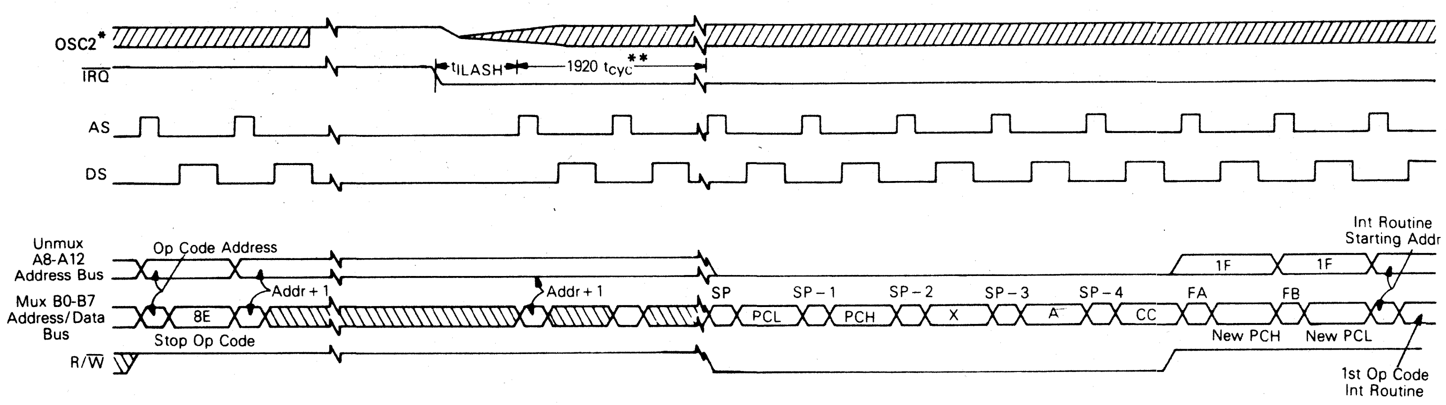


FIGURE 8 — INTERRUPT RECOVERY FROM STOP INSTRUCTION: TIMING



* 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.

$\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST)

$\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ line (see Interrupt section for more details). $\overline{\text{IRQ}}$ requires an external resistor to V_{DD} for "wire OR" operation.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ 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/$\overline{\text{W}}$ (READ/WRITE)

The R/$\overline{\text{W}}$ 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/$\overline{\text{W}}$ low = processor write; R/$\overline{\text{W}}$ high = processor read). The R/$\overline{\text{W}}$ 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 non-multiplexed 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/$\overline{\text{W}}$ 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.

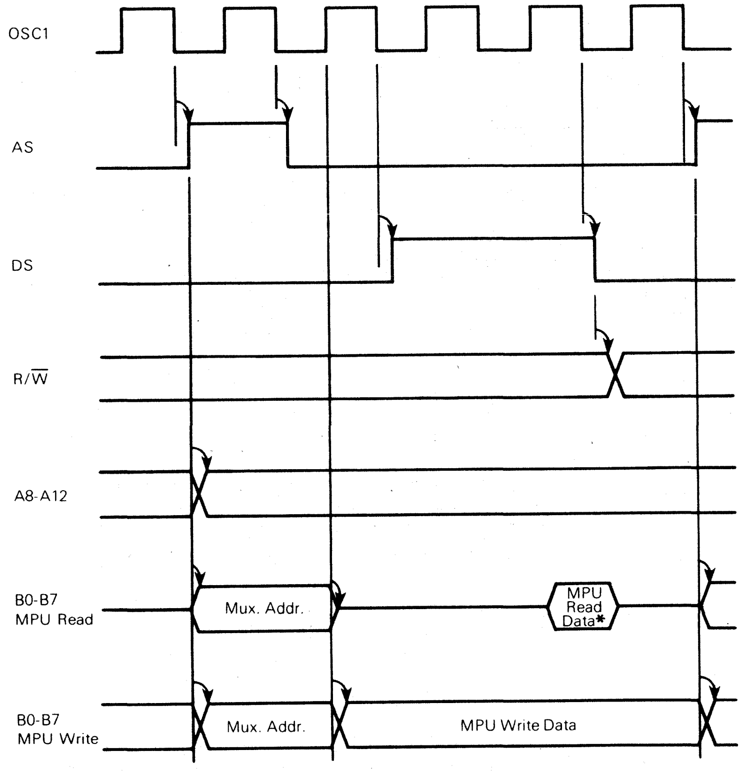
LI (LOAD INSTRUCTION)

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

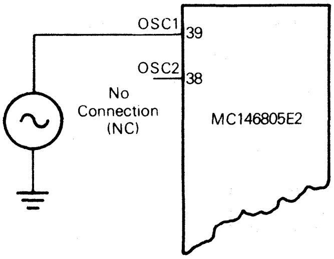
These eight pins constitute input/output port A. Each line is individually programmed to be either an input or output under software control via its data direction register as shown in Figure 11(b). An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1", and as an input 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

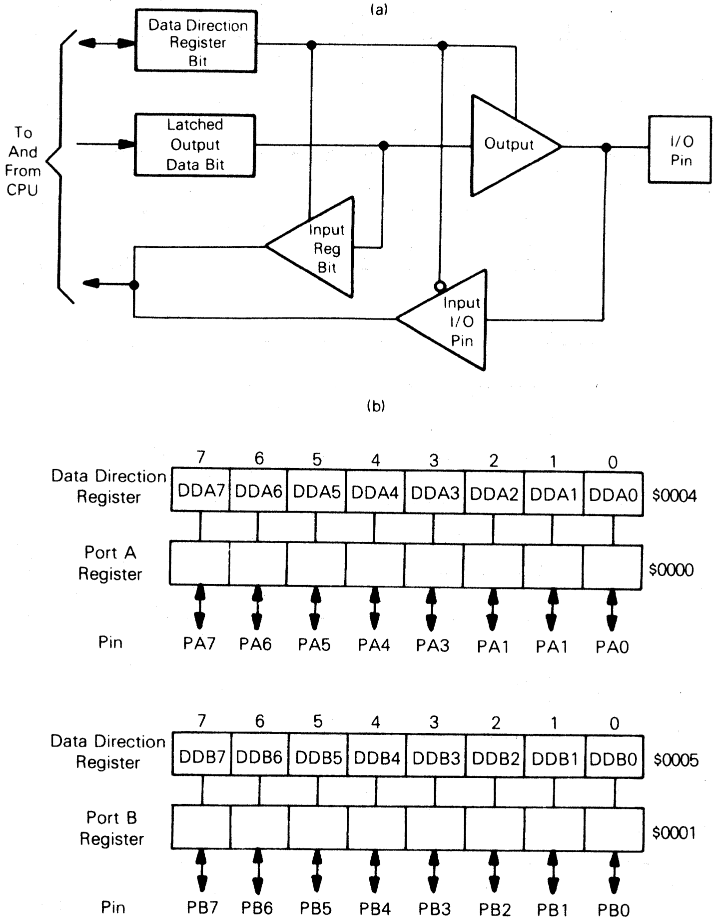


TABLE 3 — I/O PIN FUNCTIONS

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

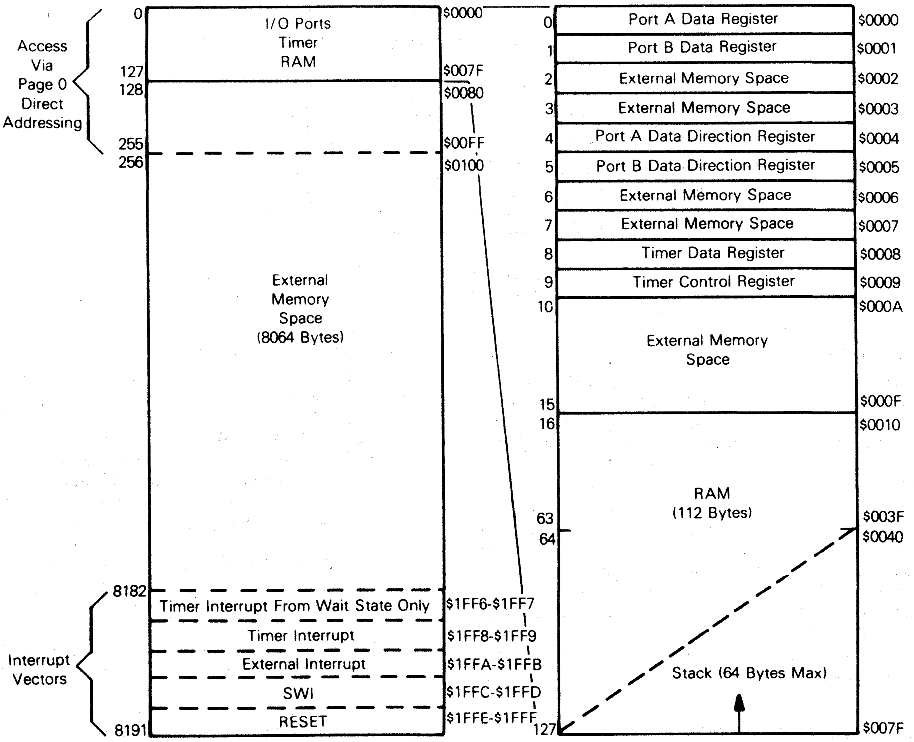


FIGURE 13 — PROGRAMMING MODEL

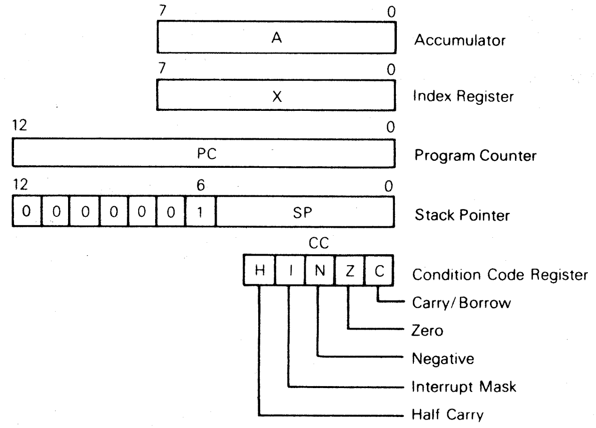
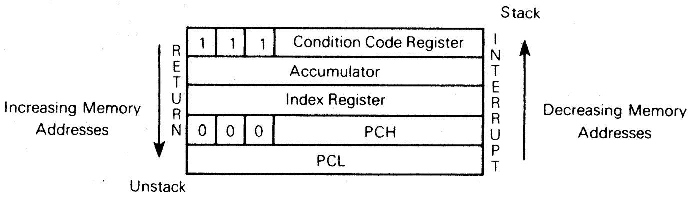


FIGURE 14 — STACKING ORDER



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) and 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 BP4XXXX 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 interrupt 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

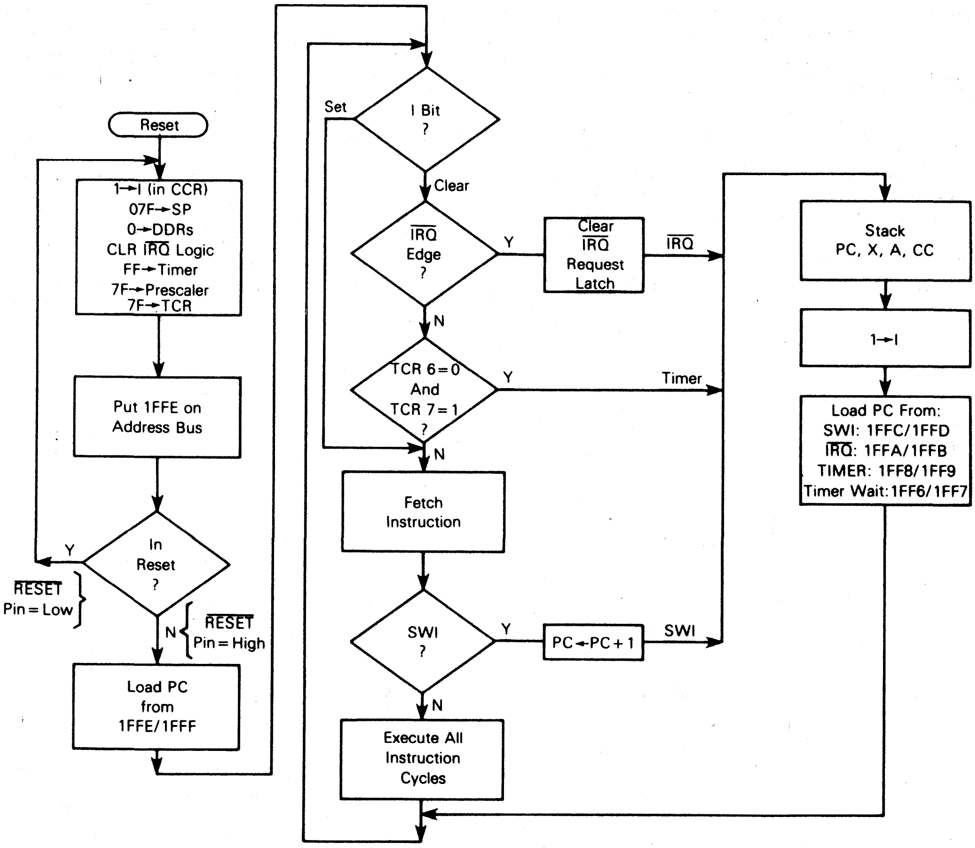
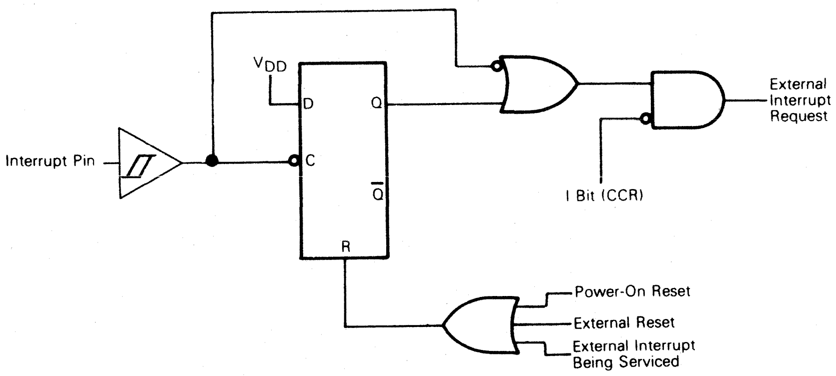
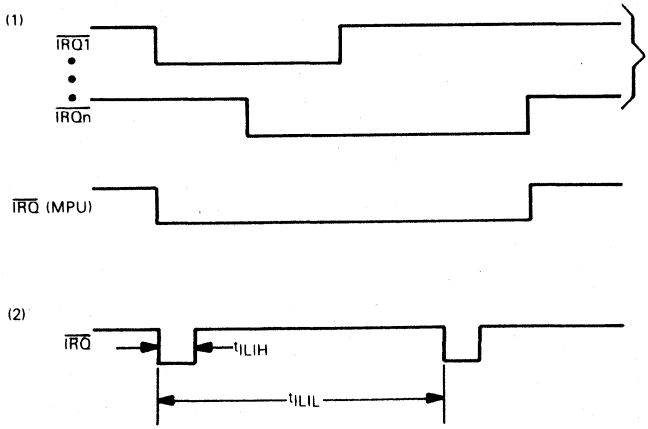


FIGURE 16 — EXTERNAL INTERRUPT

(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.

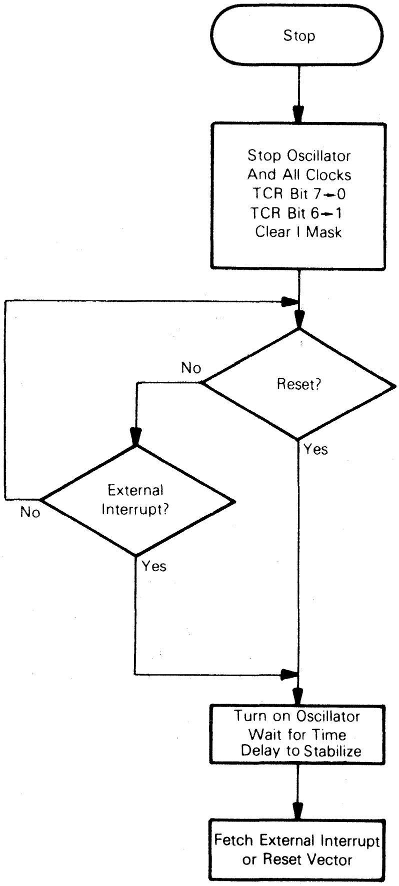
Pulse Condition

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 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 were interrupted 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).

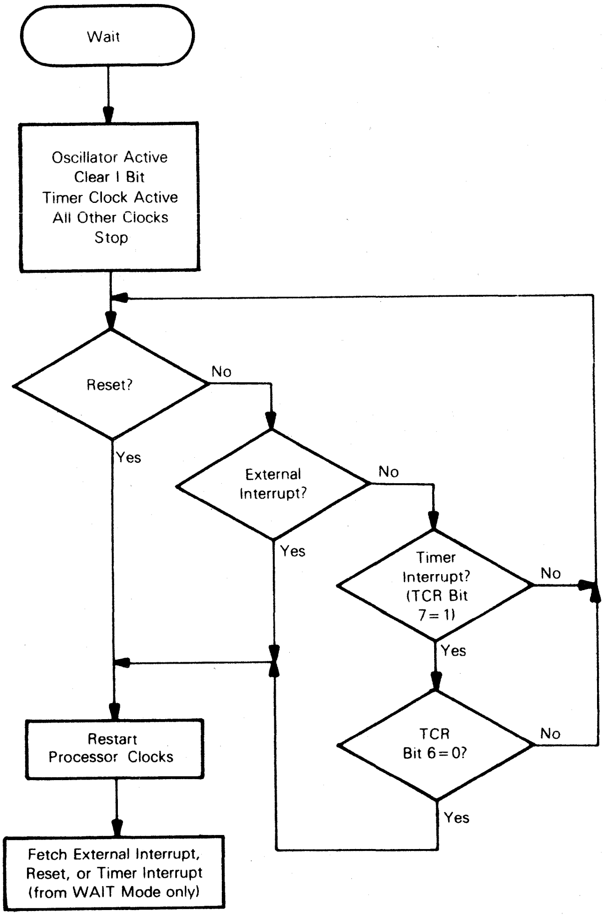
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents 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

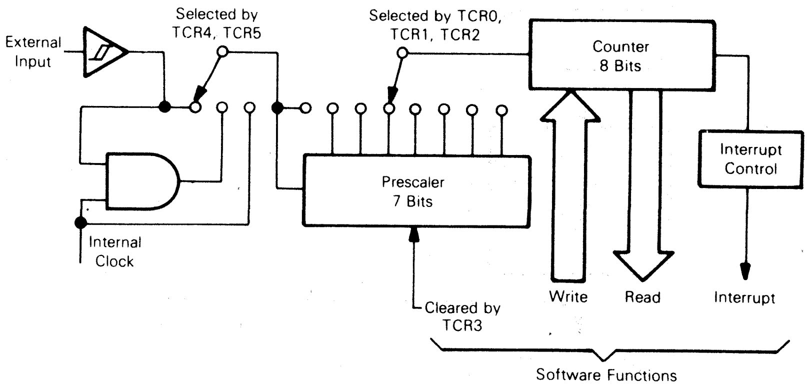
If TCR4=0 and TCR5=1, then all inputs to the timer are disabled.

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. Power-on reset and the STOP instruction cause the counter to be set to $F0.

FIGURE 19 — TIMER BLOCK DIAGRAM



NOTES:

1. Prescaler and 8-bit counter are clocked falling edge of the internal clock (AS) or external input.
2. Counter 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 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 ← 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 ← 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 ← 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 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 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.

$$EA = X + [(PC+1):(PC+2)];\ PC \leftarrow PC+3$$
$$\text{Address Bus High} \leftarrow (PC+1)+K$$
$$\text{Address Bus Low} \leftarrow K+(PC+2)$$

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.

$$EA = PC+2+(PC+1);\ PC \leftarrow EA \text{ if branch is taken;}$$
$$\text{otherwise, } PC \leftarrow PC+2$$

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.

$$EA = (PC+1);\ PC \leftarrow PC+2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC+1)$$

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.

$$EA1 = (PC+1)$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC+1)$$
$$EA2 = PC+3+(PC+2);\ PC \leftarrow EA2 \text{ if branch taken;}$$
$$\text{otherwise, } PC \leftarrow PC+3$$

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

		Relative Addressing Mode		
Function	Mnemonic	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

		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	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

		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	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)	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

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	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL								2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DTR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 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

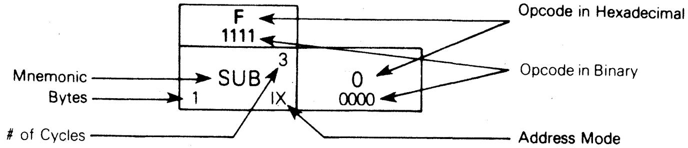


FIGURE 20 — CONNECTION TO CMOS PERIPHERALS

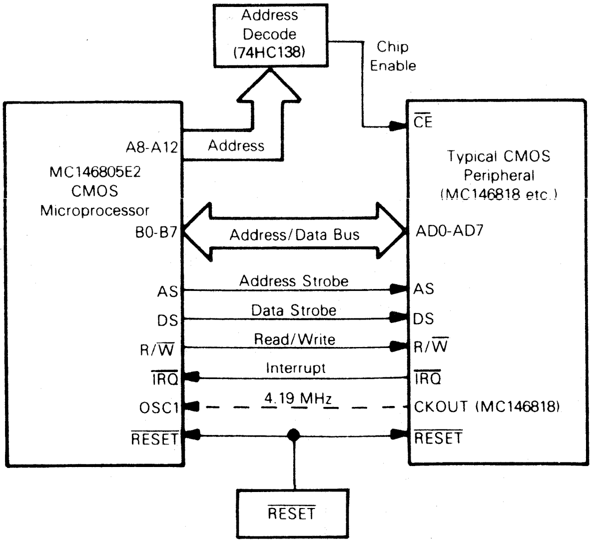


FIGURE 21 — CONNECTION TO CMOS MULTIPLEXED MEMORIES

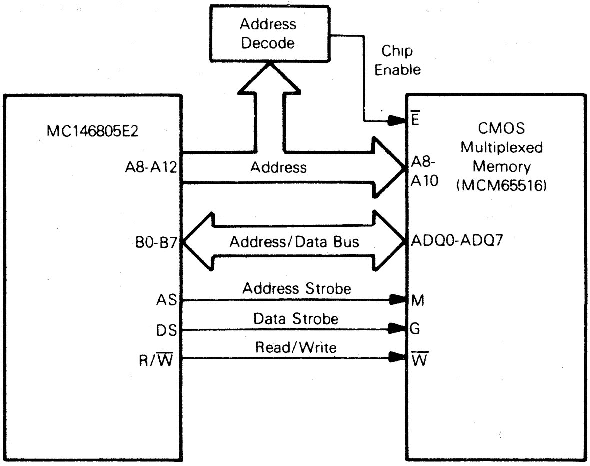
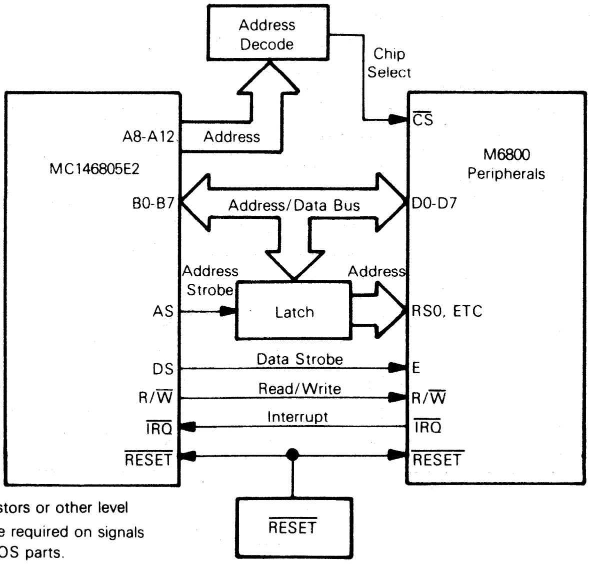


FIGURE 22 — CONNECTION TO M6800 PERIPHERALS



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

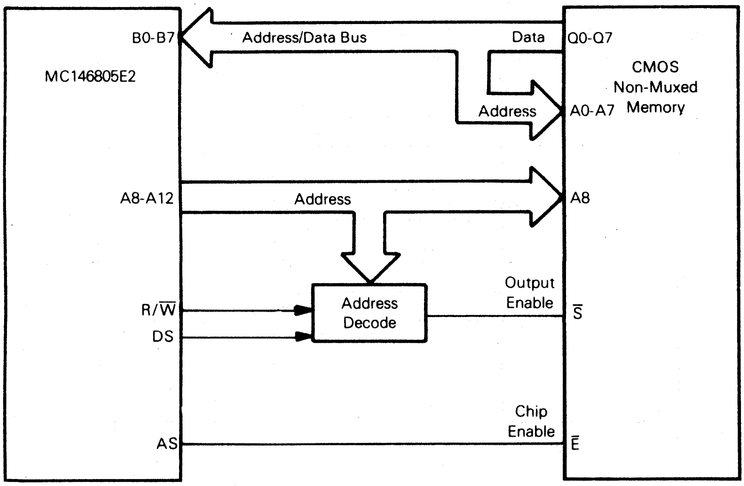


FIGURE 24 — CONNECTION TO STATIC CMOS RAMS

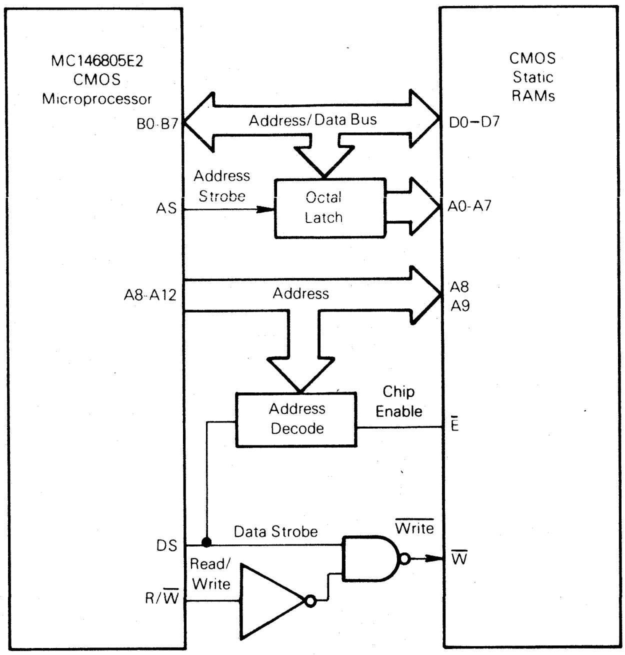


FIGURE 25 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS RAM

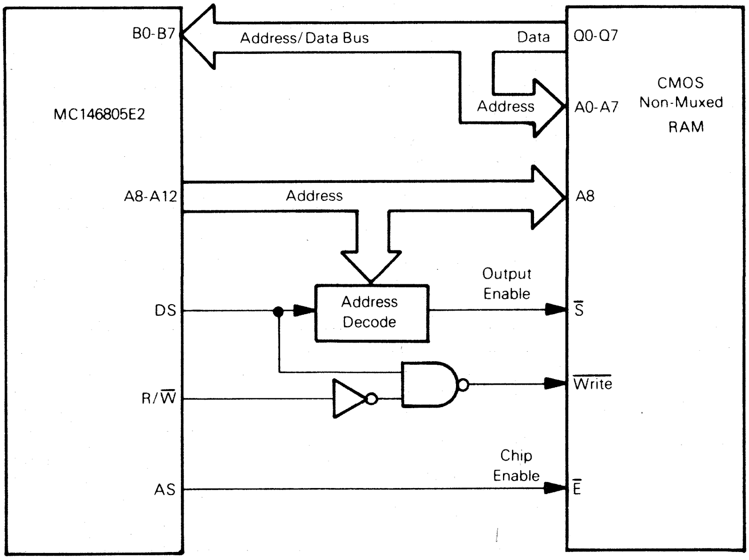


TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION

Address Mode Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Inherent						
LSR LSL ASR NEG CLR ROL COM ROR DEC INC TST	3	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
TAX CLC SEC STOP CLI SEI RSP WAIT NOP TXA	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
RTS	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	1	0	Irrelevant Data
		4	Stack Pointer + 1	1	0	Irrelevant Data
		5	Stack Pointer + 2	1	0	Irrelevant Data
		6	New Op Code Address	1	0	New Op Code
SWI	10	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	0	0	Return Address (LO Byte)
		4	Stack Pointer − 1	0	0	Return Address (HI Byte)
		5	Stack Pointer − 2	0	0	Contents of Index Register
		6	Stack Pointer − 3	0	0	Contents of Accumulator
		7	Stack Pointer − 4	0	0	Contents of CC Register
		8	Vector Address 1FFC (Hex)	1	0	Address of Int. Routine (HI Byte)
		9	Vector Address 1FFD (Hex)	1	0	Address of Int. Routine (LO Byte)
		10	Interrupt Routine Starting Address	1	0	Interrupt Routine First Opcode
RTI	9	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Op Code Next Instruction
		3	Stack Pointer	1	0	Irrelevant Data
		4	Stack Pointer + 1	1	0	Irrelevant Data
		5	Stack Pointer + 2	1	0	Irrelevant Data
		6	Stack Pointer + 3	1	0	Irrelevant Data
		7	Stack Pointer + 4	1	0	Irrelevant Data
		8	Stack Pointer + 5	1	0	Irrelevant Data
		9	New Op Code Address	1	0	New Op Code
Immediate						
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	2	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Operand Data
Bit Set/Clear						
BSET n BCLR n	5	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	Address of Operand	1	0	Operand Data
		5	Address of Operand	0	0	Manipulated Data
Bit Test and Branch						
BRSET n BRCLR n	5	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	Branch Offset
		5	Op Code Address + 2	1	0	Branch Offset
Relative						
BCC BHI BNE BEQ BCS BPL BHCC BLS BIL BMC BRN BHCS BIH BMI BMS BRA	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Branch Offset
		3	Op Code Address + 1	1	0	Branch Offset
BSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Branch Offset
		3	Op Code Address + 1	1	0	Branch Offset
		4	Subroutine Starting Address	1	0	First Subroutine Op Code
		5	Stack Pointer	0	0	Return Address (LO Byte)
		6	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
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, if **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.

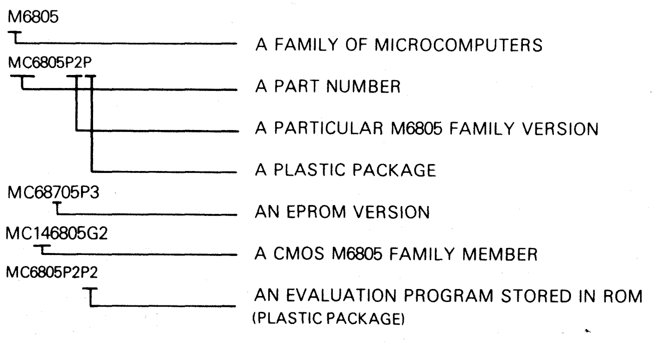
On future MC146805E2 parts, no special actions will be necessary. If you have questions, contact your Motorola distributor or Motorola sales office, or contact Motorola Microprocessor Applications Engineering in Austin, Texas.

M6805 FAMILY

	MC6805P2	MC6805P4	MC6805R2	MC6805T2	MC6805U2
Technology	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	40	28	40
On-Chip RAM (Bytes)	64	112	64	64	64
On-Chip User ROM (Bytes)	1.1K	1.1K	2K	2.5K	2K
External Bus	None	None	None	None	None
Bidirectional I/O Lines	20	20	24	19	24
Unidirectional I/O Lines	None	None	6 Inputs	None	8 Inputs
Other I/O Features	Timer	Timer	Timer, A/D	Timer, PLL	Timer
External Interrupt Inputs	1	1	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705R3	None	MC68705U3
STOP and WAIT	No	No	No	No	No

	MC68705P3	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2
Technology	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS
Number of Pins	28	40	40	40	28	40
On-Chip RAM (Bytes)	112	112	112	112	64	112
On-Chip User ROM (Bytes)	1.8K EPROM	3.8K EPROM	3.8K EPROM	None	1K	2K
External Bus	None	None	None	Yes	None	None
Bidirectional I/O Lines	20	24	24	16	16	32
Unidirectional I/O Lines	None	6 Inputs	8 Inputs	None	4 Inputs	None
Other I/O Features	Timer	Timer, A/D	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	2	2	1	1	1
EPROM Version	–	–	–	None	None	None
STOP and WAIT	No	No	No	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE



MOTOROLA

MC146805F2

Advance Information

CMOS

(HIGH-PERFORMANCE SILICON-GATE)

8-BIT MICROCOMPUTER

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

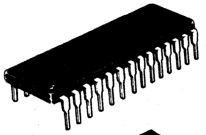
- Typical Full Speed Operating Power of 10 mW at 5 V
- Typical WAIT Mode Power of 3 mW
- Typical STOP Mode Power of 25 μW
- 8-Bit Architecture
- Fully Static Operation
- Single 3- to 6-Volt Supply
- 1089 Bytes of On-Chip User ROM
- 64 Bytes of On-Chip RAM
- Memory Mapped I/O
- 16 Bidirectional I/O Lines
- 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

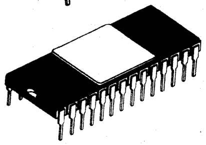
- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Ten Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- User Callable Self-Check Routines
- Two Power Saving Standby Modes

USER SELECTABLE OPTIONS

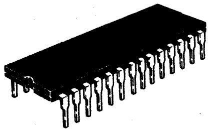
- Crystal or Low-Cost Resistor Oscillator Option
- Oscillator Internally Divided by 2 or 4
- Interrupts Edge Sensitive Only or Level and Edge Sensitive



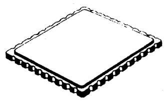
P SUFFIX
PLASTIC PACKAGE
CASE 710



L SUFFIX
CERAMIC PACKAGE
CASE 719

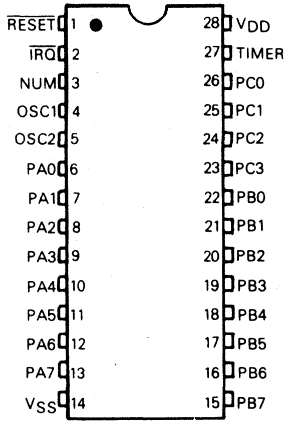


S SUFFIX
CERDIP PACKAGE
CASE 733



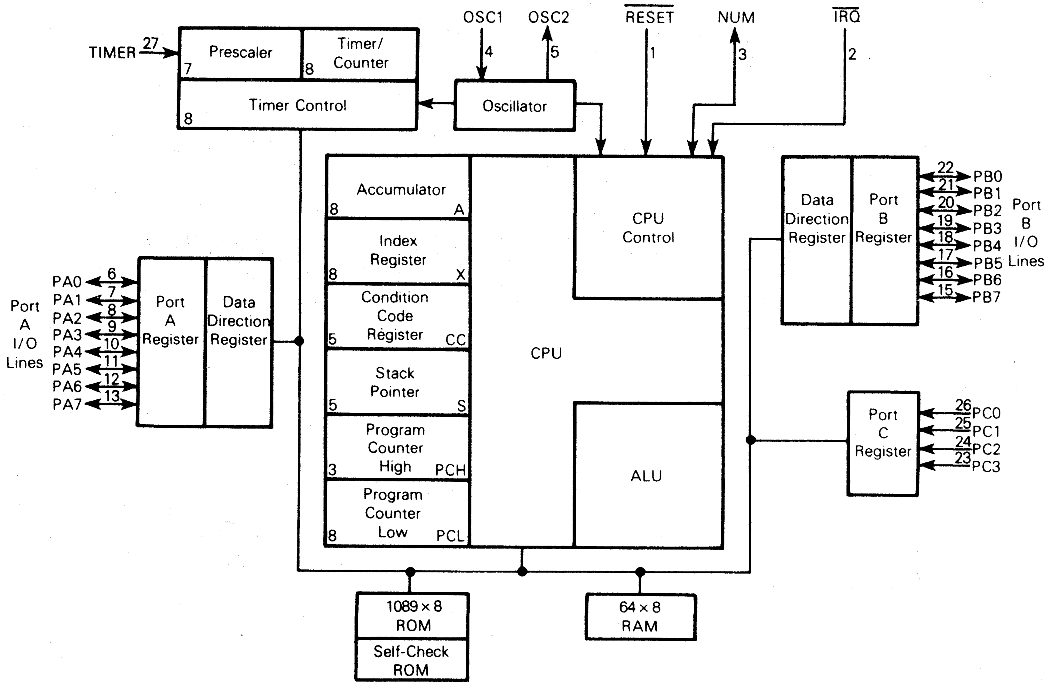
Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS



ADI-879

FIGURE 1 — MC146805F2 CMOS MICROCOMPUTER



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 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 $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. 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 TBD	°C/W

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, $I_{Load} \leq 10.0$ μA	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Output High Voltage ($I_{Load}=-200$ μA) PA0-PA7, PB0-PB7	V_{OH}	4.1	–	V
Output Low Voltage, ($I_{Load}=800$ μA) PA0-PA7, PB0-PB7	V_{OL}	–	0.4	V
Input High Voltage Ports PA0-PA7, PB0-PB7, PC0-PC3 TIMER, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$ OSC1	V_{IH}	 $V_{DD}-2.0$ $V_{DD}-0.8$ $V_{DD}-1.5$	 V_{DD} 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, $t_{cyc}=1$ μs) RUN (Measured During Self-Check, $V_{IL}=0.2$ V, $V_{IH}=V_{DD}-0.2$ V) WAIT (See Note 2) STOP (See Note 2)	I_{DD}	 – – –	 5 2 200	 mA mA μA
I/O Ports Input Leakage – PA0-PA7, PB0-PB7	I_{IL}	–	±10	μA
Input Current – $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER, OSC1, PC0-PC3	I_{in}	–	±1	μA
Output Capacitance – Ports A and B	C_{out}	–	12	pF
Input Capacitance – $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, 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 $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, TIMER
    OSC1 input is a square wave 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 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	f_{osc}			MHz
Crystal		—	4.0	
External Clock		dc	4.0	

*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

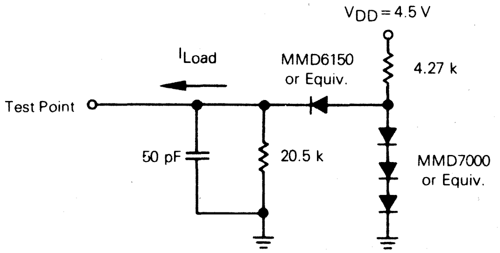


FIGURE 3 — TYPICAL OPERATING CURRENT vs INTERNAL FREQUENCY

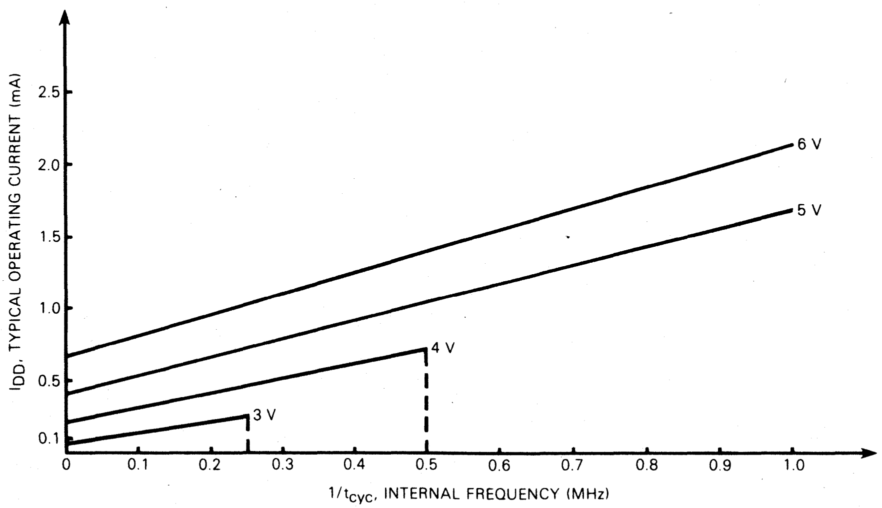


FIGURE 4 — TIMER RELATIONSHIPS

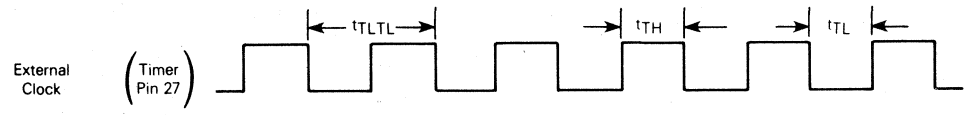
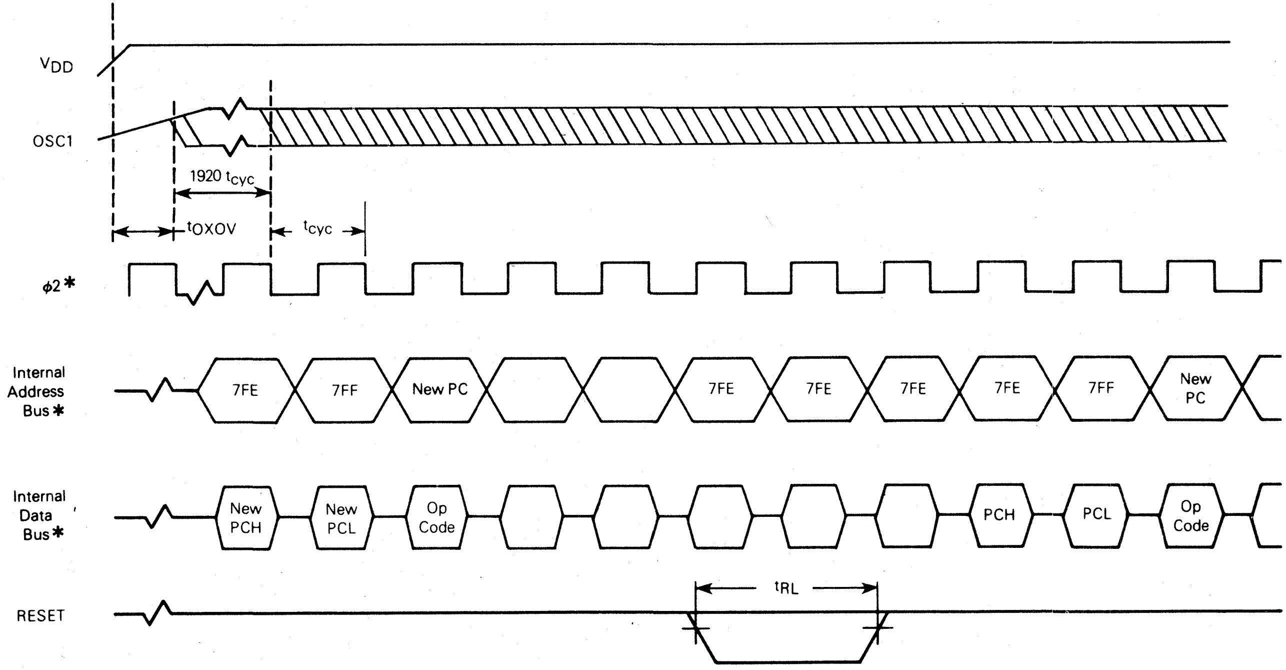
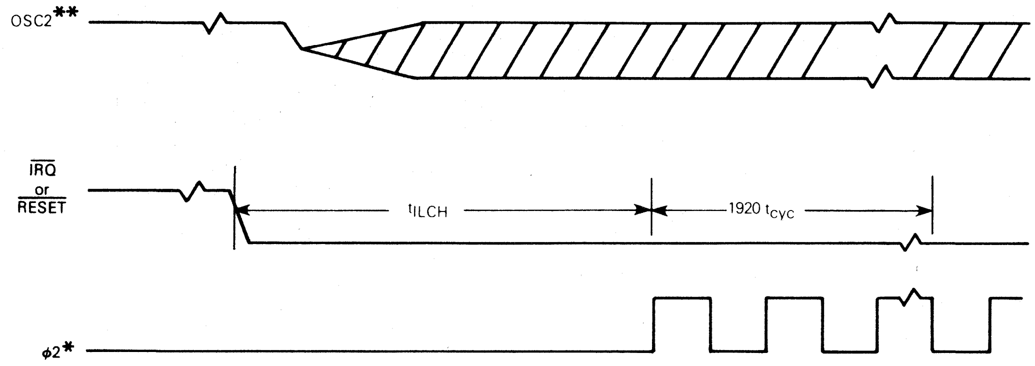


FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$



* 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.

$\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST)

$\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See the Interrupt section for more detail.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ 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 may be used with either the RC or crystal oscillator mask option. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

PA0-PA7

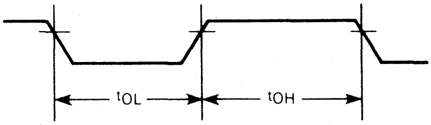
These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

FIGURE 7 — OSCILLATOR CONNECTIONS

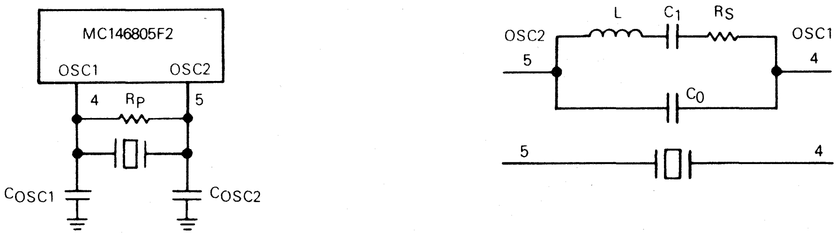
Crystal Parameters

	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	–

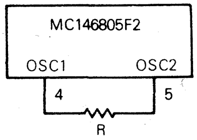
Oscillator Waveform



(a) Crystal Oscillator Connections and Equivalent Crystal Circuit



(b) RC Oscillator Connection



(c) External Clock Source Connections

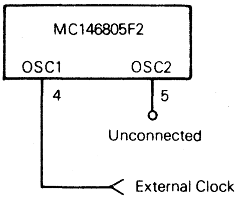
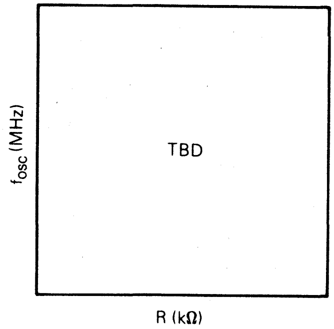


FIGURE 8 — FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



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

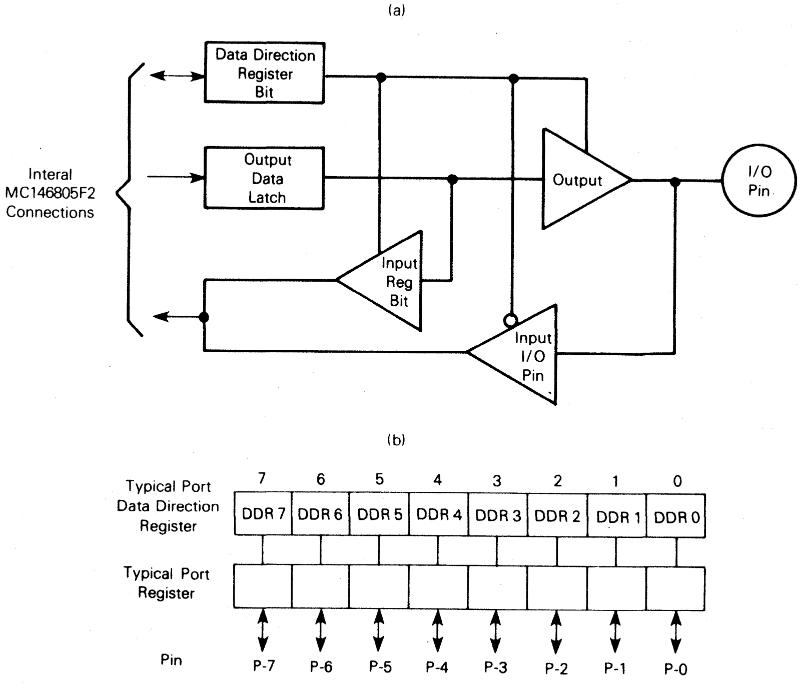


TABLE 2 — I/O PIN FUNCTIONS

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.)

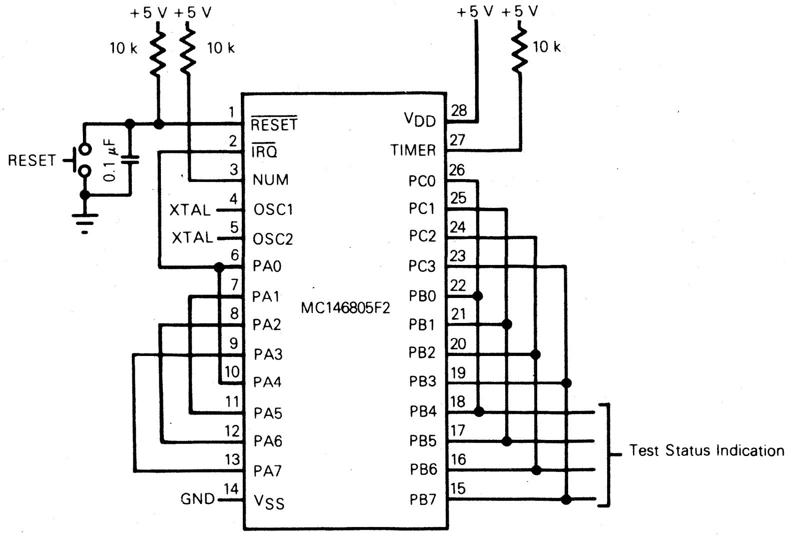
TIMER TEST SUBROUTINE

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



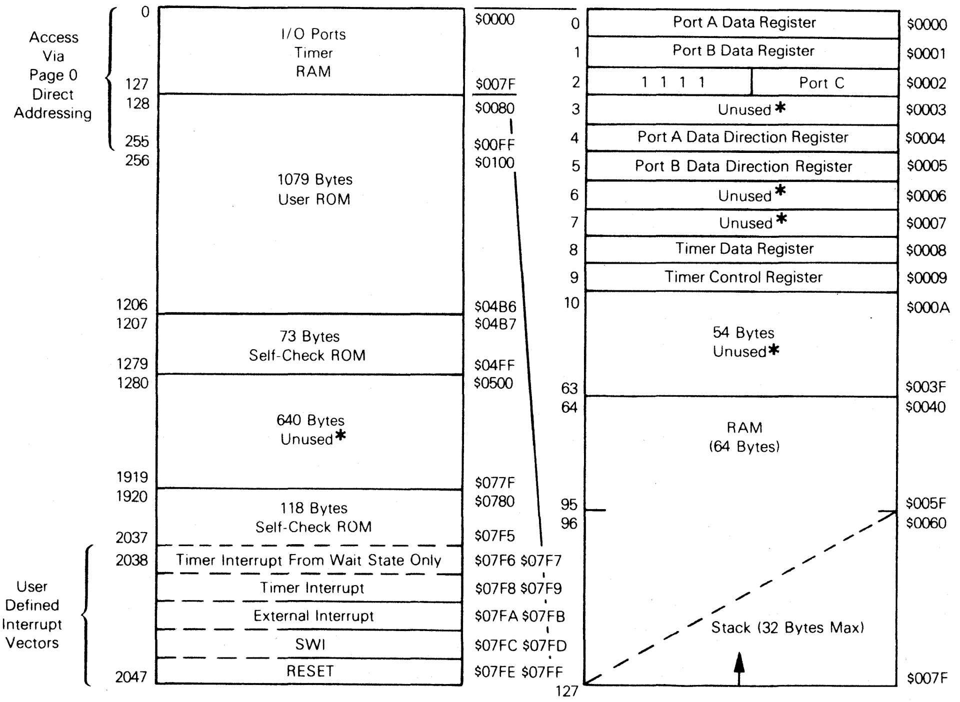
MEMORY

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

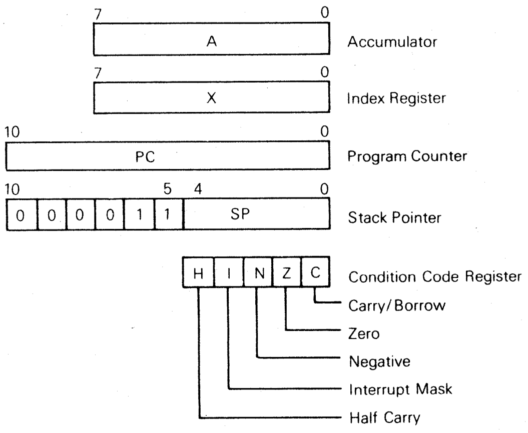
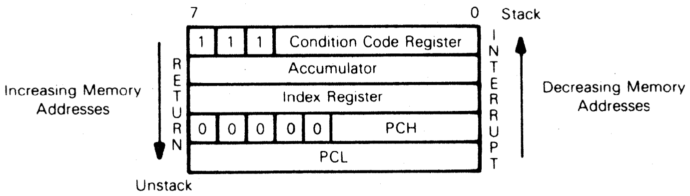


FIGURE 13 — STACKING ORDER



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.

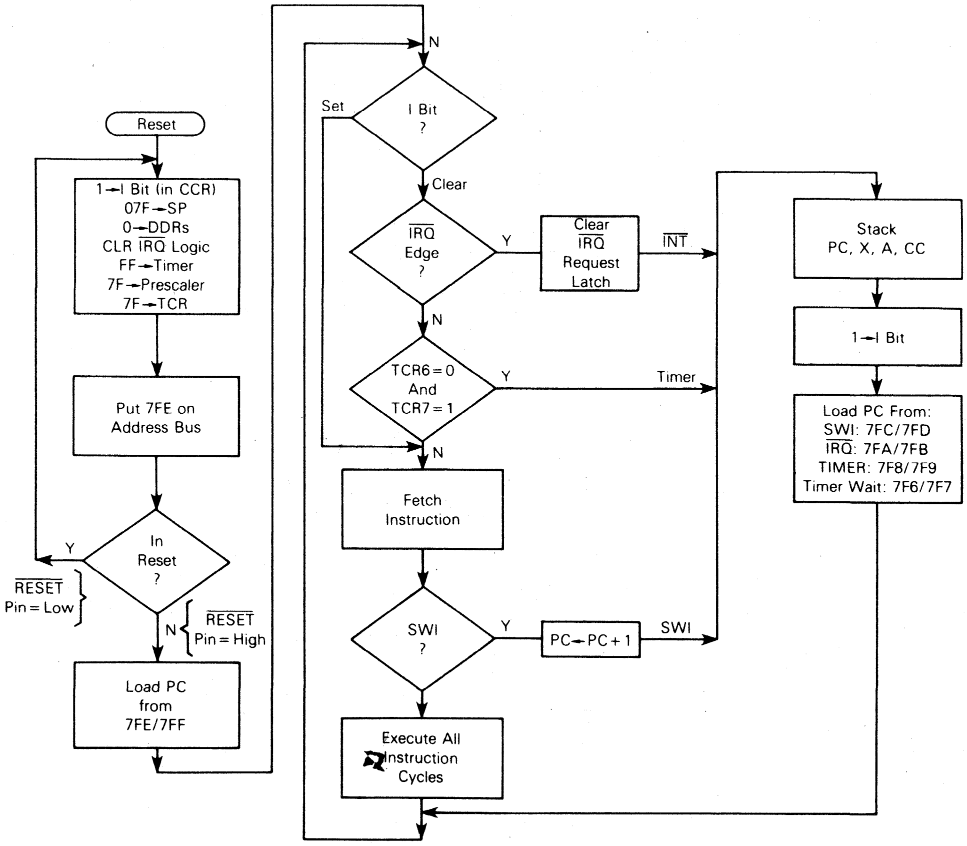
When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

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 ($\overline{\text{IRQ}}$) 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 ($\overline{\text{IRQ}}$) 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 15. The second configuration shows many interrupt lines "wire ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

SOFTWARE INTERRUPT (SWI)

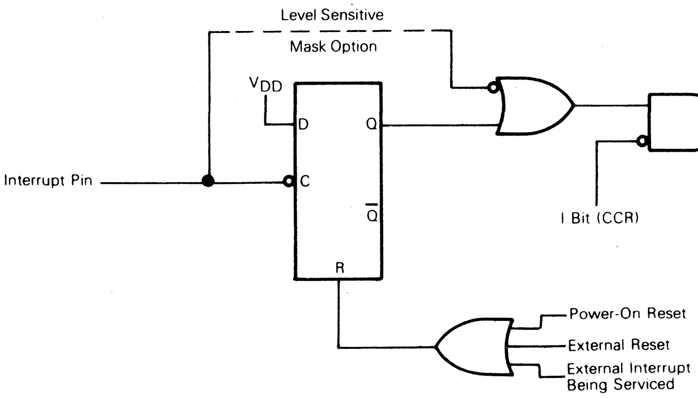
The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations \$7FC and \$7FD.

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are $\overline{\text{RESET}}$, STOP, and WAIT.

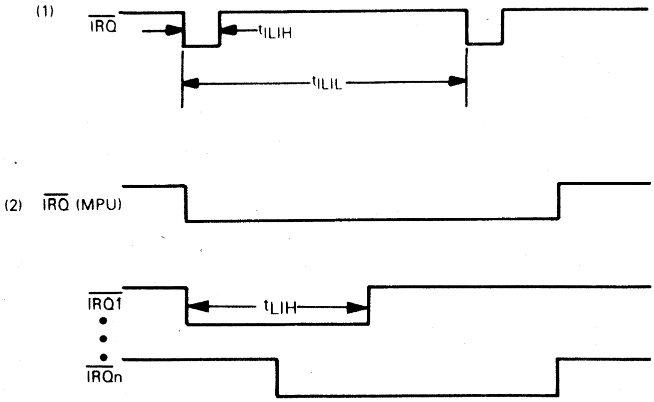
$\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input pin and the internal power-on reset function each cause the program to vector to an initialization program. This vector is specified by the contents

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

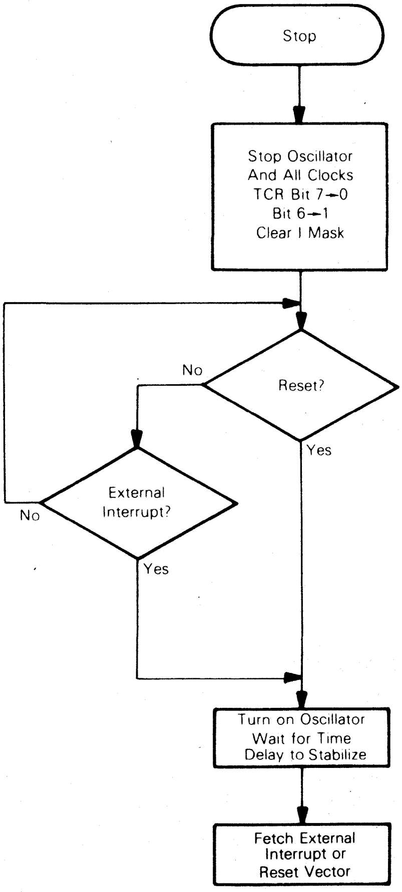
If after servicing an interrupt the $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

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 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).

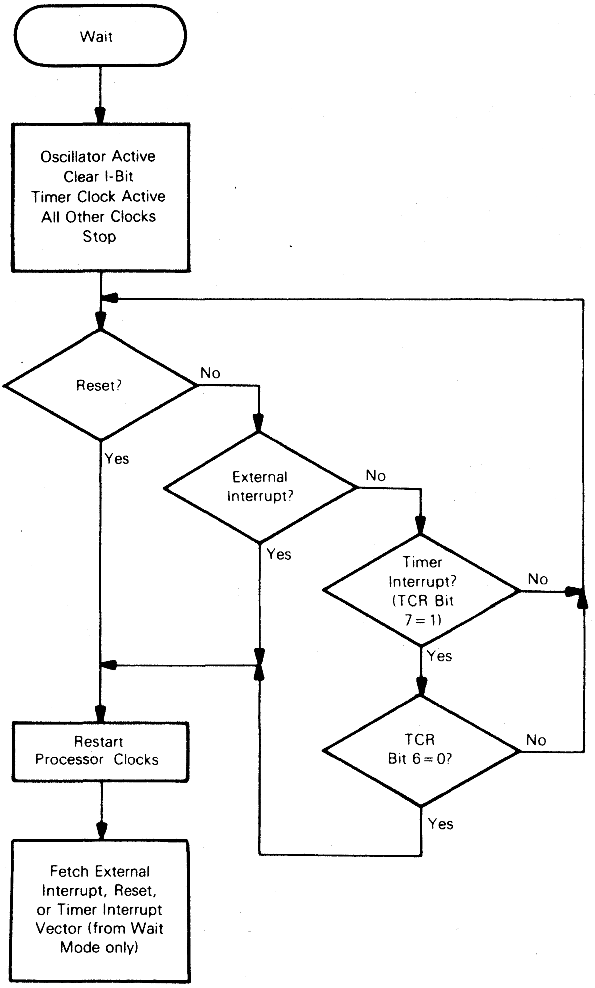
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output within the range of ÷1 to ÷128 which is used as the counter input. The processor cannot write into or read from the prescaler, 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

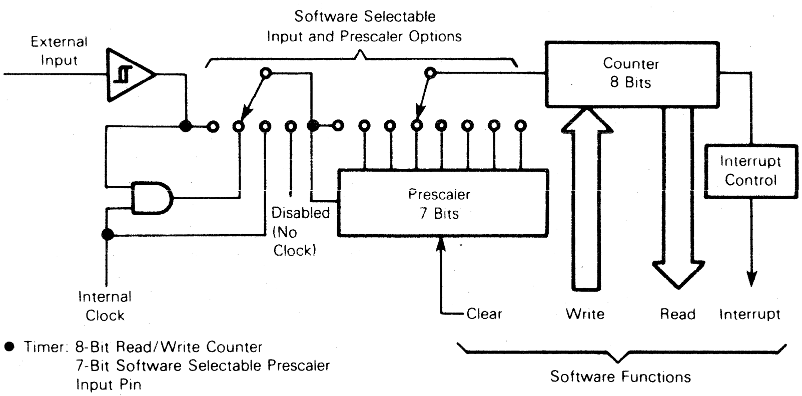
If TCR5=1 and TCR4=0, all inputs to the timer are disabled.

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



- Timer: 8-Bit Read/Write Counter
  7-Bit Software Selectable Prescaler
  Input Pin
  Timer Interrupt

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.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$

$$\text{Address Bus High} \leftarrow K;\ \text{Address Bus Low} \leftarrow X + (PC + 1)$$

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.

$$EA = X + [(PC + 1):(PC + 2)];\ PC \leftarrow PC + 3$$

$$\text{Address Bus High} \leftarrow (PC + 1) + K;$$

$$\text{Address Bus Low} \leftarrow X + (PC + 2)$$

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.

$$EA = PC + 2 + (PC + 1);\ PC \leftarrow EA \text{ if branch taken;}$$

$$\text{otherwise, } PC \leftarrow PC + 2$$

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 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.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

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 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.

$$EA1 = (PC + 1)$$

$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

$$EA2 = PC + 3 + (PC + 2);\ PC \leftarrow EA2 \text{ if branch taken;}$$

$$\text{otherwise, } PC \leftarrow PC + 3$$

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

	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	
Low \ Hi	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 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
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

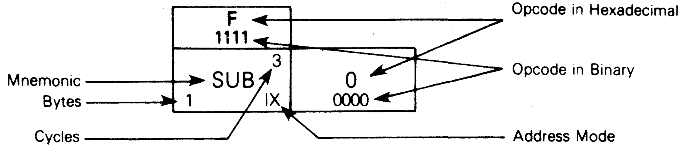


TABLE 10 — INSTRUCTION SET

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

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

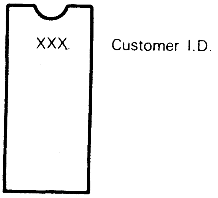
To initiate a ROM pattern for the MCU it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

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



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.

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

MOTOROLA

MC146805G2

Advance Information

8-BIT MICROCOMPUTER UNIT

The MC146805G2 Microcomputer Unit (MCU) belongs to the M6805 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

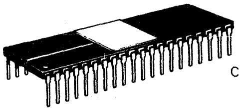
- Typical Full Speed Operating Power of 15 mW at 5 V
- Typical WAIT Mode Power of 4 mW
- Typical STOP Mode Power of 5 μW
- Fully Static Operation
- 112 Bytes of On-Chip RAM
- 2106 Bytes of On-Chip ROM
- 32 Bidirectional I/O Lines
- High Current Drive
- Internal 8-Bit Timer with Software Programmable 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

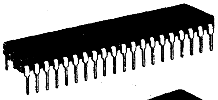
- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Most Self-Check Routines User Callable
- Two Power Saving Standby Modes

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

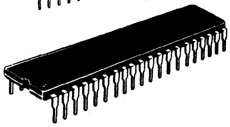
8-BIT
MICROCOMPUTER



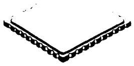
L SUFFIX
CERAMIC PACKAGE
CASE 715



S SUFFIX
CERDIP PACKAGE
CASE 734

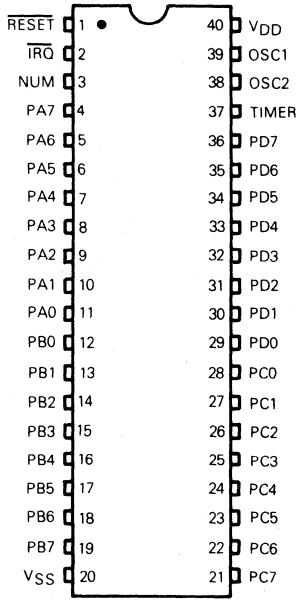


P SUFFIX
PLASTIC PACKAGE
CASE 711



Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS



ADI-864-R1

MAXIMUM RATINGS (Voltages Referenced to V_{SS})

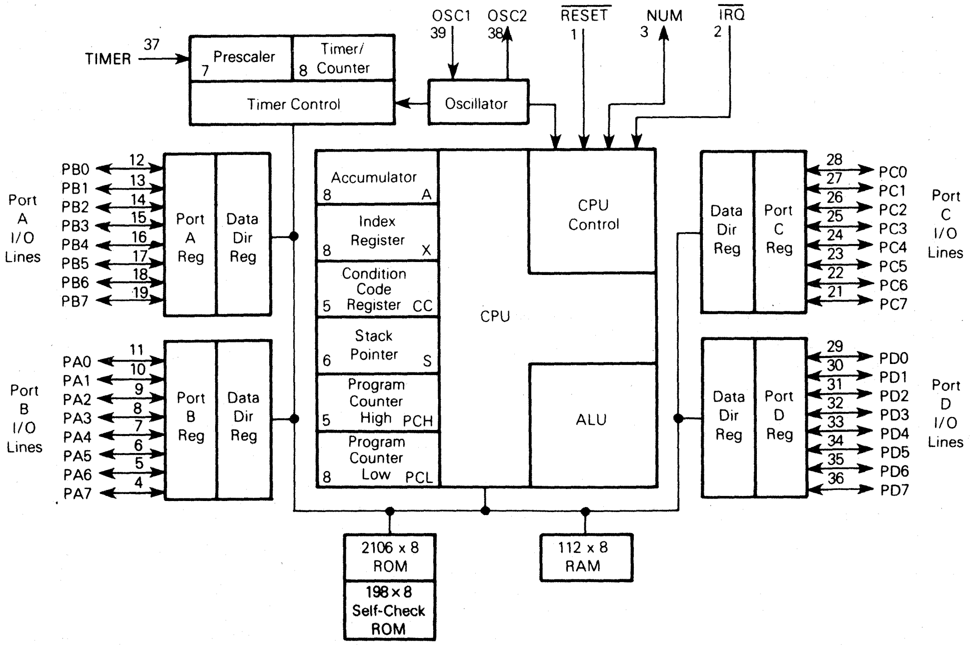
Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain Total (PD4-PD7 only)	I_{OH}	40	mA

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance			
Plastic	–	100	–
Cerdip	θ_{JA}	60	°C/W
Ceramic		50	
Chip Carrier		TBD	

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 $V_{out}) \leq V_{DD}$. 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



DC ELECTRICAL CHARACTERISTICS (See Note 2) ($V_{DD}=5.0$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=0°$ to 70°C unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage $I_{Load} \le 10.0\ \mu A$	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Output High Voltage				
($I_{Load}=-100\ \mu A$) 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 ($I_{Load}=800\ \mu A$) 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
TIMER, $\overline{\mathrm{IRQ}}$, $\overline{\mathrm{RESET}}$	V_{IH}	$V_{DD}-0.8$	V_{DD}	V
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, $t_{cyc}=1\ \mu s$) RUN (measured during self-check, $V_{IL}=0.2$ V, $V_{IH}=V_{DD}-0.2$ V)	I_{DD}	–	6	mA
WAIT (See Note 1)	I_{DD}	–	3	mA
STOP (See Note 1)	I_{DD}	–	250	µA
I/O Ports Input Leakage PA0-PA7, PB0- PB7, PC0-PC7, PD0-PD7	I_{IL}	–	±10	µA
Input Current $\overline{\mathrm{RESET}}$, $\overline{\mathrm{IRQ}}$, TIMER, OSC1	I_{in}	–	±1	µA
Capacitance				
Ports	C_{out}	–	12	pF
$\overline{\mathrm{RESET}}$, $\overline{\mathrm{IRQ}}$, TIMER, OSC1	C_{in}	–	8	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 $\overline{\mathrm{RESET}}$, $\overline{\mathrm{IRQ}}$, 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).

2. Electrical Characteristics for V_{DD} = 3 V available soon.

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 5)	t_{OXOV}	–	100	ms
Stop Recovery Startup Time (Crystal Oscillator) (Figure 6)	t_{ILCH}	–	100	ms
Timer Pulse Width (Figure 4)	t_{TH}, t_{TL}	0.5	–	t_{cyc}
Reset Pulse Width (Figure 5)	t_{RL}	1.5	–	t_{cyc}
Timer Period (Figure 4)	t_{TLTL}	1.0	–	t_{cyc}
Interrupt Pulse Width Low (Figure 15)	t_{ILIH}	1.0	–	t_{cyc}
Interrupt Pulse Period (Figure 15)	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.

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

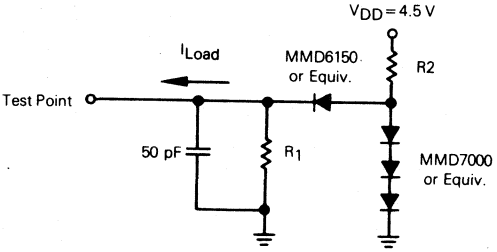


FIGURE 3 — TYPICAL OPERATING CURRENT vs INTERNAL FREQUENCY

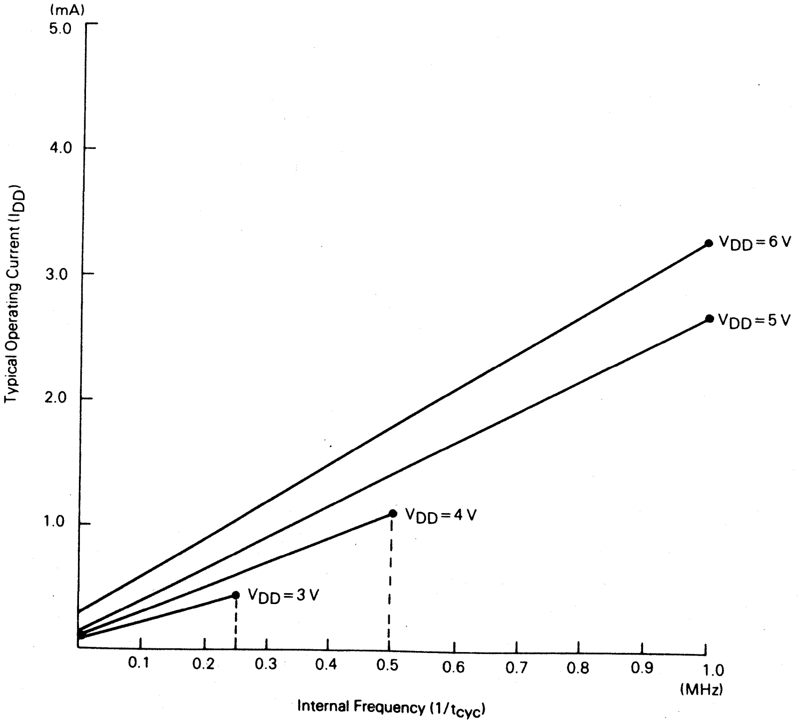


FIGURE 4 — TIMER RELATIONSHIPS

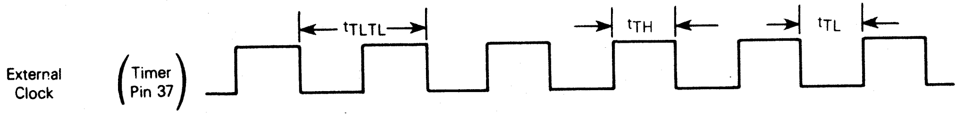
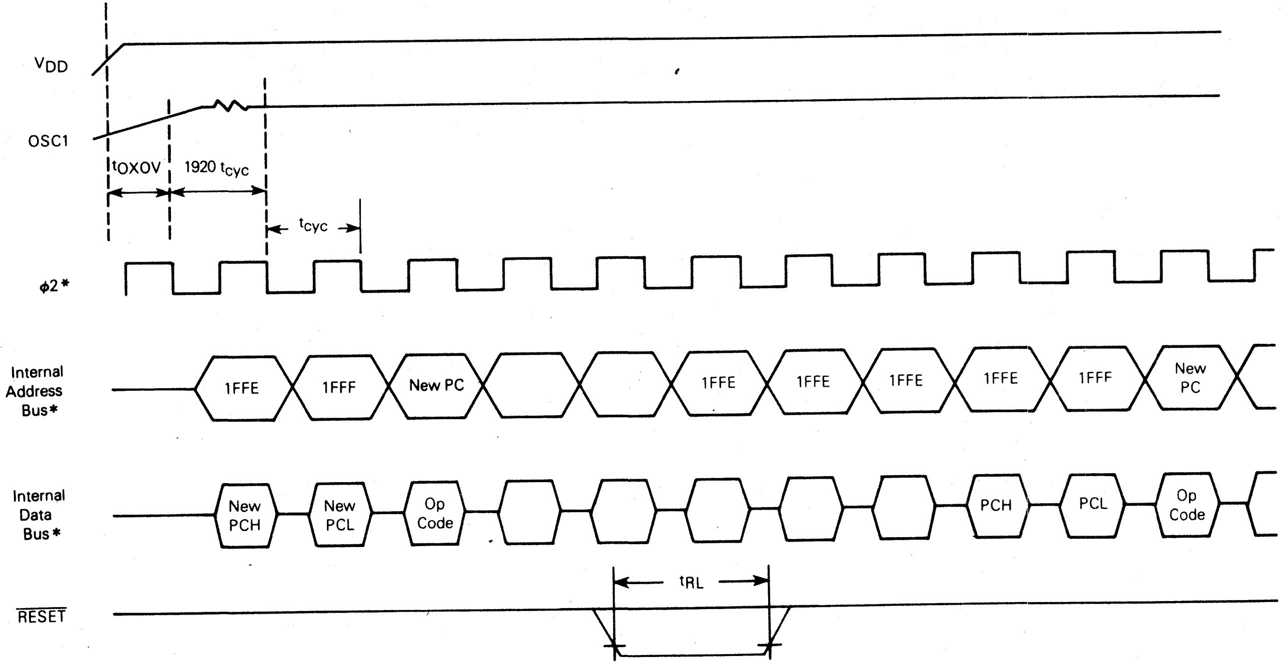
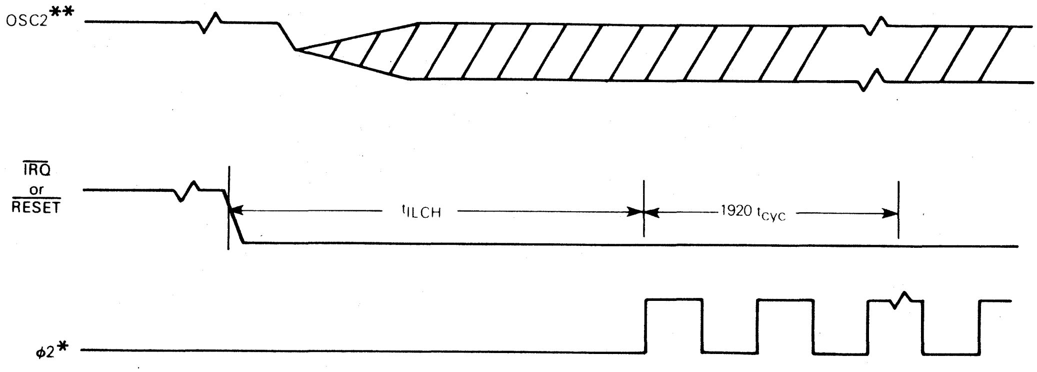


FIGURE 5 — POWER-ON RESET AND $\overline{\text{RESET}}$



* Internal timing signal not available externally.

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.

$\overline{IRQ}$ (MASKABLE INTERRUPT REQUEST)

$\overline{IRQ}$ is mask 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 mask 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 Reset section for a detailed description.

TIMER

The TIMER input may be used as an external clock for the on-chip timer. Refer to Timer section for a detailed description.

NUM — NON-USER MODE

This pin is intended for use in self-check only. User applications should 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. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the external 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 7(b). The relationship between R and f_{osc} is shown in Figure 8.

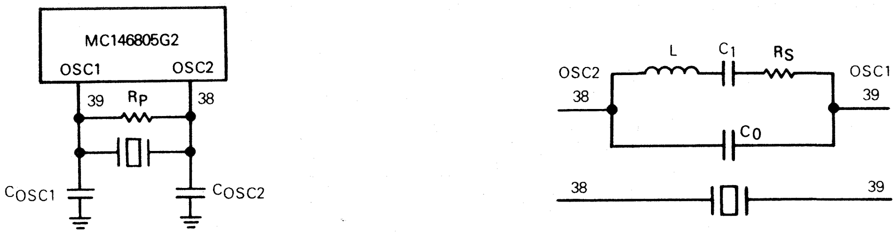
CRYSTAL — The circuit shown in Figure 7(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc} in the electrical characteristics table. Using an external CMOS oscillator 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 7(c). An external clock may be used with either the RC or crystal oscillator mask option. t_{OXOV} or t_{ILCH} do not apply when using an external clock input.

FIGURE 7 — 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

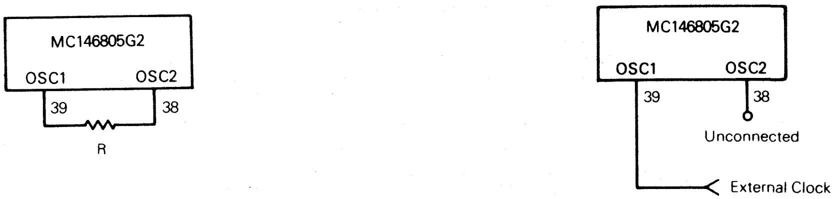
Crystal Parameters



Crystal Oscillator Connections

Equivalent Crystal Circuit

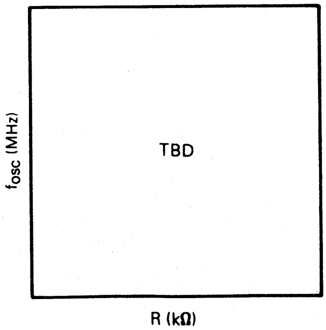
(a)



(b) RC Oscillator Connection

(c) External Clock Source Connections

FIGURE 8 — TYPICAL FREQUENCY vs RESISTANCE FOR RC OSCILLATOR OPTION ONLY



PA0-PA7

These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to Input/Output Programming section for a detailed description.

PB0-PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to Input/Output Programming section for a detailed description.

PC0-PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to the Input/Output Programming section for a detailed description.

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 the Input/Output Programming Section for a detailed description.

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 9 and Table 2.

FIGURE 9 — TYPCIAL PORT I/O CIRCUITRY

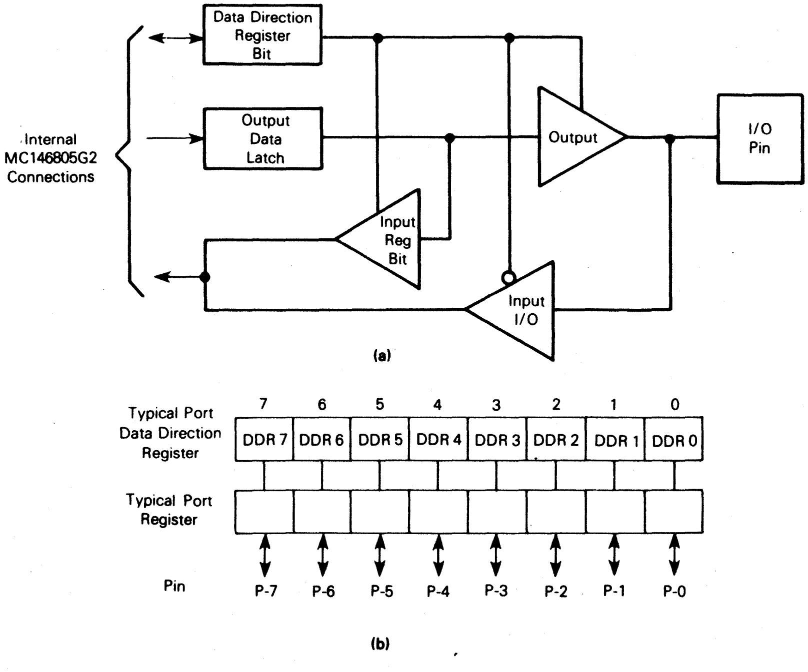


TABLE 2 — I/O PIN FUNCTIONS

$R/\overline{W}$	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.

SELF-CHECK

The MC146805G2 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, 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.

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 11.

FIGURE 10 — SELF-CHECK CIRCUIT

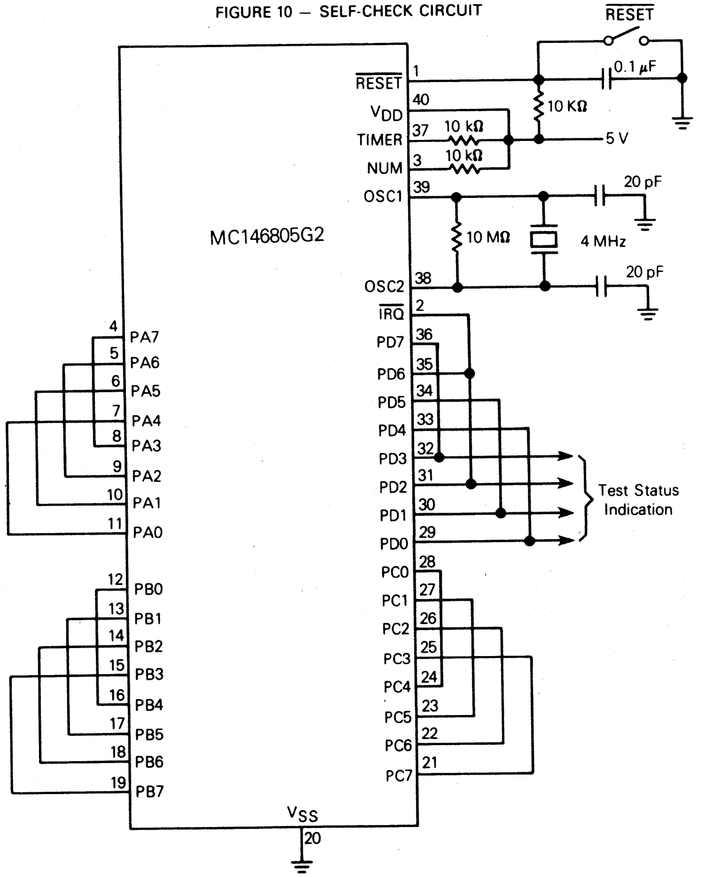
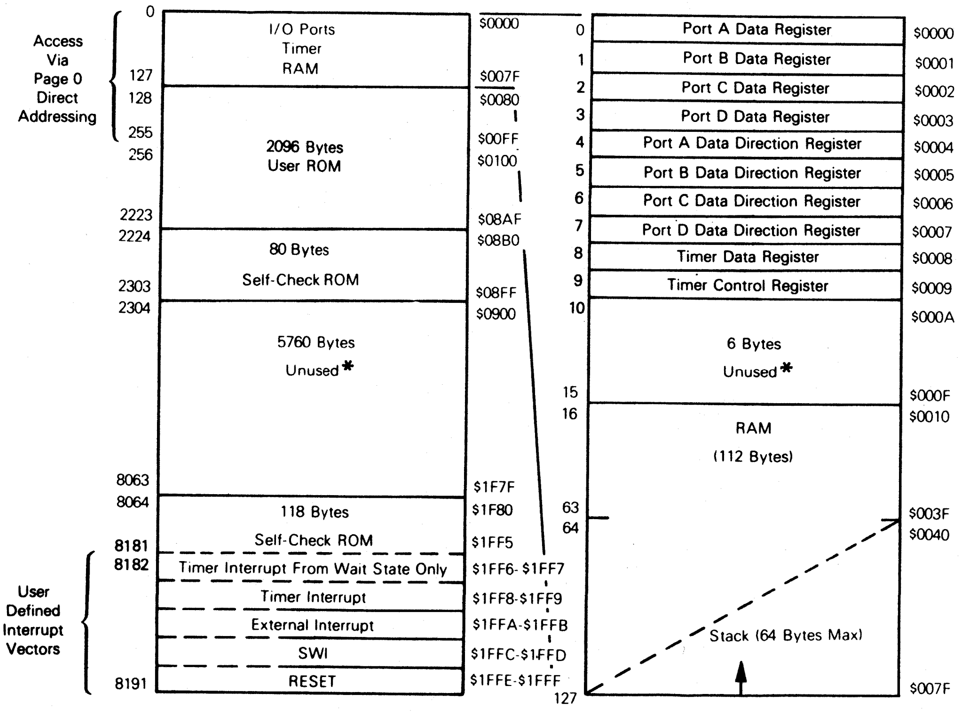


TABLE 3 — SELF-CHECK RESULTS

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
Any Cycling				Good Part
All Others				Bad Part

FIGURE 11 — ADDRESS MAP



*Reads of unused locations undefined.

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 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 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, 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 set 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

FIGURE 12 — PROGRAMMING MODEL

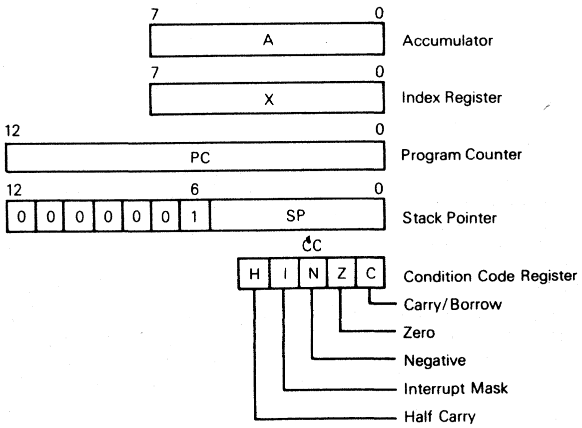
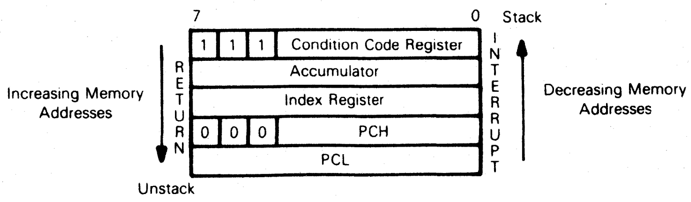


FIGURE 13 — STACKING ORDER



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.

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 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) – 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 "0".

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 MC146805G2 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_{cyc}. 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 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 a "0". All ports are defined as inputs.
- Stack pointer is set 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 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 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 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 stack 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

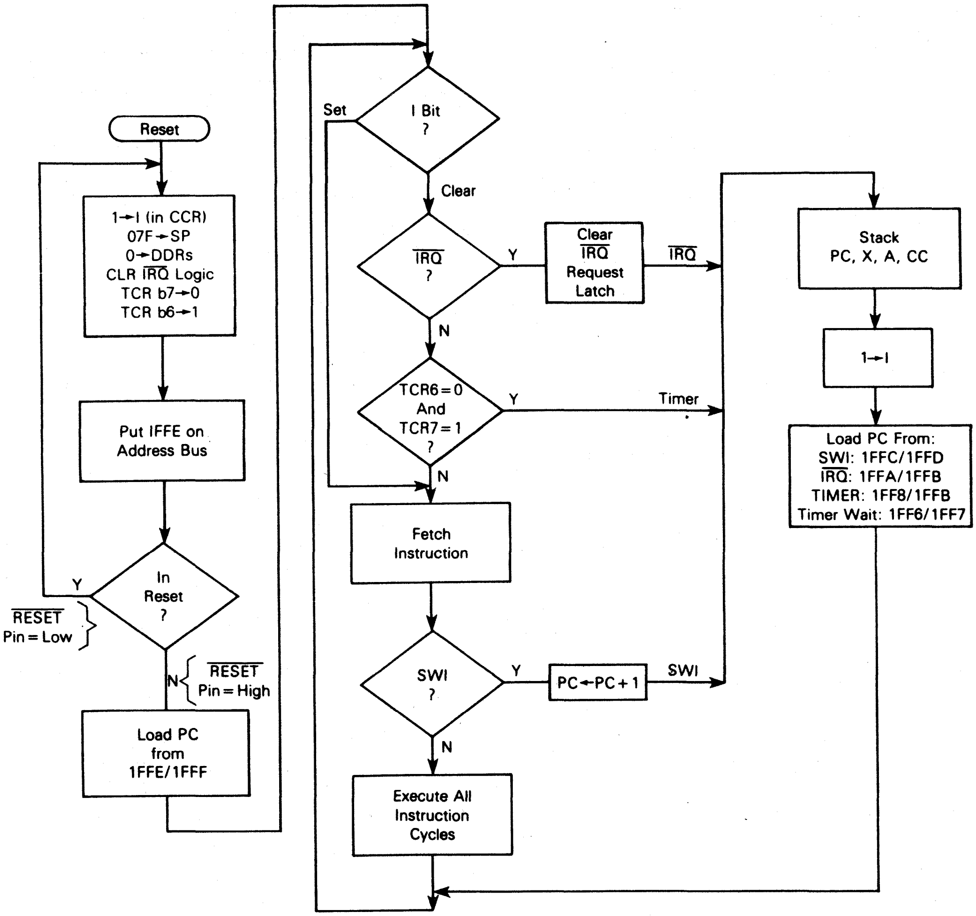
If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the 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

FIGURE 14 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART



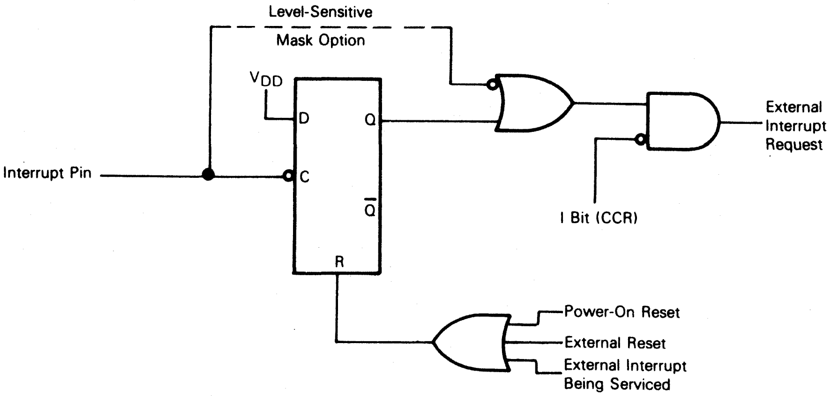
the service routine address is specified by the contents of \$1FFA and \$1FFB. Either a level- and edge-sensitive (or edge-sensitive only) are available as mask options. Figure 15 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the 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)

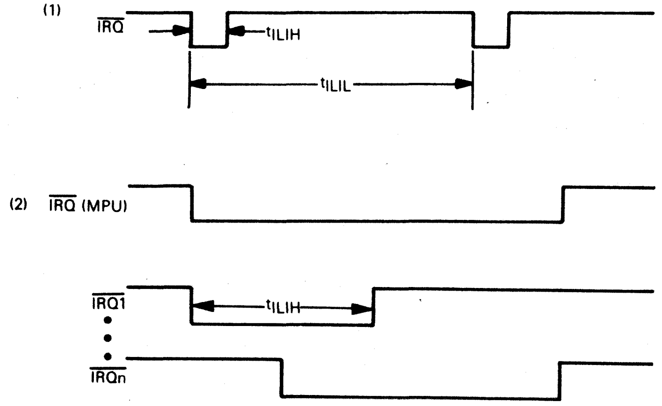
The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory locations \$1FFC and \$1FFD. See Figure 14 for interrupt and instruction processing flowchart.

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.

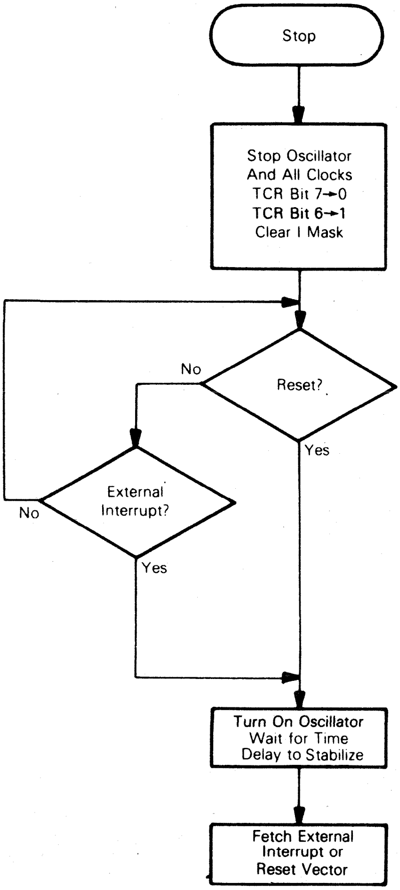
LOW-POWER MODES

STOP

The STOP instruction places the MC146805G2 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 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 16 — 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 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 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 with a 7-bit software selectable prescaler. Figure 18 contains a block diagram of the timer. The counter may be present 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.

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 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).

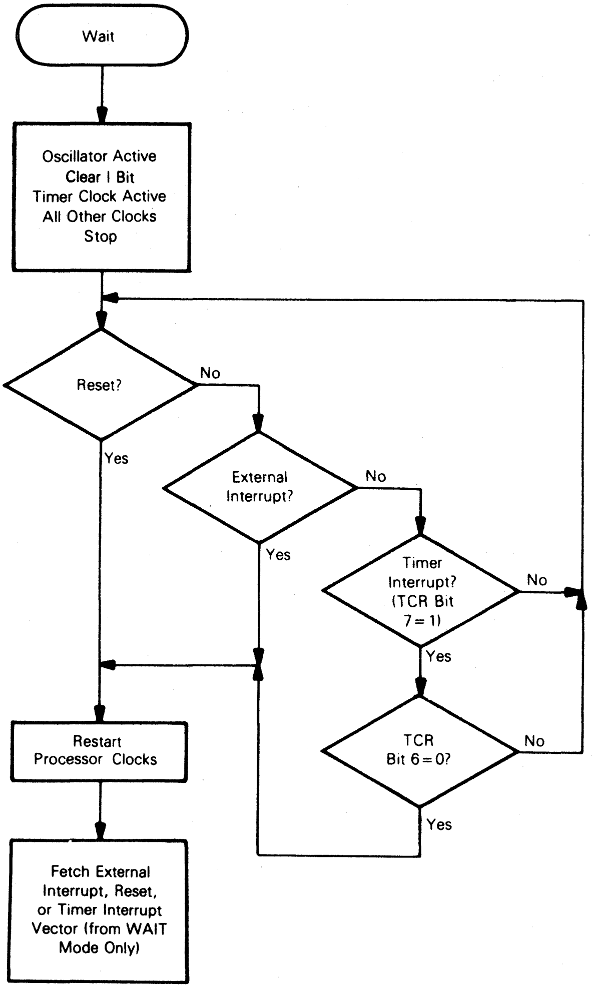
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals "1". 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, 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 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.

FIGURE 17 — WAIT FUNCTION FLOWCHART



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

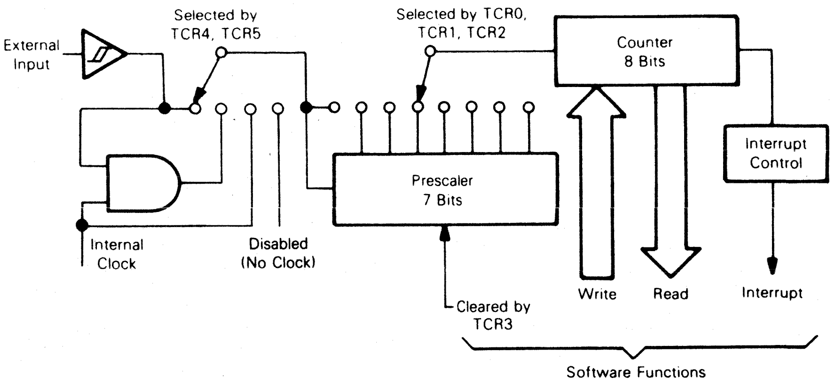
If TCR4=0 and TCR5=1, then all inputs to the timer are disabled.

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 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.

FIGURE 18 — TIMER BLOCK DIAGRAM



NOTES:

1. Prescaler and 8-bit counter are clocked on the falling edge of the internal clock or external input.
2. Counter 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 (period = t_{cyc}).

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 a "0". (Unaffected by $\overline{\text{RESET}}$.)

TCR2, TCR1, TCR0 — Prescaler select 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 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 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.

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. 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, which 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 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.

$$EA = X;\ PC \leftarrow PC+1$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow X$$

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 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 bite. The content of the index register is not changed.

$$EA = X + [(PC+1):(PC+2)];\ PC \leftarrow PC+3$$

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.

$$EA = PC + 2 + (PC+1);\ PC \leftarrow EA \text{ if branch taken; otherwise, } PC \leftarrow PC+2$$

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 to 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 ← 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 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)

$$EA2 = PC + 3 + (PC+2);\ PC \leftarrow EA2 \text{ if branch taken; otherwise, } PC \leftarrow PC+3$$

TABLE 4 — REGISTER/MEMORY INSTRUCTIONS

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		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

Function	Mnemonic	Addressing Modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		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 Op Code	 # Bytes	 # Cycles	 Bit Test and Branch 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 — M146805 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	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEG 1 INH	3 NEG 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1 0001
2 0010	5 BRSET1 3 BTB	5 BSET1 2 BSC	3 BHI 2 REL								2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2 0010
3 0011	5 BRCLR1 3 BTB	5 BCLR1 2 BSC	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3 0011
4 0100	5 BRSET2 3 BTB	5 BSET2 2 BSC	3 BCC 2 REL	5 LSR 2 DTR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4 0100
5 0101	5 BRCLR2 3 BTB	5 BCLR2 2 BSC	3 BCS 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5 0101
6 0110	5 BRSET3 3 BTB	5 BSET3 2 BSC	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6 0110
7 0111	5 BRCLR3 3 BTB	5 BCLR3 2 BSC	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7 0111
8 1000	5 BRSET4 3 BTB	5 BSET4 2 BSC	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8 1000
9 1001	5 BRCLR4 3 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B 1011
C 1100	5 BRSET6 3 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E 1110
F 1111	5 BRCLR7 3 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 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

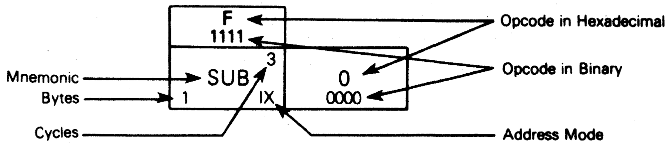


TABLE 10 – INSTRUCTION SET

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

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 MCM2708s

MDOS disk file

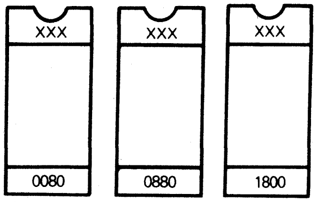
To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person, or your local Motorola representative.

EPROMs

The MCM2708 or MCM2716 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. See Figure A-1 for recommended marking procedure.

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.

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

MOTOROLA

MC146818

Advance Information

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.

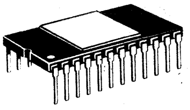
- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel 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 ($\overline{\text{IRQ}}$)
- 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

CMOS

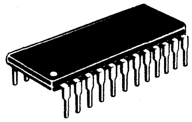
(HIGH-PERFORMANCE
SILICON-GATE COMPLEMENTARY MOS)

**REAL-TIME CLOCK
PLUS RAM**

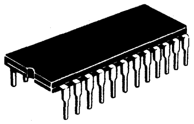
L SUFFIX
CERAMIC PACKAGE
CASE 716



P SUFFIX
PLASTIC PACKAGE
CASE 709

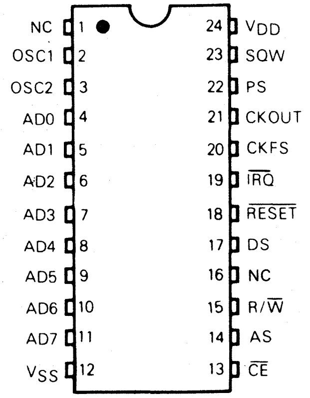


S SUFFIX
CERDIP PACKAGE
CASE 623



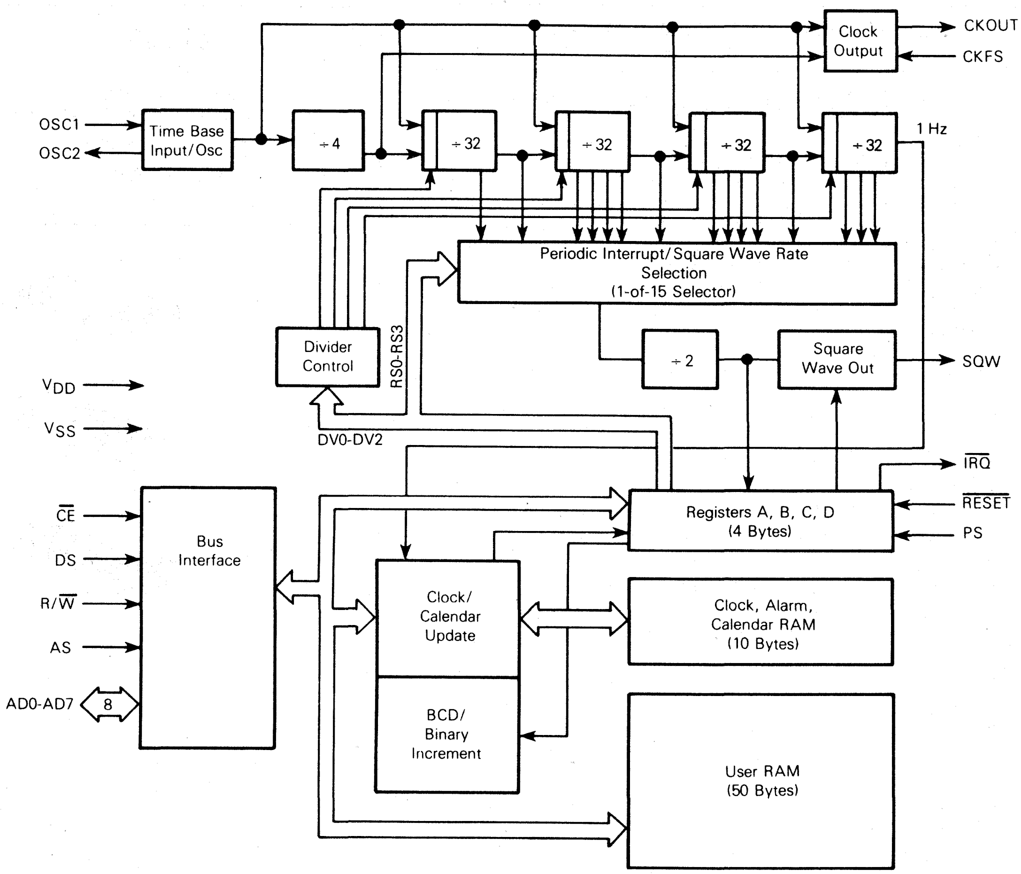
CHIP CARRIER
ALSO AVAILABLE

PIN ASSIGNMENTS



ADI-856-R1

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	θ_{JA}	120	°C/W
Cerdip		65	
Ceramic		50	

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 $V_{out}) \leq V_{DD}$. 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 ($V_{DD}=5$ 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 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}	–	4	mA
$f_{osc}=1.048516$ MHz	I_{DD2}	–	1.5	mA
$f_{osc}=32.768$ kHz	I_{DD3}	–	100	μA
I_{DD} – Quiescent $f_{osc}=$ DC; OSC1 = DC; All Other Inputs = $V_{DD}-0.2$ V; No Clock	I_{DD4}	–	100	μA
Output High Voltage ($I_{Load}=-1.6$ mA, All Outputs Except $\overline{IRQ}$) ($I_{Load}=-1.0$ mA, SQW)	V_{OH}	4.1	–	V
Output Low Voltage ($I_{Load}=1.6$ mA, All Outputs Except OSC2) ($I_{Load}=1.0$ 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 — AD0-AD7	I_{TSL}	–	±10	μA

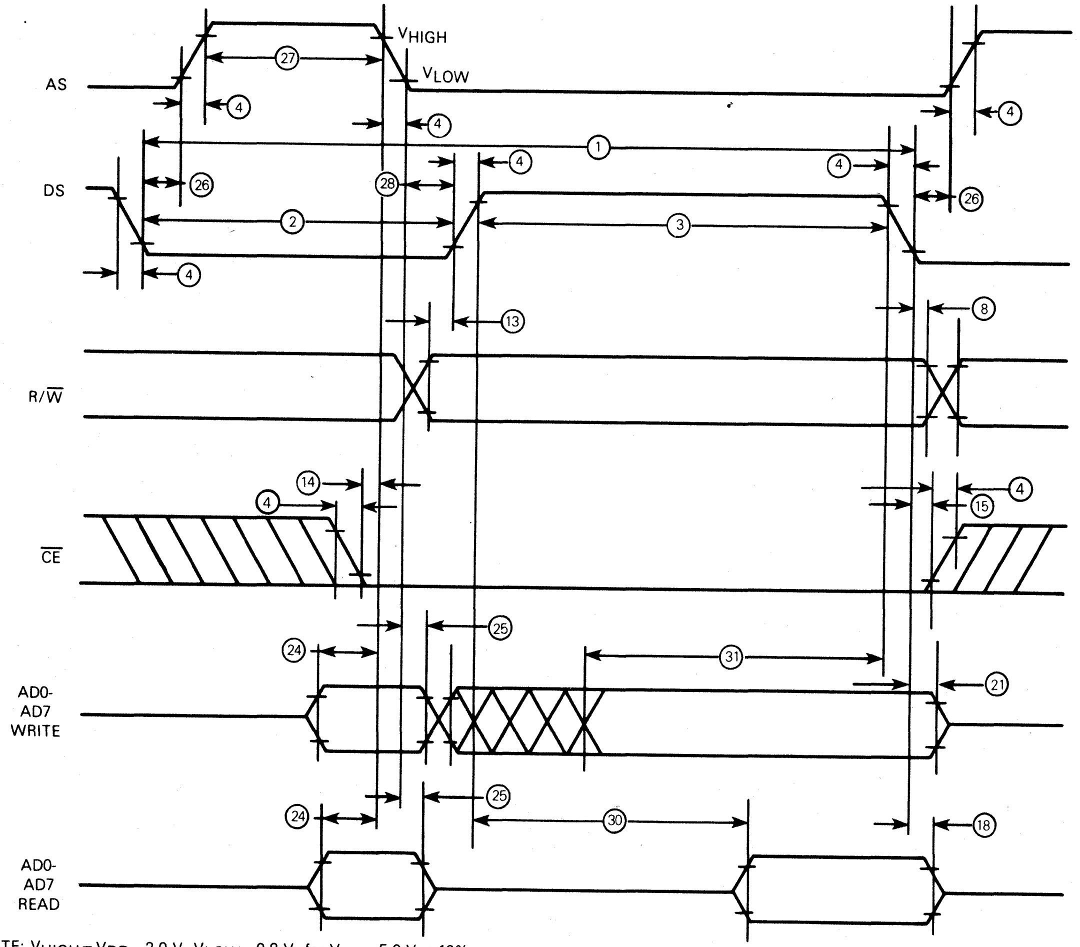
BUS TIMING ($V_{DD}=5$ Vdc ±10%, $V_{SS}=0$ Vdc, $T_A=T_L$ to T_H unless otherwise noted)

Ident. Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	953	DC	ns
2	Pulse Width, DS/E Low or $\overline{RD}$/$\overline{WR}$ High	PW_{EL}	300	–	ns
3	Pulse Width, 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/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	165	–	ns
14	Chip Enable Setup Time Before AS/ALE Fall	t_{CS}	55	*	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}	50	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	50	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	135	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	90	–	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}	200	–	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.

*Refer to **IMPORTANT NOTICES** appearing on last page 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

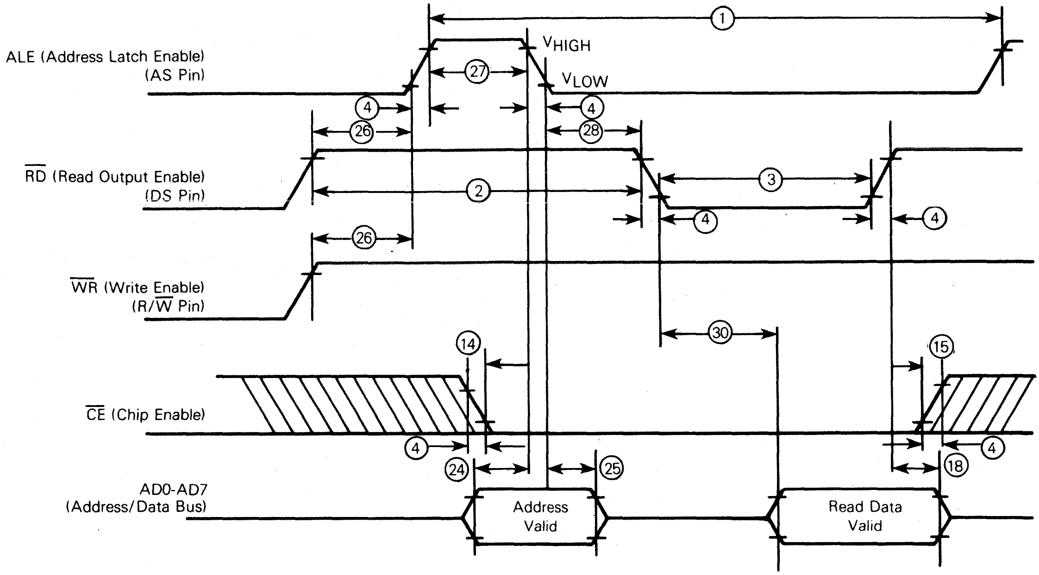
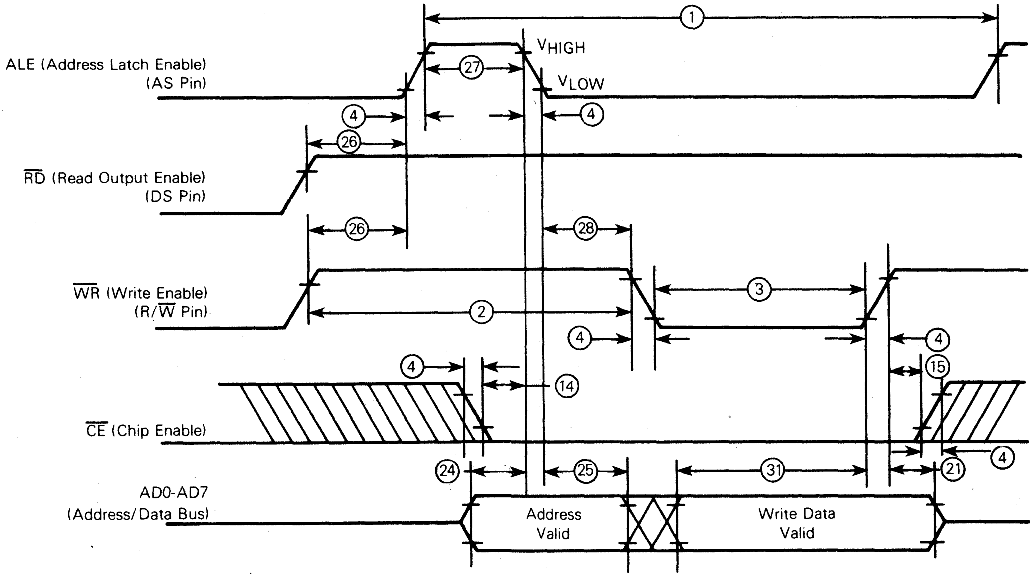


FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS

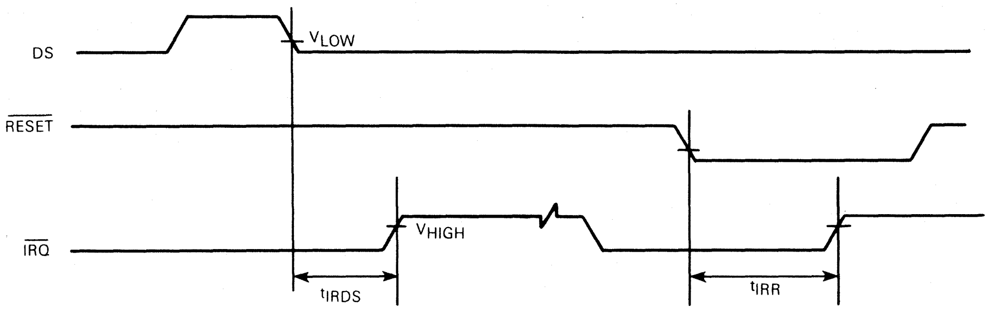


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

TABLE 1 — SWITCHING CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

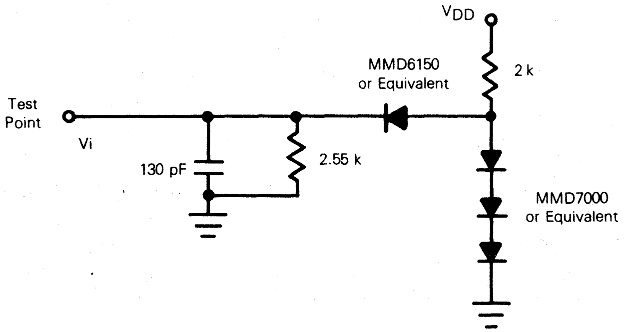
Description	Symbol	Min	Max	Unit
Oscillator Startup	t_{RC}	—	100	ms
Reset Pulse Width	t_{RWL}	5	—	μs
Reset Delay Time	t_{RLH}	5	—	μs
Power Sense Pulse Width	t_{PWL}	5	—	μs
Power Sense Delay Time	t_{PLH}	5	—	μs
$\overline{\text{IRQ}}$ Release from DS	t_{IRDS}	—	2	μs
$\overline{\text{IRQ}}$ Release from $\overline{\text{RESET}}$	t_{IRR}	—	2	μs
VRT Bit Delay	t_{VRTD}	—	2	μs

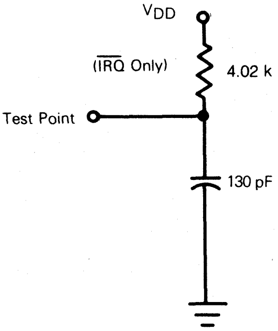
FIGURE 5 — $\overline{\text{IRQ}}$ RELEASE DELAY



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

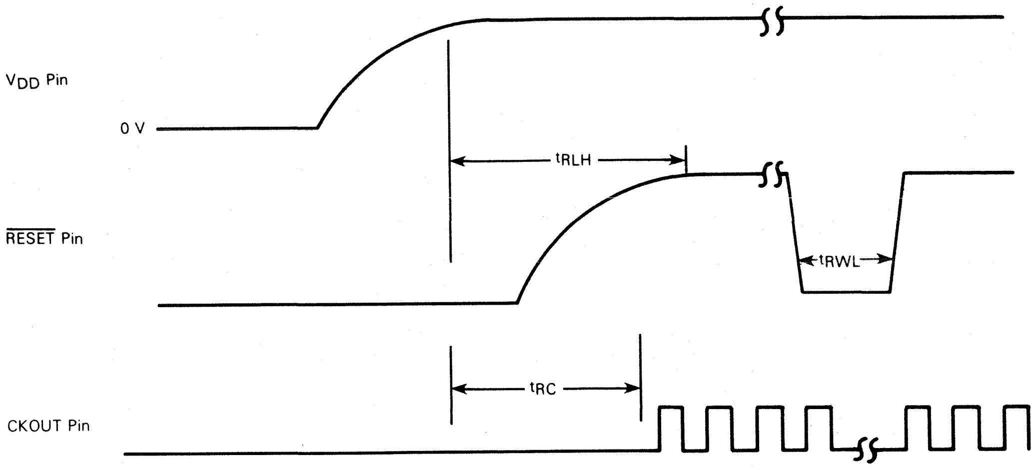
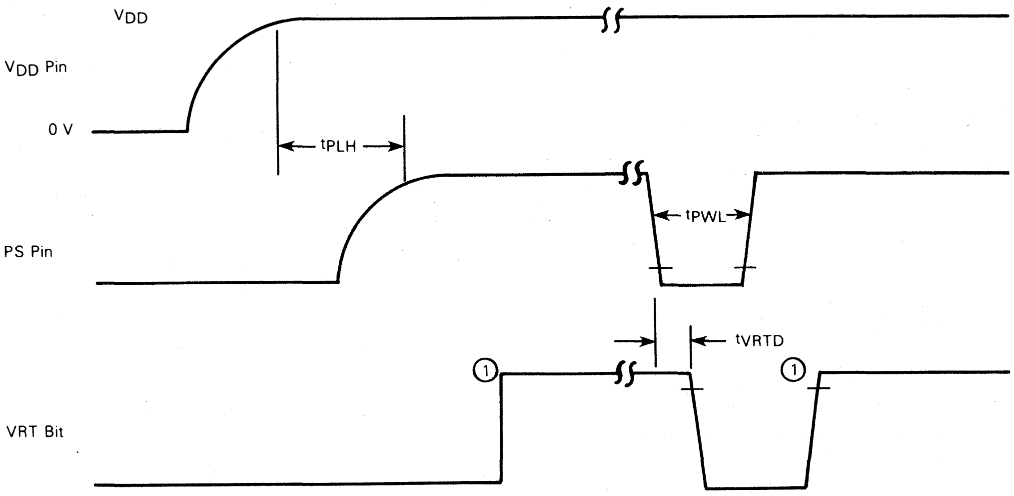


FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT



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.

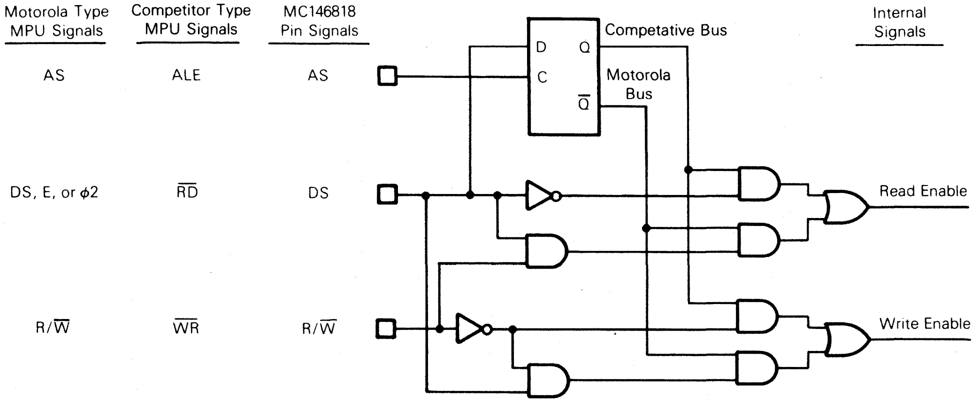
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated by the Motorola MC6800 and the other by the Intel 8080 and its companion part, the 8228.

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

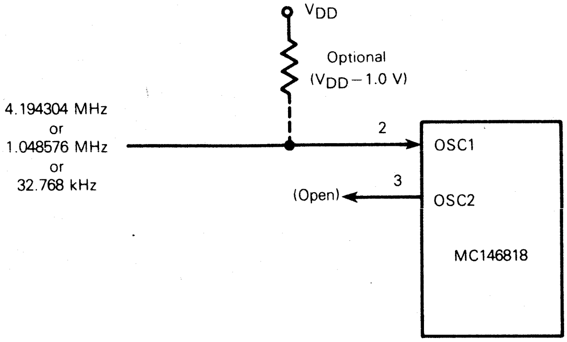


FIGURE 11 — CRYSTAL OSCILLATOR CONNECTION

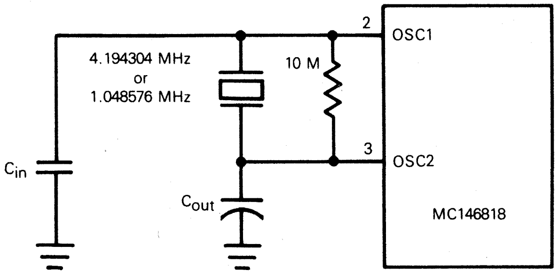
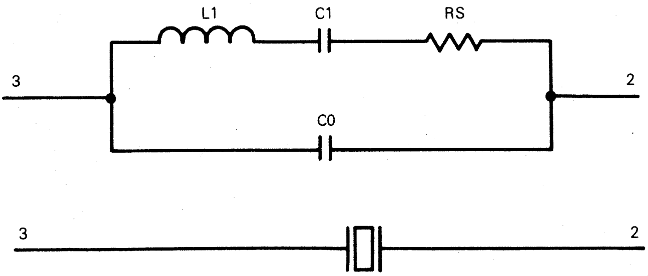


FIGURE 12 — CRYSTAL PARAMETERS

Crystal Equivalent Circuit



f_{osc}	4.194304 MHz	1.048576 MHz
Rs max	75 Ω	400 Ω
C0 max	7 pF	5 pF
C1	0.012 pF	0.008 pF
C_{in}/C_{out}	15-30 pF	15-40 pF
Q	50 k	35 k

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 $\phi2$ ($\phi2$ 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 (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 (in the other Motorola case of MOTEL) and during $\overline{RD}$ and $\overline{WR}$ (in the 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.

$\overline{IRQ}$ — 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.

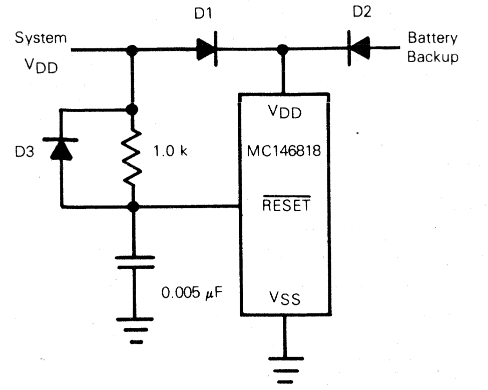
$\overline{RESET}$ — RESET, INPUT

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,

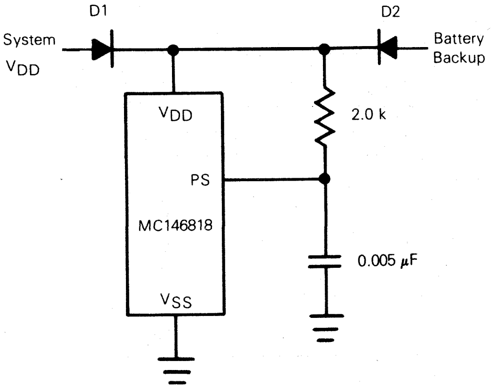
FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR $\overline{\text{RESET}}$



D1 = 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 = D2 = 1N4148 or Equivalent

g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{\text{IRQ}}$ 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 ($\overline{\text{CE}}$) pin controls all bus inputs (R/$\overline{\text{W}}$, DS, AS, AD0-AD7). $\overline{\text{CE}}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{\text{CE}}$ 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

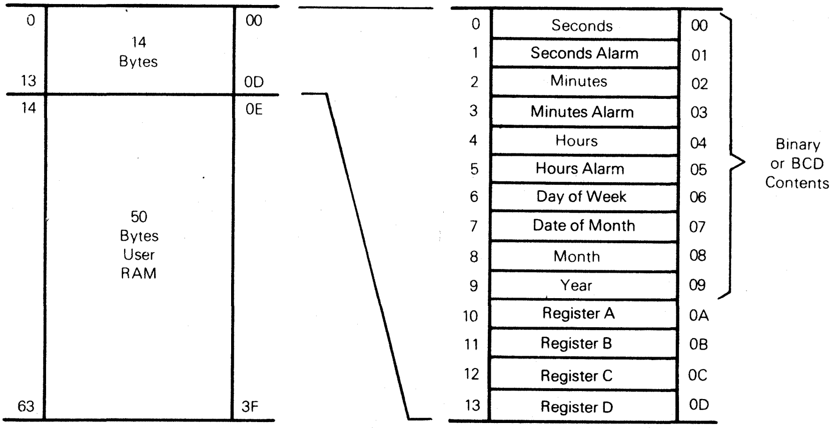


TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

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	$01-10C (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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ pin is asserted low. $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 t_{PI}	SQW Output Frequency	Periodic Interrupt Rate t_{PI}	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.

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

b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

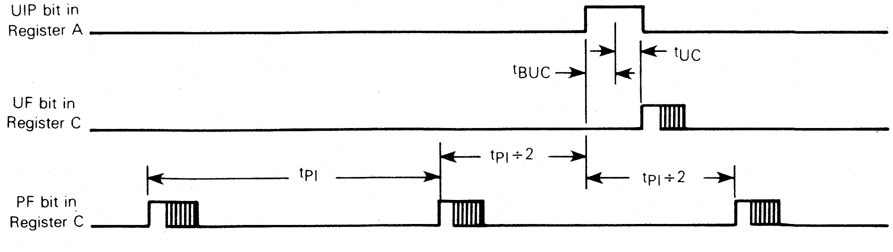
Read/Write Register except UIP

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 second later. These three read/write bits are not affected by 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 RESET.

REGISTER B ($0B)

MSB LSB

b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE

Read/Write Register

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 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 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 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 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 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 IRQ signal. The 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 IRQ. The 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 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 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

b7	b6	b5	b4	b3	b	b1	b0
IRQF	PF	AF	UF	0	0	0	0

Read-Only Register

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 IRQ pin is driven low. All flag bits are cleared after Register C is read by the program or when the 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 IRQ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a 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 IRQ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A 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 IRQ. UF is cleared by a Register C read or a 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
VRT	0	0	0	0	0	0	0

Read Only 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."

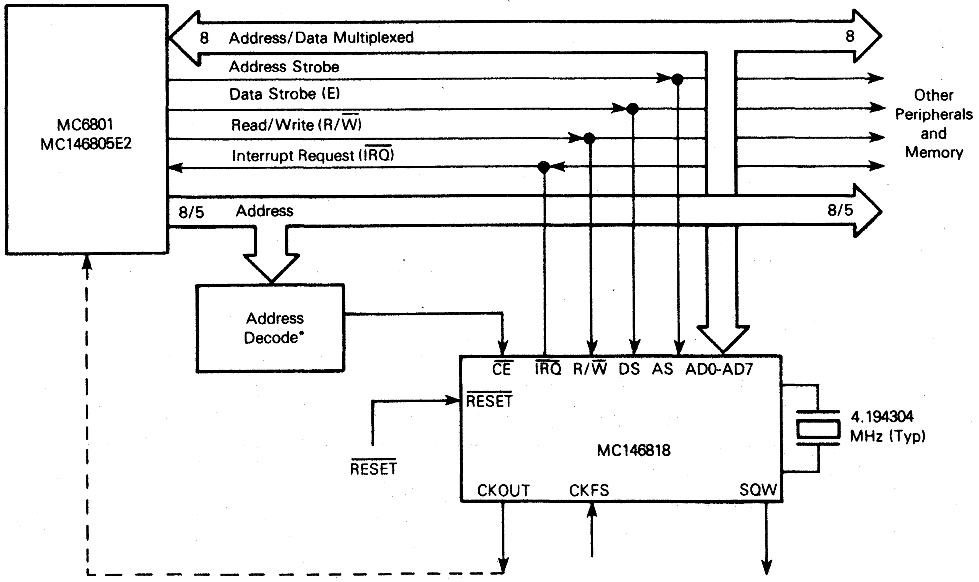
TYPICAL INTERFACING

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 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

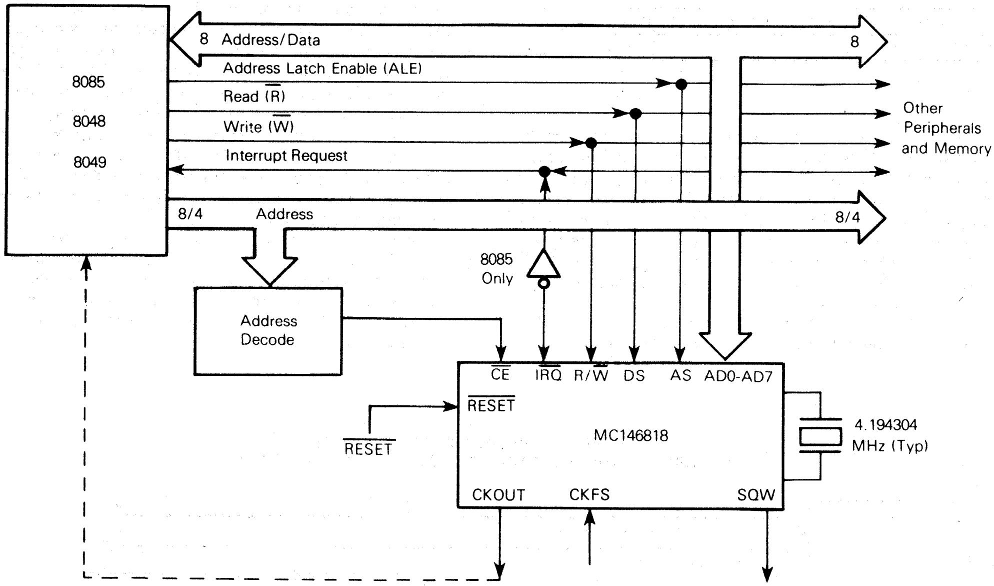
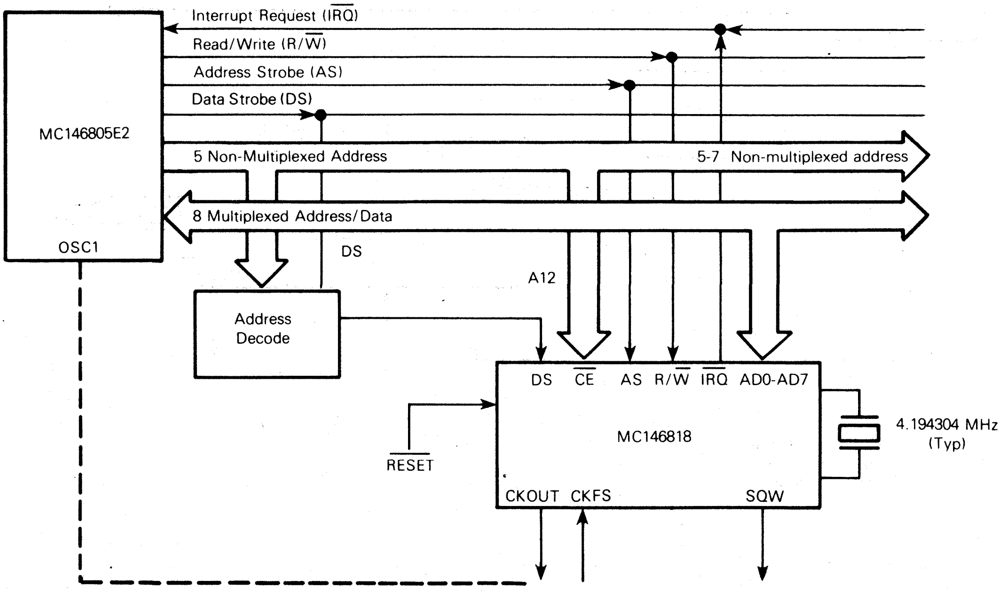


FIGURE 19 — MC146818 INTERFACE WITH MC146805E2 CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING



This illustrates the use of CMOS gating for address decoding.

FIGURE 20 — MC146818 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER

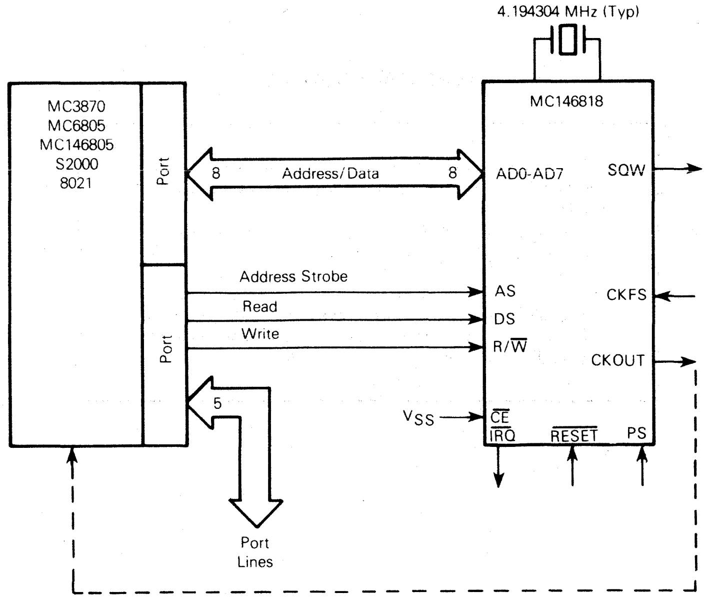
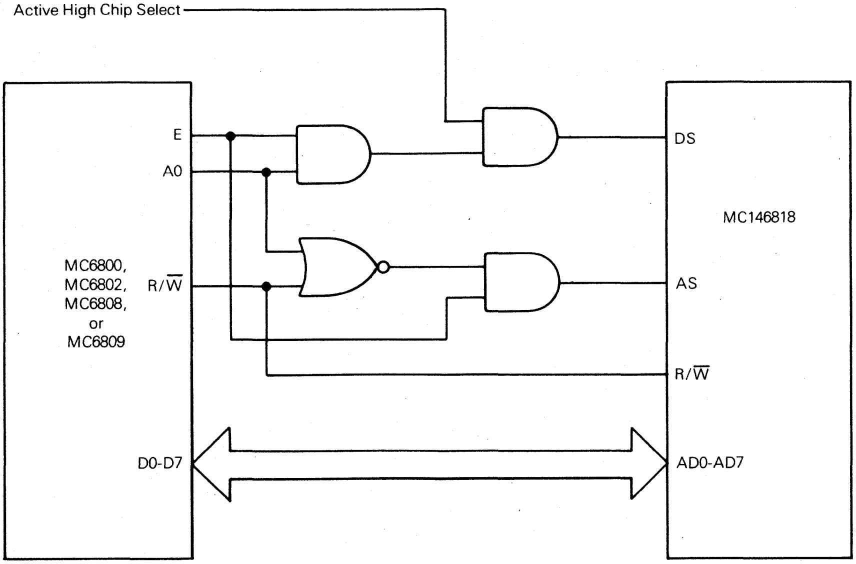


FIGURE 21 — MC146818 INTERFACED WITH MOTOROLA PROCESSORS



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.

MOTOROLA

MC146823

CMOS

(HIGH-DENSITY HIGH-PERFORMANCE SILICON-GATE)

PARALLEL INTERFACE

Product Preview

CMOS PARALLEL INTERFACE

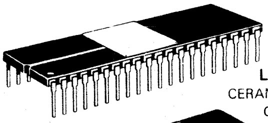
The MC146823 CMOS parallel interface (CPI) provides a universal means of interfacing external signals with the MC146805E2 CMOS microprocessor and other multiplexed bus microprocessors. The unique MOTEL circuit on-chip allows direct interfacing to most industry CMOS microprocessors, as well as many NMOS 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:

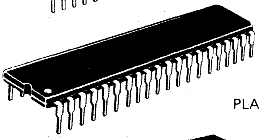
- 24 Individually Programmed I/O Pins
- MOTEL Circuit for Bus Compatibility with Many Microprocessors
- Multiplexed Bus Compatibility with: MC146805E2, MC6801, MC6803, and Competitive Microprocessors
- Data Direction Registers for Ports A, B, and C
- Four Port C I/O Pins May Be Used as Control Lines 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 V to 5.5 V Operating V_{DD}

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

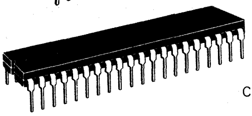
Package Type	Order Number
Plastic — P Suffix	MC146823P
Ceramic (Side Brazed) — L Suffix	MC146823L
Cerdip — S Suffix	MC146823S
Chip Carrier — Z Suffix	MC146823Z



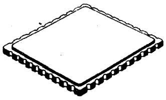
L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

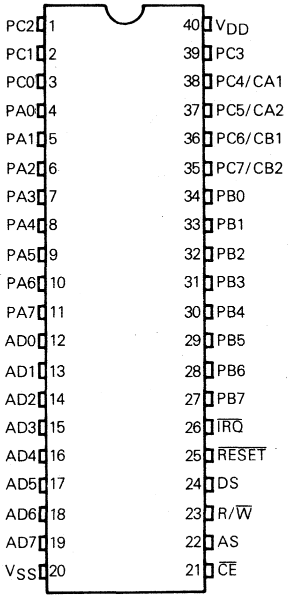


S SUFFIX
CERDIP PACKAGE
CASE 734



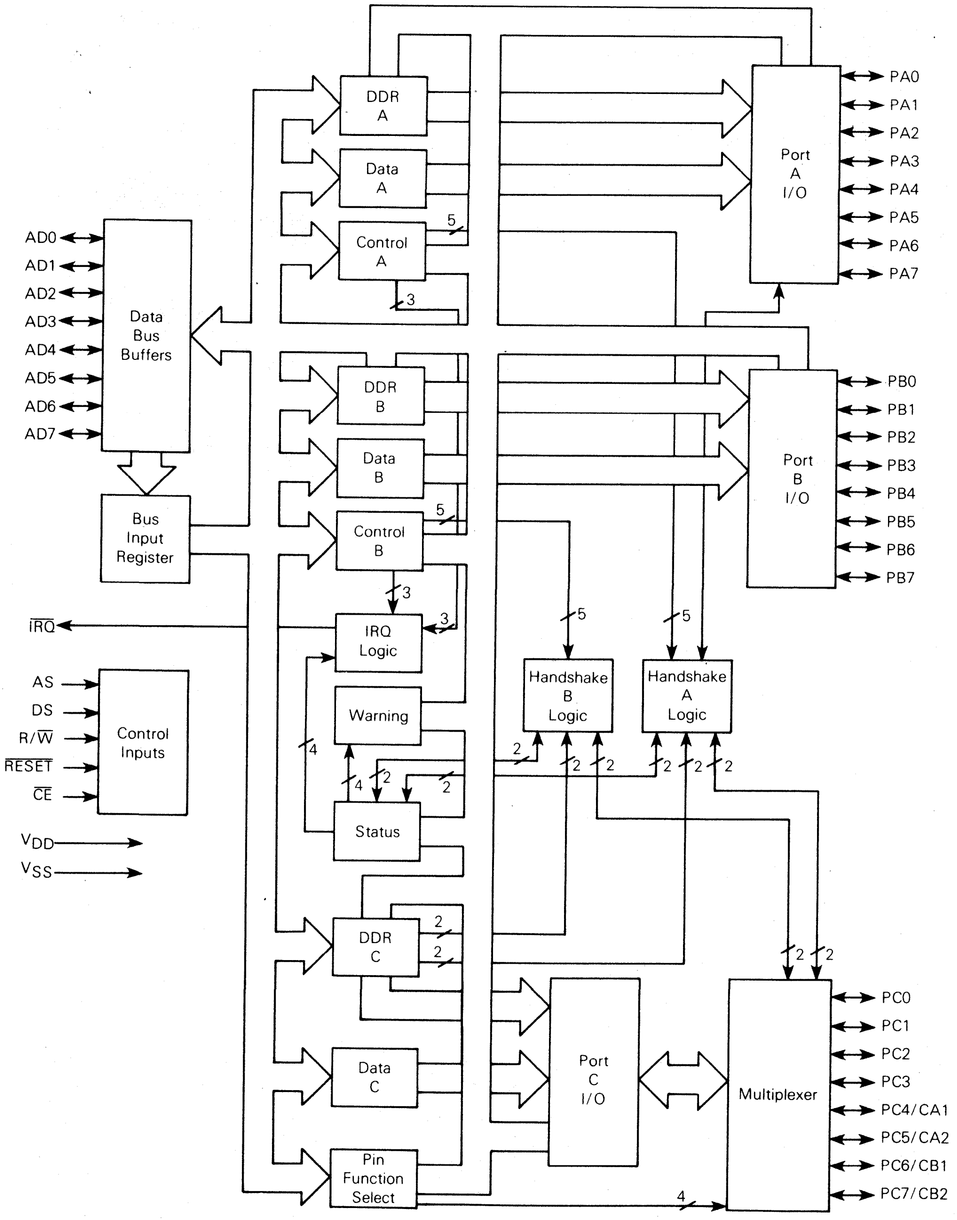
Z SUFFIX
40-PIN
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS



NP-337-R1

BLOCK DIAGRAM



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; 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.

The microprocessor type is automatically selected by the MOTEL circuit through latching the state of the DS/RD pin with AS/ALE. Since DS is always low during AS and RD is always high during ALE, the latch automatically indicates with which type microprocessor bus it is interfaced.

PIN DESCRIPTIONS

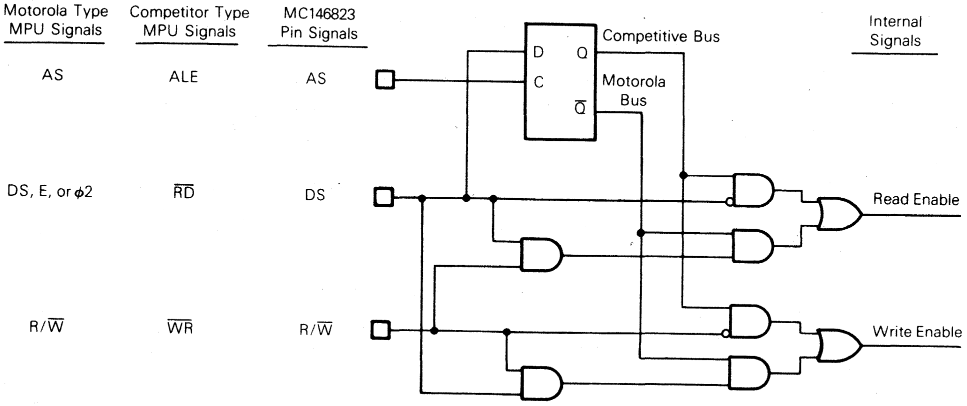
The following paragraphs contain a brief description of the input and output pins. References (if applicable) are given to other paragraphs that contain more detail about the function being performed.

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 for the multiplexed address valid time 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) for read data hold time after DS falls in the Motorola case of MOTEL or after $\overline{RD}$ rises in the other case.

FIGURE 2 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



MULTIPLEXED 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 $\overline{\text{RD}}$ signifies the time that the CPI is to drive the bidirectional bus. In write cycles, the trailing edge of DS or rising edge of $\overline{\text{WR}}$ causes the parallel interface to latch the written data present on the bidirectional bus.

The second MOTEL interpretation of DS is that of $\overline{\text{RD}}$, 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/$\overline{\text{W}}$)

The MOTEL circuit treats the R/$\overline{\text{W}}$ input pin in one of two ways. First, when a Motorola microprocessor is connected, R/$\overline{\text{W}}$ 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/$\overline{\text{W}}$ while DS is high, whereas a write cycle is a low on R/$\overline{\text{W}}$ while DS is high.

The second interpretation of R/$\overline{\text{W}}$ is as a negative write pulse, $\overline{\text{WR}}$, $\overline{\text{MEMW}}$, and $\overline{\text{I/OW}}$ from competitor's microprocessors. The MOTEL circuit in this mode gives the R/$\overline{\text{W}}$ pin the same meaning as the write ($\overline{\text{W}}$) pulse on many generic RAMs.

CHIP ENABLE ($\overline{\text{CE}}$)

The $\overline{\text{CE}}$ input signal must be asserted (low) for the bus cycle in which the MC146823 is to be accessed. $\overline{\text{CE}}$ is not latched and must be stable prior to and during DS (in the Motorola case of MOTEL) and prior to and during $\overline{\text{RD}}$ and $\overline{\text{WR}}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{\text{CE}}$ cause no actions to take place within the MC146823. When $\overline{\text{CE}}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{\text{CE}}$ is high, all data, DS, and R/$\overline{\text{W}}$ inputs from the microprocessor are disconnected within the MC146823. This permits the MC146823 to be isolated from a powered-down microprocessor.

RESET ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ 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 ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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.

FIGURE 3 — TYPICAL PORT I/O CIRCUITRY

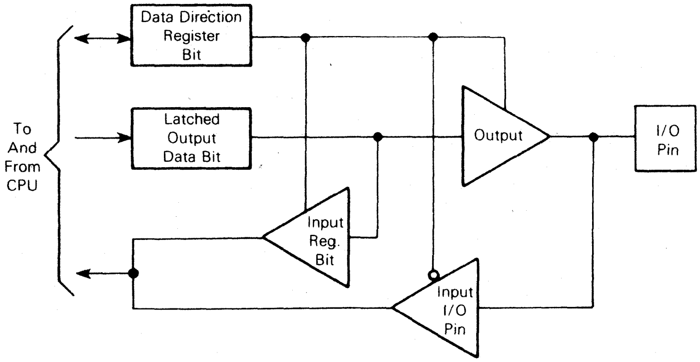


TABLE 1 — PORT DATA REGISTER ACCESSES

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.

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 **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 read and writes. P1DB and P2DB are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to any port B 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 read 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 ($\overline{\text{IRQ}}$) to go low. $\overline{\text{IRQ}}$ 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	$\overline{\text{IRQ}}$ 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.
See Control Register description for actual bit numbers.

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 3 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 ($\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ to go low when IRQF (interrupt flag) in the HSR is set to a logic one. $\overline{\text{IRQ}}$ 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.

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.

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 and 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 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 any port B register is latched into the port B output latch (see Figure 3) regardless of the state of the 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 AND WARNING BITS BY PORT A DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction
PDA	None	None	None
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.
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.

TABLE 5 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS AND WARNING BITS BY PORT B DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction
PDB	None	None	None
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.
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.

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 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 PC5/CB1. Transitions on port C pins not defined as handshake pins do not affect 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:

$$\begin{aligned}\text{Bit } 7 = \text{IRQF} = {} & [\text{HSB2} \bullet \text{CRB2}(3)] \\ & + [\text{HSA2} \bullet \text{CRA2}(3)] \\ & + [\text{HSB1} \bullet \text{CRB1}(0)] \\ & + [\text{HSA1} \bullet \text{CRA1}(0)]\end{aligned}$$

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
HSA2	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:

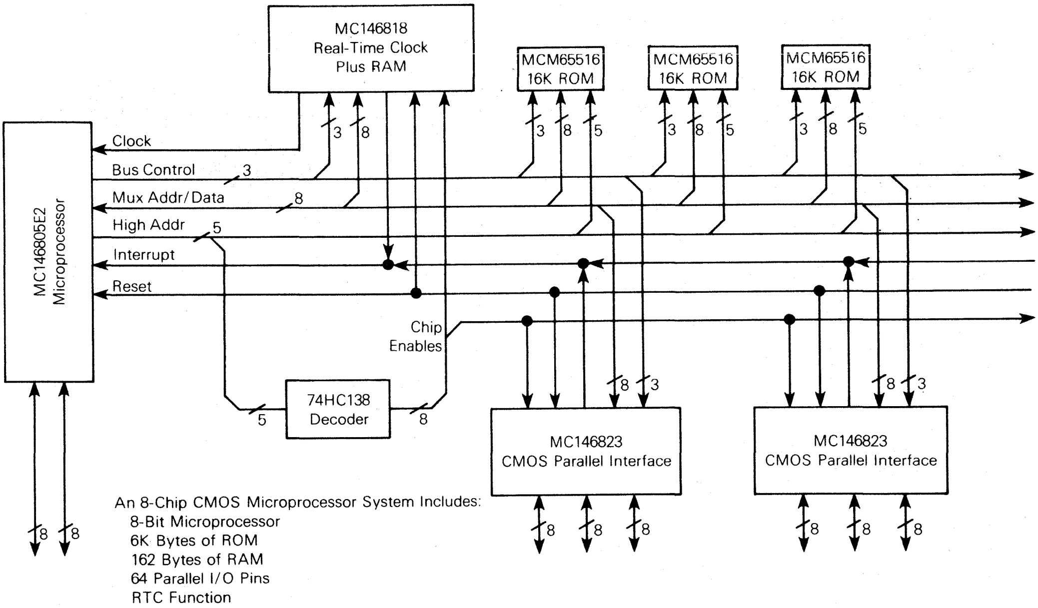
1. Read status register	(User determines which if any enabled handshake transition occurred)
2. Read/write port data indicated by status register	(Clears associated status bit and latches appropriate warning register bit in the buffer latch)
3. Read warning register	(Latched warning bit is cleared and 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.

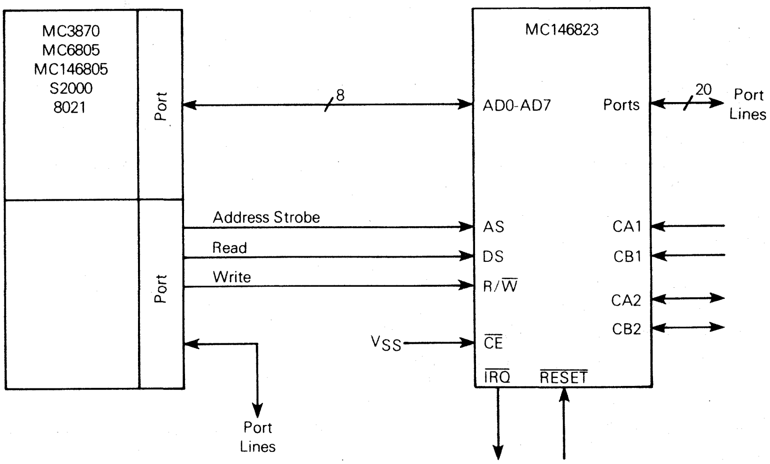
A single-chip microcomputer (MCU) may be interfaced with 11 port lines as shown in Figure 5. This interface also requires some software overhead to gain up to 13 additional I/O lines and the MC146823 handshake lines.

FIGURE 4 — A TYPICAL CMOS MICROPROCESSOR SYSTEM



An 8-Chip CMOS Microprocessor System Includes:
- 8-Bit Microprocessor
- 6K Bytes of ROM
- 162 Bytes of RAM
- 64 Parallel I/O Pins
- RTC Function

FIGURE 5 — MC146823 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE-CHIP MICROCOMPUTER



MOTOROLA

MC1468705F2

CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT EPROM MICROCOMPUTER

Product Preview

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.

HARDWARE FEATURES

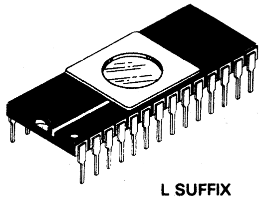
- Low Power Wait Mode
- Typical Stop Mode Power of 25 μW
- 8-Bit Architecture
- Fully Static Operation
- Single 3- to 5.5-Volt Supply
- 1080 Bytes of On-Chip User EPROM
- 64 Bytes of On-Chip RAM
- Memory Mapped I/O
- 16 Bidirectional I/O Lines
- Four Input-Only Lines
- Internal 8-Bit Timer with Software 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

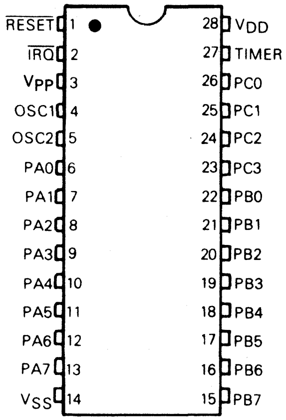
USER PROGRAMMABLE OPTIONS

- Crystal or Low-Cost Resistor Oscillator Option
- Oscillator Internally Divided by Two or Four
- Interrupts Edge Sensitive Only or Level and Edge Sensitive



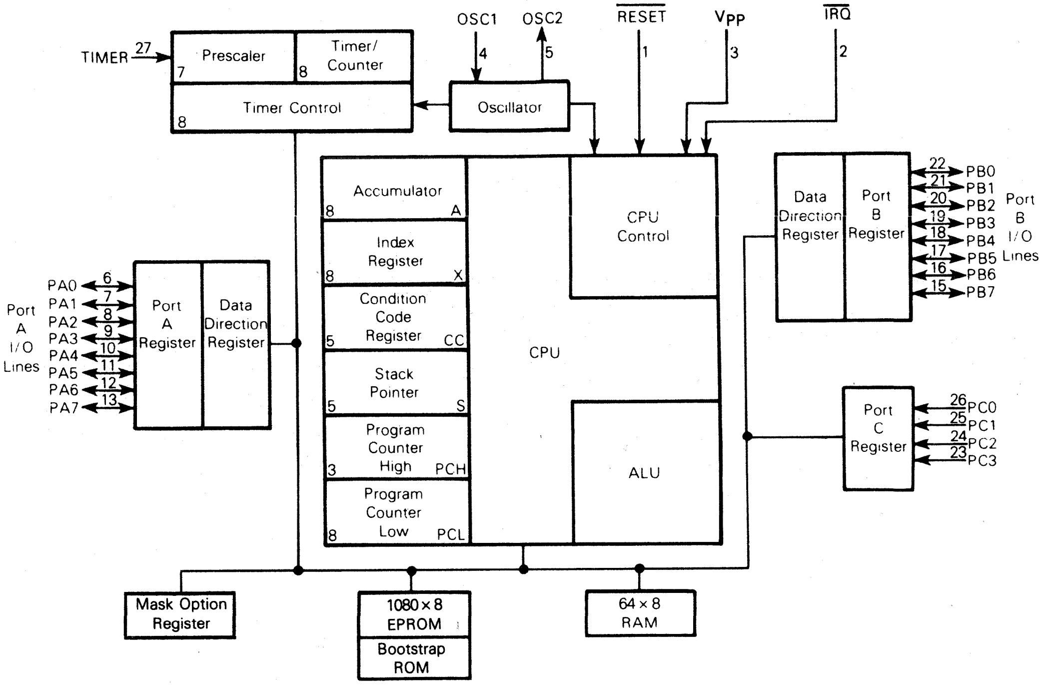
L SUFFIX
CERAMIC PACKAGE
CASE 719

PIN ASSIGNMENTS

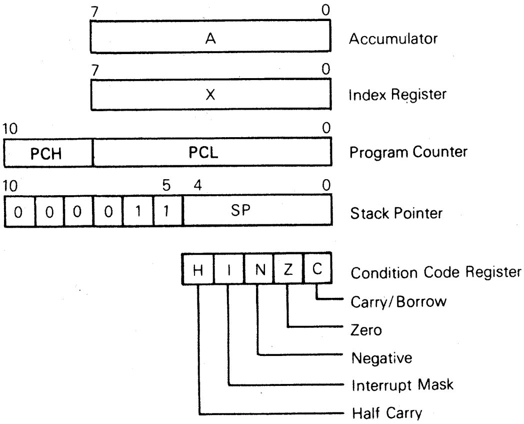


NP-366

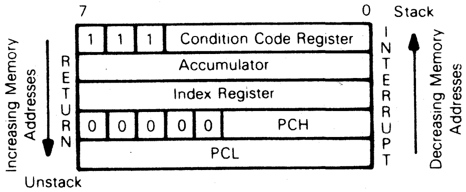
BLOCK DIAGRAM



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

8-BIT EPROM MICROCOMPUTER UNIT

The MC1468705G2 Microcomputer Unit (MCU) is an EPROM member of the M146805 Family of microcomputers, and is the CMOS EPROM MCU version of the ROM based MC146805G2. 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, Bootstrap 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 MC1468705G2 MCU.

HARDWARE FEATURES:

- CMOS EPROM Microcomputer
- Power Saving STOP and WAIT Modes
- 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 Volt Supply
- On-Chip Oscillator with RC or Crystal Options selected by Mask Option Register
- Plug-In Compatible with the MC146805G2
- Bootstrap Program in ROM Simplifies EPROM Programming

SOFTWARE FEATURES

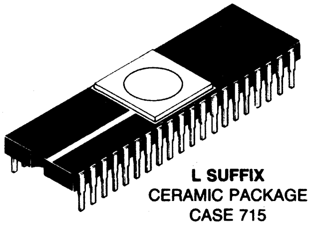
- Similar to the MC6800
- Efficient Use of Program Space
- Versatile Interrupt Handling
- True Bit Manipulation
- Addressing Modes with Indexed Addressing for Tables
- Efficient Instruction Set
- Memory Mapped I/O
- Two Power Saving Standby Modes

ORDERING INFORMATION (T_A = 0°C to 50°C)

Package Type	Frequency	Order Number
Ceramic L Suffix	4.0 MHz	MC1468705G2L

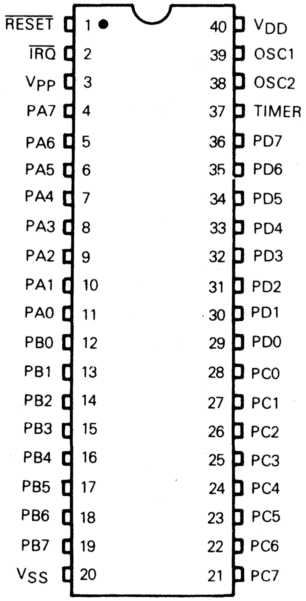
CMOS
(HIGH-PERFORMANCE SILICON-GATE)

8-BIT EPROM
MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENTS



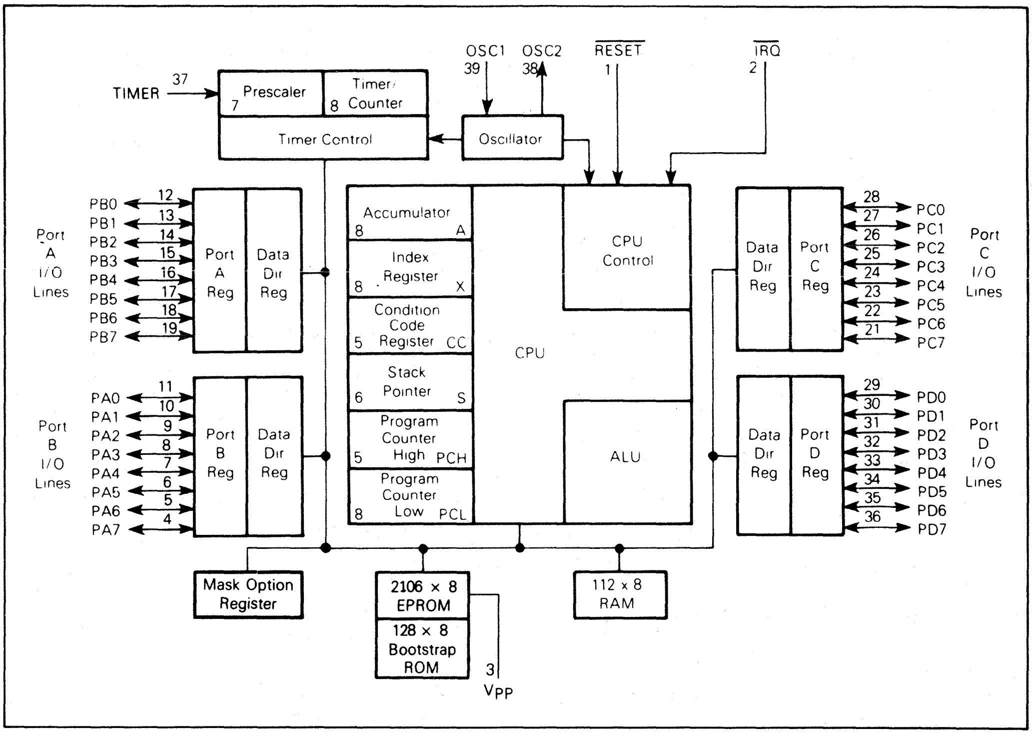


FIGURE 1 — BLOCK DIAGRAM

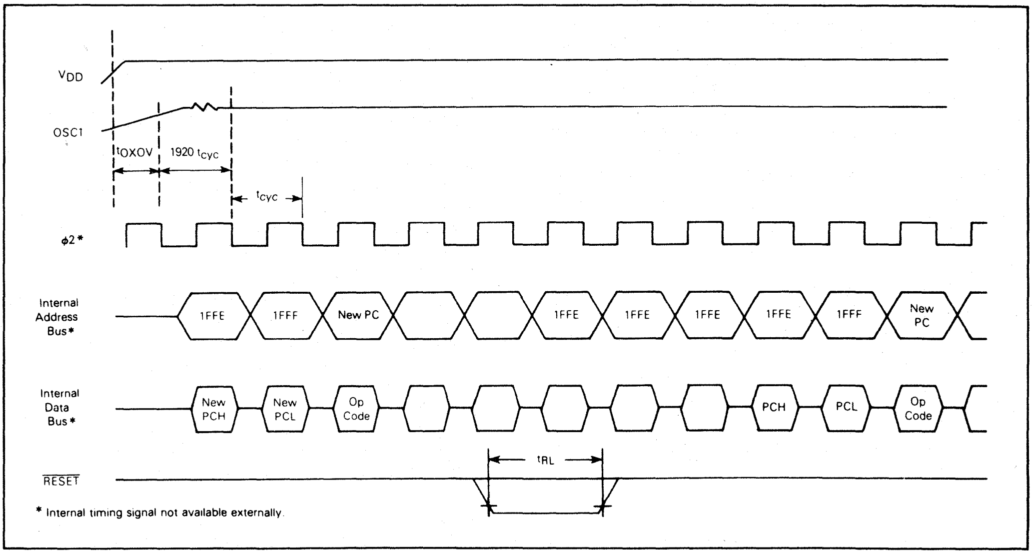
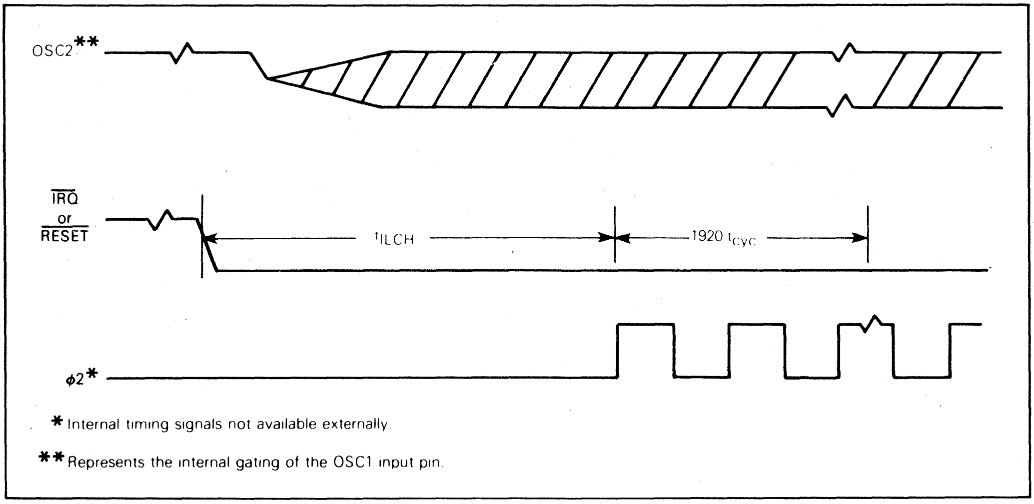


FIGURE 2 — POWER-ON RESET AND $\overline{\text{RESET}}$



* Internal timing signals not available externally

** Represents the internal gating of the OSC1 input pin.

FIGURE 3 — STOP RECOVERY AND POWER-ON RESET

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.

$\overline{\text{IRQ}}$ (Maskable Interrupt Request)

$\overline{\text{IRQ}}$ is programmable option 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 $\overline{\text{IRQ}}$ 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 option is selected to include level sensitivity, then the $\overline{\text{IRQ}}$ input requires an external resistor to V_{DD} for "wire-OR" operation. See INTERRUPTS for more detail. This pin also detects a lower voltage that is used to initiate the bootstrap program.

RESET

The RESET input is not required for startup but can be used to reset the MCU's 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.

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. In normal operation, this pin is connected to V_{SS}. Refer to PROGRAMMING FIRMWARE.

OSC1, OSC2

The MC1468705G2 can be configured to accept either a crystal input or an RC network. Additionally, the internal clocks can be derived by either a divide-by-two or divide-by-four of the external frequency (f_{OSC}). Both of these options are programmable via the mask option register (MOR) in the EPROM array.

RC — If the RC oscillator option is selected, then a resistor is connected to the oscillator pins as shown in Figure 4(b).

CRYSTAL — The circuit shown in Figure 4(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 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 startup stabilization time. Crystal frequency limits are also affected by V_{DD}.

EXTERNAL CLOCK — An external clock should be applied to the OSC1 input with the OSC2 input not connected, as shown in Figure 4(c). An external clock may be used with either the RC or crystal oscillator option.

PA0–PA7

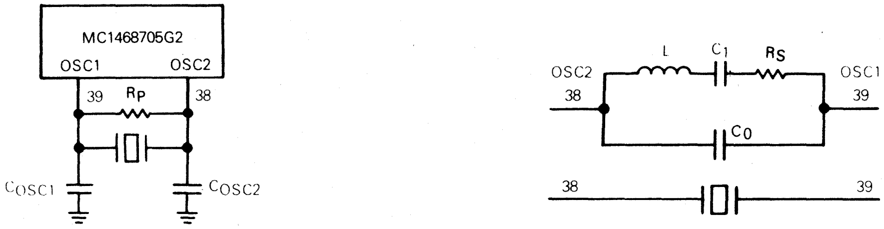
These eight I/O lines comprise Port A. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a detailed description.

PB0–PB7

These eight lines comprise Port B. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a detailed description.

	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

Crystal Parameters



Crystal Oscillator Connections

Equivalent Crystal Circuit

(a)

(b) RC Oscillator Connection

(c) External Clock Source Connections

FIGURE 4 — OSCILLATOR CONNECTIONS

PC0–PC7

These eight lines comprise Port C. The state of any pin is software programmable. Refer to INPUT/OUTPUT PROGRAMMING for a detailed description.

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 detailed description.

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 5 and Table 1.

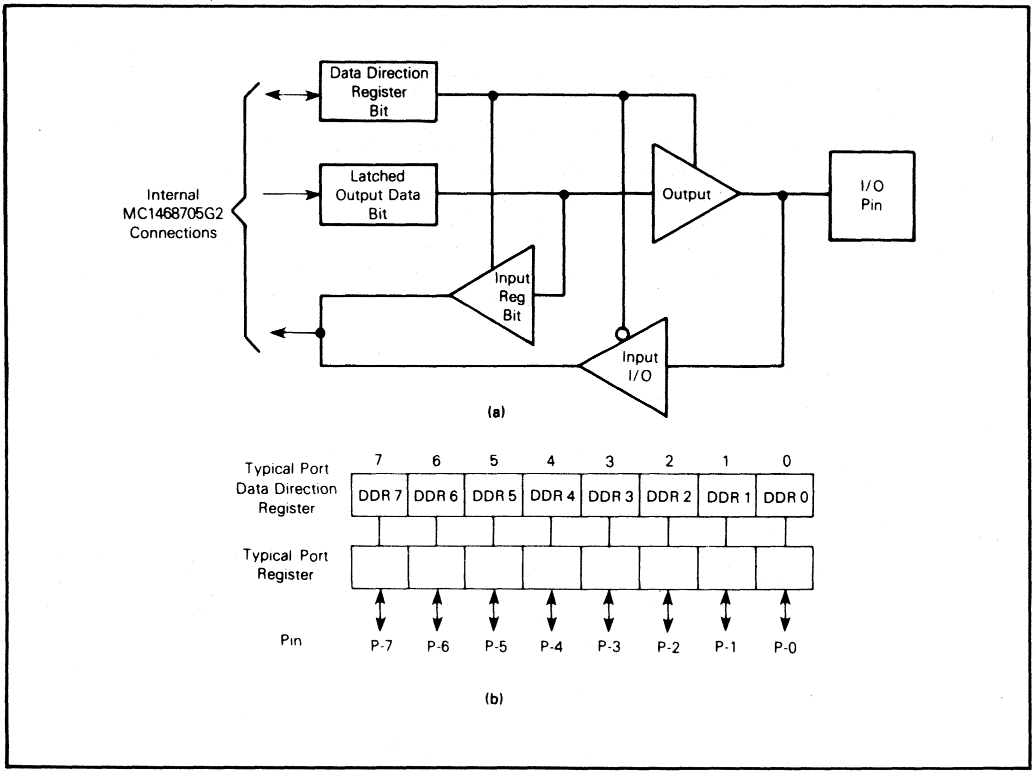


FIGURE 5 — TYPICAL PORT I/O CIRCUITRY

TABLE 1 — I/O PIN FUNCTIONS

R/W̄	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.

EPROM PROGRAMMING

When programming the EPROM array within the MC1468705G2, parts are used in a special arrangement. See PROGRAMMING FIRMWARE and Figure 16 for a detailed description.

MEMORY

As shown in Figure 6, the MCU is capable of addressing 8192 bytes of memory and I/O registers with its program counter. The MC1468705G2 MCU has implemented 2357 bytes of these locations. This consists of: 2106 bytes of user EPROM, 128 bytes of bootstrap ROM, 112 bytes of user RAM, 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 memory locations $0080 to $08AF. The second area is reserved for ten interrupt/reset vector bytes at memory locations $1FF6 to $1FFF. The MCU uses 10 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 $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 identical to the MC146805G2.

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 8. 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.

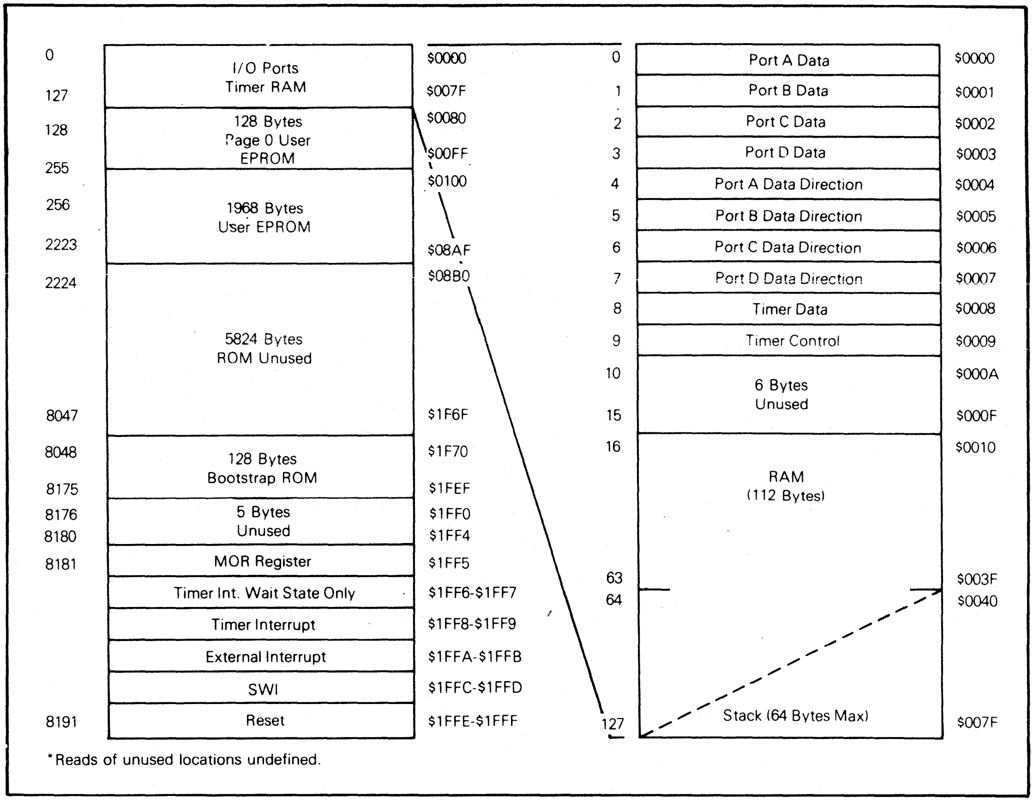


FIGURE 6 — ADDRESS MAP

REGISTERS

The MC1468705G2 contains five registers, as shown in the programming model in Figure 7. The interrupt stacking order is shown in Figure 8.

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 set 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" 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 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.

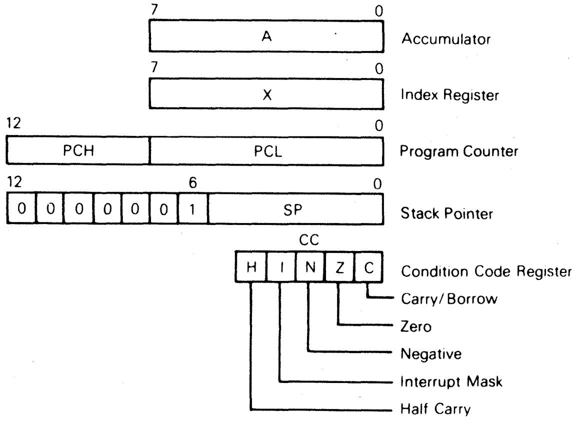
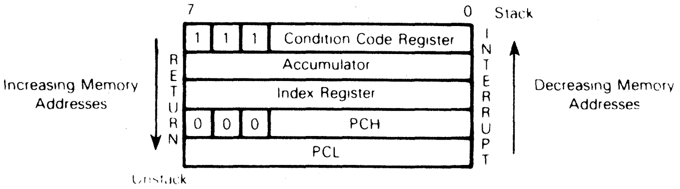


FIGURE 7 — PROGRAMMING MODEL



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 8 — STACKING ORDER

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 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 logic one).

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 MC1468705G2 has two reset modes: an active low external reset pin ($\overline{\text{RESET}}$) and a power-on reset function.

$\overline{\text{RESET}}$

The $\overline{\text{RESET}}$ input pin is ued 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 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 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 zero.
- Timer control register interrupt mask bit TCR6 is set to a one.
- All data direction register bits are cleared to zeros. All ports are defined as inputs.
- Stack pointer is set 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 one.
- STOP and WAIT latches are reset.
- External interupt latch is reset.

All other functions, such as other registers (including I/O ports), the timer, etc. are not cleared by the reset conditions.

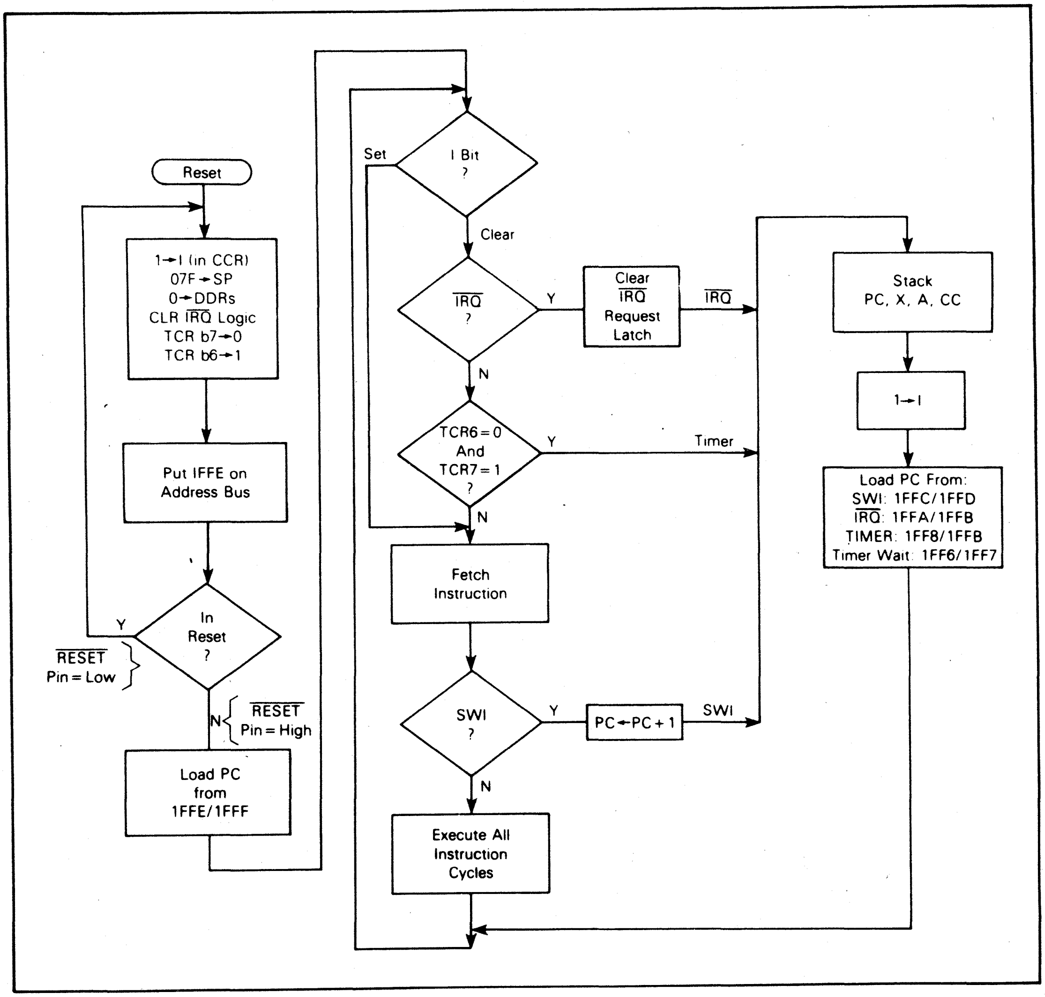


FIGURE 9 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

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 MC1468705G2 EPROM. The bootstrap program provides: 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-triggered/edge and level-triggered interrupt recognition.

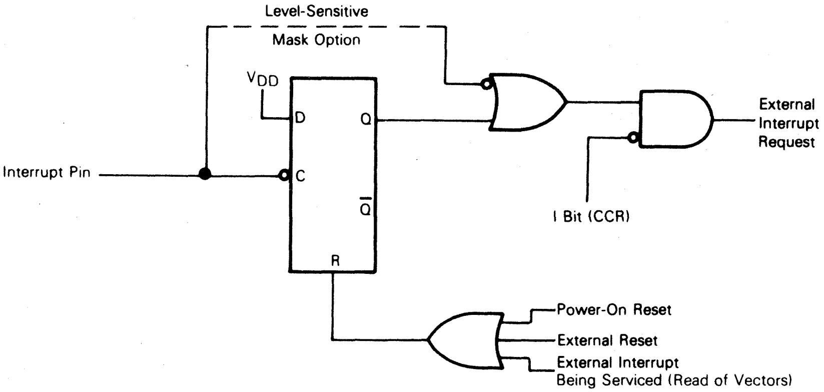
INTERRUPTS

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). Systems often require that normal processing be interrupted so that some external event may be serviced.

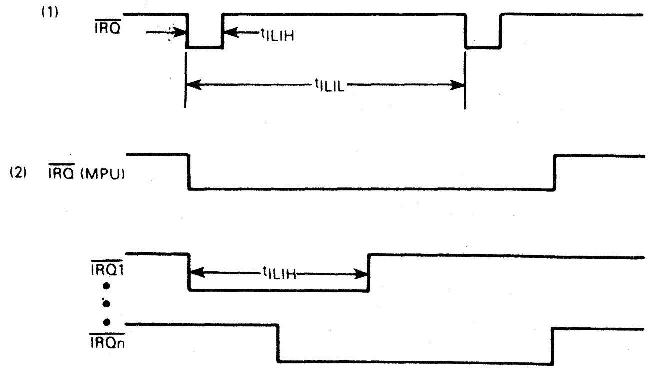
Interrupts cause the processor registers to be saved on the stack and the interrupt mask set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and a return to normal processing. The stack order is shown in Figure 8.

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.

(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 $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

FIGURE 10 — EXTERNAL INTERRUPT

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 9 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 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 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 service routine address is specified by the contents of $1FFA and $1FFB. Either a level- and edge-sensitive (or edge-sensitive only) are available as mask option register (MOR) controlled programmable options. Figure 10 shows both a functional diagram and timing for the interrupt line. The timing diagram shows two different treatments of the interrupt line ($\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 10. The second configuration shows many interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing an interrupt the interrupt line remains low, then the next interrupt is recognized.

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 in the condition code register. The service routine address is specified by the contents of memory locations $1FFC and $1FFD. See Figure 9 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 function the internal oscillator is turned off, causing all internal processing and the timer to be halted; refer to Figure 11.

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 input/output lines remain unchanged.

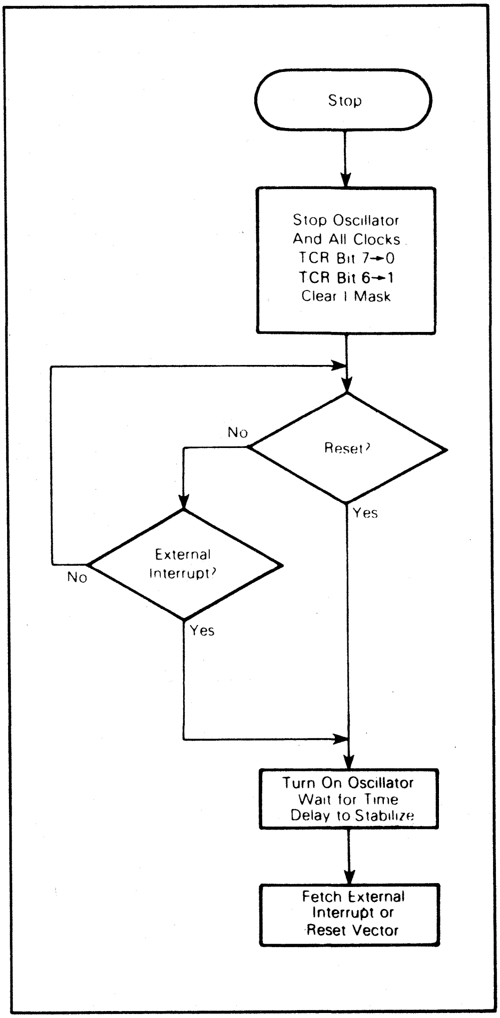


FIGURE 11 — STOP FUNCTION FLOWCHART

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 the timer circuit; refer to Figure 12. 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 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 in the timer wait interrupt) is serviced since the MCU is no longer in the WAIT mode.

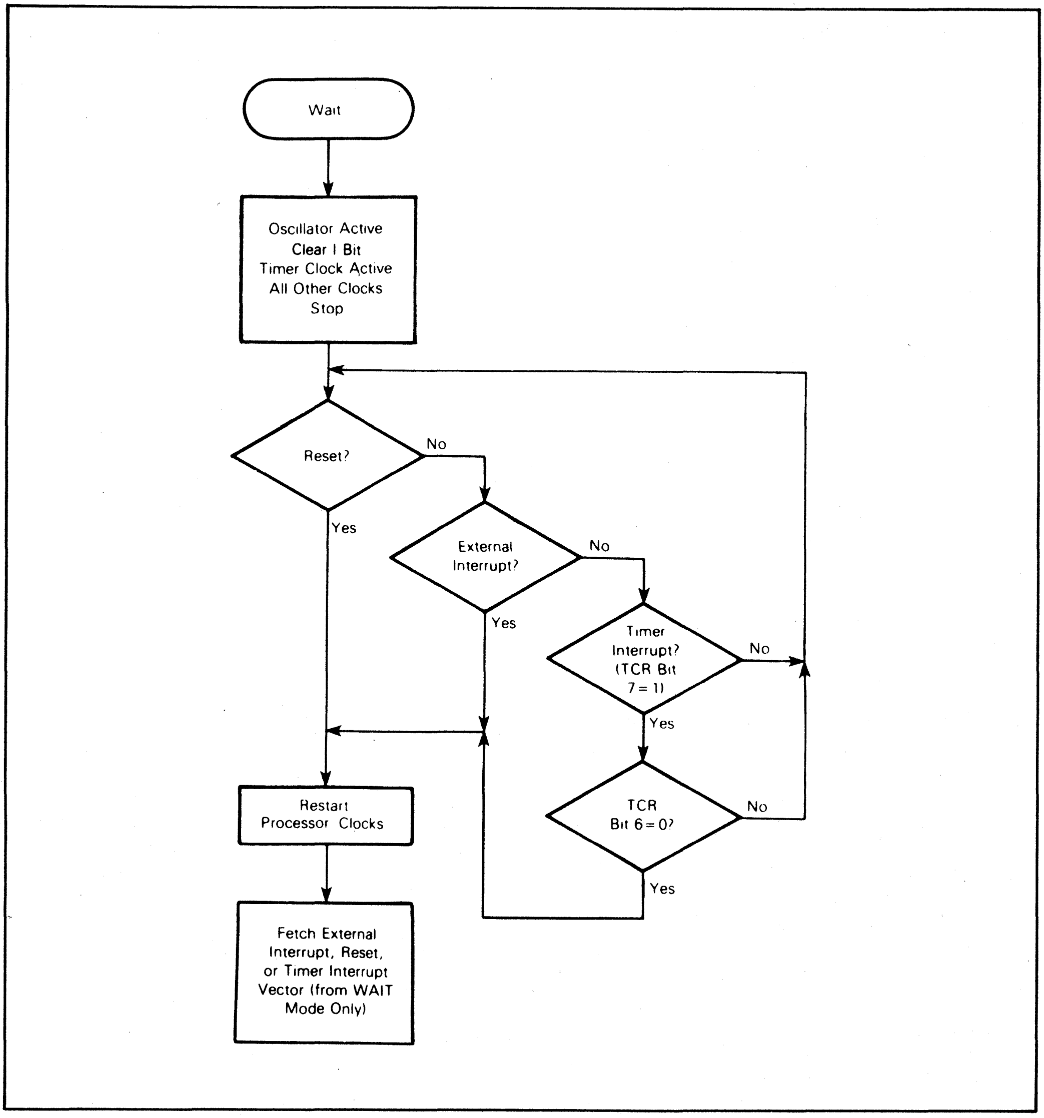


FIGURE 12 — WAIT FUNCTION FLOWCHART

MODES OF OPERATION

The MC1468705G2 has two modes of operation. These modes are the normal (single-chip) mode and the bootstrap mode (the firmware 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. If the reset line is brought low, and the $\overline{\text{IRQ}}$ pin is within its normal operational range (V_{SS}-V_{DD}), the rising edge of the $\overline{\text{RESET}}$ pin will cause 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. V_{PP} volts 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 the TIMER pin should be in the high state and is latched internally on the rising edge of the external $\overline{\text{RESET}}$ pin. Refer to Figure 13.

TIMER

The MCU timer contains an 8-bit software programmable counter (timer data register) with a 7-bit software selectable prescaler. Figure 14 contains a block diagram of the timer.

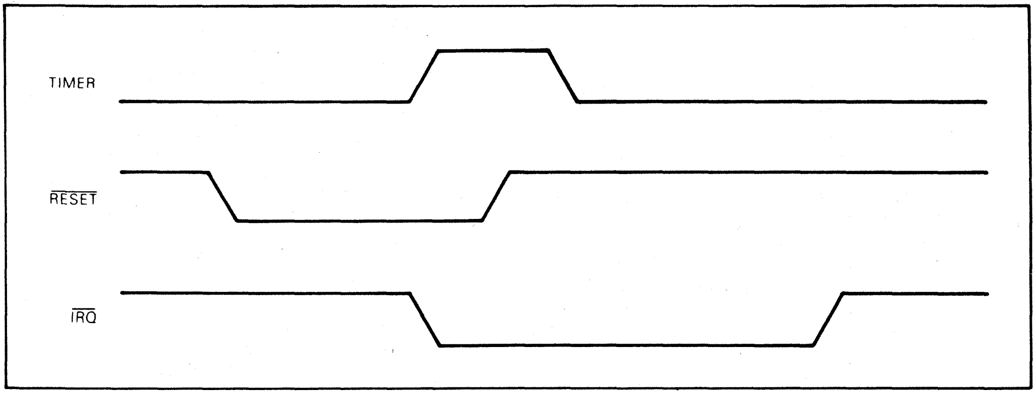
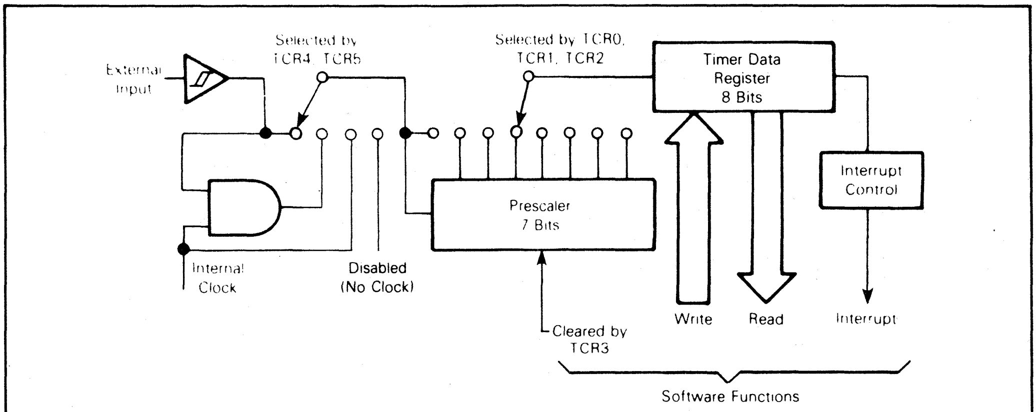


FIGURE 13 — BOOTSTRAP MODE



NOTES:

1. Prescaler and timer data register are clocked on the falling edge of the internal clock or external input.
2. The timer data register counts down continuously.

FIGURE 14 — TIMER BLOCK DIAGRAM

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.

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 the bit clear 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 zeros 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, TCR5 control bits. Refer TIMER CONTROL REGISTER.

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 plus or minus one clock cycle and, therefore, accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

If TCR 4 = 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 14 shows a block diagram of the timer subsystem. Power-on reset and the STOP instruction affect the state of the counter.

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

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

TCR2, TCR1, TCR0 — Prescaler select 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

MASK OPTIONS

The MC1468705G2 "mask options" are implemented as an EPROM byte at address $1FF5 and are EPROM programmable.

MASK OPTION REGISTER (MOR Register)

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.

Bit 7

0 Crystal
1 RC

Bit 6

0 Divide by 4 Oscillator Clock
1 Divide by 2 Oscillator Clock

Bit 4

0 Pulse Interrupts
1 Pulse and Level Interrupts

The EPROM in the erased state will read all zeros. When V_{PP} is at the program voltage the MC1468705G2 will operate under crystal, divide by 4, and pulse only interrupts options, irrespective of the contents of the MOR register. When the V_{PP} voltage is removed the microcomputer will return to the options as specified by the contents of the MOR register.

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 except when erasing. This protects both the EPROM and light-sensitive modes.

PROGRAMMING FIRMWARE

A bootstrap program in ROM allows the MC1468705G2 to program the EPROM. The alternate vectoring used to implement the self-check in the MC146805G2 is used to start execution of this program.

When the V_{PP} voltage is placed on pin 3 and the bootstrap program is called, the MC1468705G2 is able to program itself. This ability to program itself is a function of the external hardware and the interplay between the internal hardware and the firmware. When the part is placed in the bootstrap mode, the bootstrap vector will be fetched and the bootstrap firmware will start to execute.

Note that the MCM2716 UV EPROM(s) must be programmed first with the exact duplicate of the information that is to be transferred to the MC1468705G2. Since the MC1468705G2 and the MCM2716 EPROM(s) 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 15 and 16).

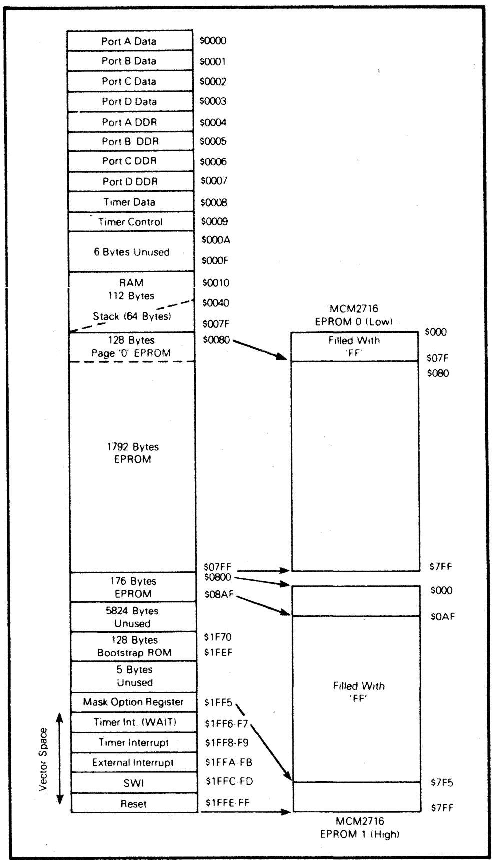
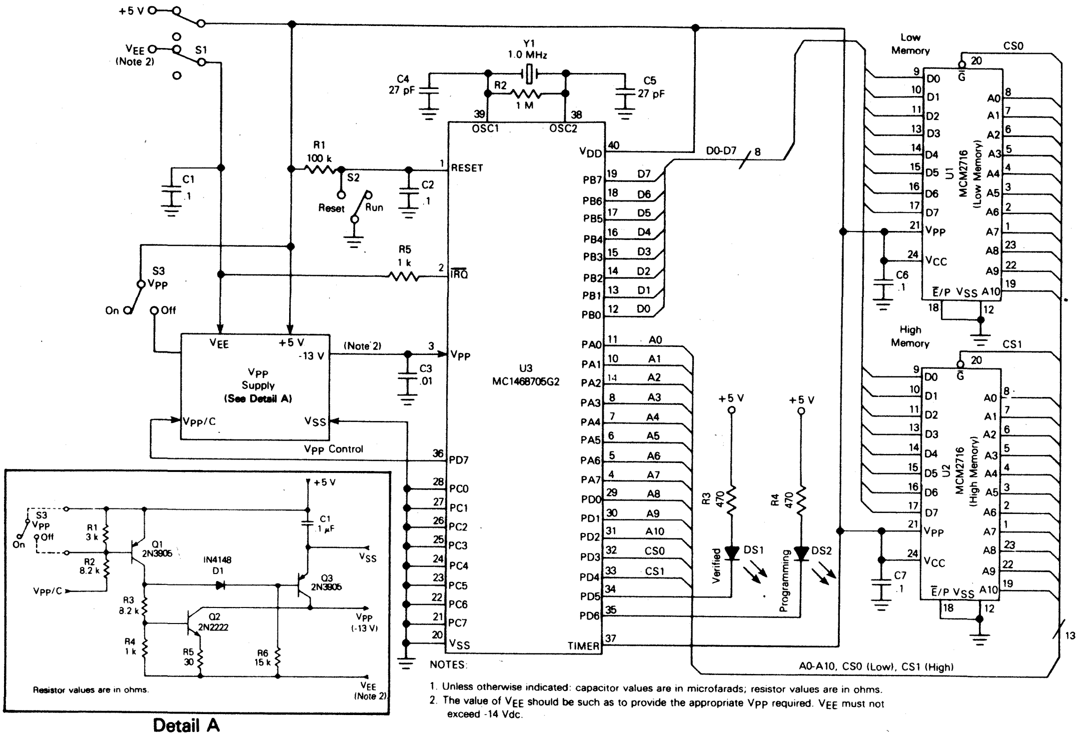


FIGURE 15 — MC1468705G2
MEMORY MAPPING ONTO MCM2716

CAUTION

Be sure that S1 is open and S2 is clocked when inserting the MC1468705G2 and MCM2716 EPROM(s) into their respective sockets. This ensures that RESET is held low while inserting the devices.

To program the MC1468705G2, close S1 (to apply V_{CC}, V_{EE}, and to provide a positive voltage to the timer pin and a negative voltage to the $\overline{IRQ}$ pin). Next, open S2 (to remove $\overline{RESET}$). When released from Reset, the EPROM is programmed one byte at a time with the corresponding data in the MCM2716 EPROMs. The MC1468705G2 bootstrap provides the address and chip select (CS0/CS1) signals to permit complete self-programming at which time the program LED is turned on. Close S2 and open S1 prior to removing the devices from their sockets.



NOTES:

1. Unless otherwise indicated: capacitor values are in microfarads; resistor values are in ohms.
2. The value of V_{EE} should be such as to provide the appropriate Vpp required. V_{EE} must not exceed -14 Vdc.

FIGURE 16 — Programming Connections

Upon completion of the program, the routine enters the verify mode. If every programmed byte corresponds to the data in the MCM2716s, the verify light is turned on and stays on until RESET.

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 of 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 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. 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 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 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 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 5.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 6.

OPCODE MAP

Table 7 is an opcode map for the instructions used on the MCU.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 8.

TABLE 2 — 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 3 — 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

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. 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 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 7.

The term "effective address" (EA) is used in describing the various addressing modes, which 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 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 reg-

TABLE 4 — 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 5 — 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	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	–	–	–

isters 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)

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.

$$EA = PC + 2 + (PC + 1);\ PC \leftarrow EA \text{ if branch taken;}$$
$$\text{otherwise, } PC \leftarrow PC + 2$$

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 to 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.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

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 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.

$$EA1 = (PC + 1)$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$
$$EA2 = PC + 3 + (PC + 2);\ PC \leftarrow EA2 \text{ if branch taken;}$$
$$\text{otherwise, } PC \leftarrow PC + 3$$

TABLE 6 — 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 7 — MC1468705G2 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	
Low \ Hi	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	REL	Relative
A	Accumulator	BSC	Bit Set/Clear
X	Index Register	BTB	Bit Test and Branch
IMM	Immediate	IX	Indexed (No Offset)
DIR	Direct	IX1	Indexed, 1 Byte (8-Bit) Offset
EXT	Extended	IX2	Indexed, 2 Byte (16-Bit) Offset

LEGEND

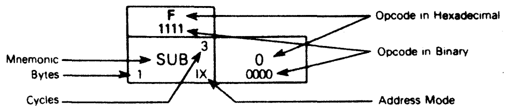


TABLE 8 — INSTRUCTION SET

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

MOTOROLA

MC68HC04P2

Product Preview

8-BIT HCMOS MICROCOMPUTER UNIT

The MC68HC04P2 HCMOS Microcomputer Unit (MCU) is a member of the M68HC04 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. The following are some of the hardware and software highlights of the MC68HC04P2 MCU.

HARDWARE FEATURES

- Low Power HCMOS
- Power Saving Stop and Wait Modes
- 8-Bit Architecture
- Pin Compatible with the MC6805P2, MC68705P3, and MC6804P2
- 32 Bytes of RAM
- Memory Mapped I/O
- 1024 Bytes of User ROM
- 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

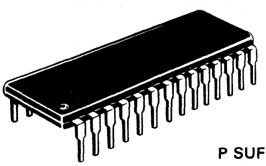
USER SELECTABLE OPTIONS

- 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

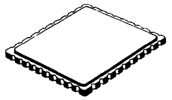
EXORciser is a registered trademark of Motorola Inc.

HCMOS
(HIGH-DENSITY CMOS
SILICON-GATE)

8-BIT HCMOS
MICROCOMPUTER

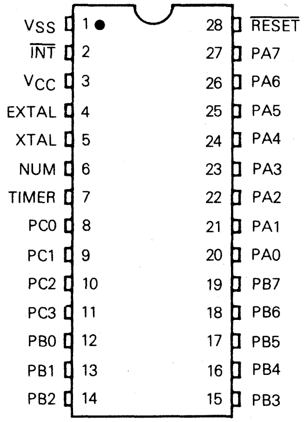


P SUFFIX
PLASTIC PACKAGE
CASE 710



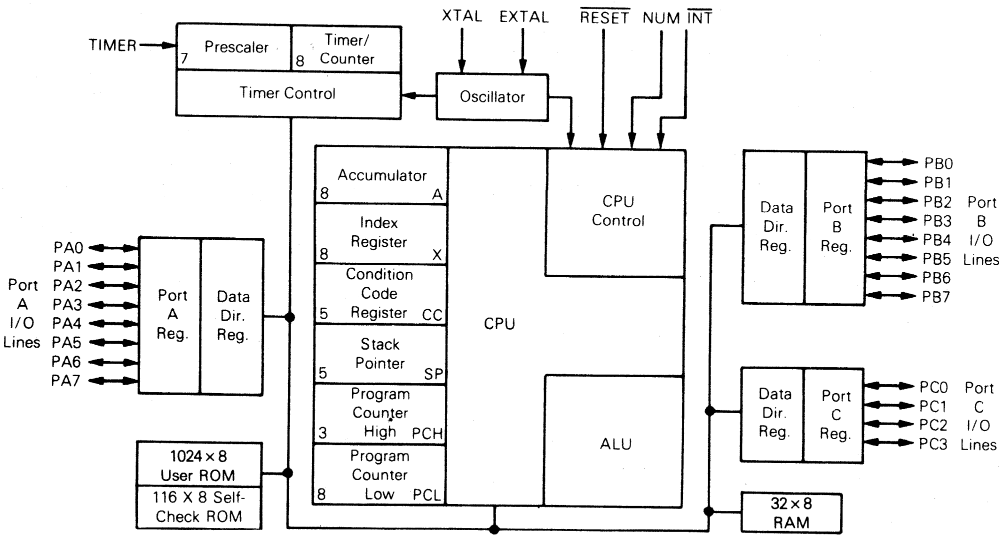
Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS

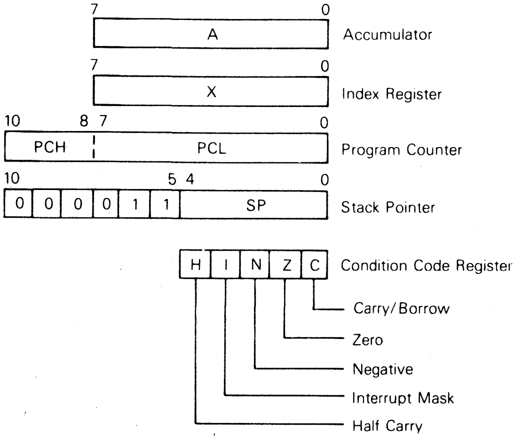


NP-368

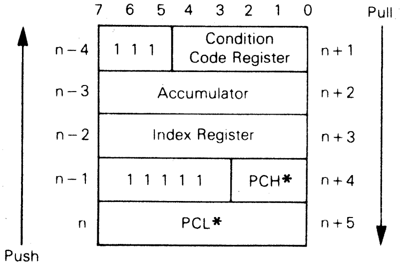
BLOCK DIAGRAM



PROGRAMMING MODEL



INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH are stacked.

MOTOROLA

MC68HC04P3

PRODUCT PREVIEW

8-BIT MICROCOMPUTER

THE MC68HC04.. MICROCOMPUTER UNIT (MCU) IS A LOW COST SINGLE CHIP MICROCOMPUTER IMPLEMENTED IN HIGH DENSITY SILICON GATE HCMOS. IT CONTAINS ON-CHIP CPU, ROM, RAM, TIMER AND PROGRAMMABLE I/O. IT IS DESIGNED FOR LOW COST, LOW POWER CONSUMPTION APPLICATIONS AND RETAINS THE PROVEN CAPABILITIES OF THE M6805 BASED INSTRUCTION SET.

FEATURES

- 128 BYTES OF ON-CHIP RAM.
- 2K BYTES OF ON-CHIP ROM.
- STANDBY MODE RETAINS RAM CONTENTS.
- 20 PROGRAMMABLE I/O LINES, MEMORY MAPPED.
- 8 BIT TIMER WITH 8 BIT PRESCALER + TIMER I/O.
- MASKABLE TIMER INTERRUPT.
- SELF-CHECK MODE.
- 4 LEVEL HARDWARE STACK.
- INSTRUCITON SET SIMILAR TO (14)6805 FAMILY INCLUDING STOP, WAIT, AND TRUE BIT MANIPULATION.
- 28 PIN PACKAGE.

HCMOS

(HIGH PERFORMANCE
SILICON GATE)

8-BIT

SINGLE CHIP MICROCOMPUTER

P SUFFIX

28 PIN PLASTIC

DIL PACKAGE

PIN ASSIGNMENTS

Vss	1	28	RESET
INT	2	27	PA7
Vcc	3	26	PA6
EXTAL	4	25	PA5
XTAL	5	24	PA4
MDS	6	23	PA3
TIMER	7	22	PA2
PC0	8	21	PA1
PC1	9	20	PA0
PC2	10	19	PB6
PC3	11	18	PB5
PB0	12	17	PB5
PB1	13	16	PB4
PB2	14	15	PB3

MOTOROLA

MC68HC05C4

HCMOS

(HIGH DENSITY CMOS
SILICON-GATE)

**8-BIT HCMOS
MICROCOMPUTER**

Product Preview

8-BIT HCMOS MICROCOMPUTER UNIT

The MC68HC05C4 HCMOS Microcomputer Unit (MCU) is a member of the M68HC05 Family of low-cost single-chip microcomputers. This 8-bit microcomputer 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.

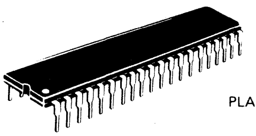
The following are some of the hardware and software highlights of the MC68HC05C4 MCU.

HARDWARE FEATURES

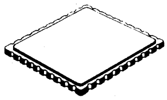
- HCMOS Technology
- 8-Bit Architecture
- Power Saving Stop and Wait Modes
- Fully Static Operation
- 176 Bytes of On-Chip RAM
- 4144 Bytes of On-Chip ROM
- 24 Bidirectional I/O Lines
- 2.1 MHz Internal Operating Frequency at 5 Volts
- Internal 16-Bit Timer Similar to the MC6801 Timer
- Serial Communication Interface System
- Serial Peripheral Interface System
- Self-Check Mode
- External Interrupt, Timer, Serial Communication Interface, and Serial Peripheral Interface Interrupts
- Master Reset and Power-On Resets
- 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
- 8 × 8 Multiply Instruction
- 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
- Upwards Software Compatible with the M146805 Family

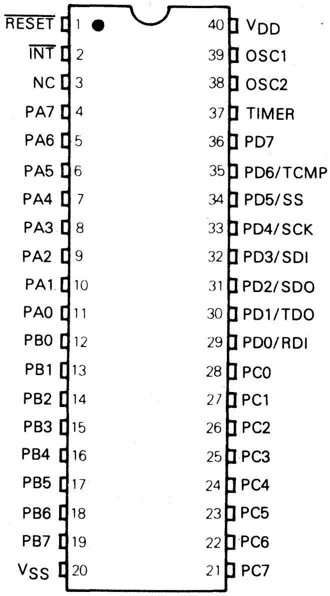


P SUFFIX
PLASTIC PACKAGE
CASE 711

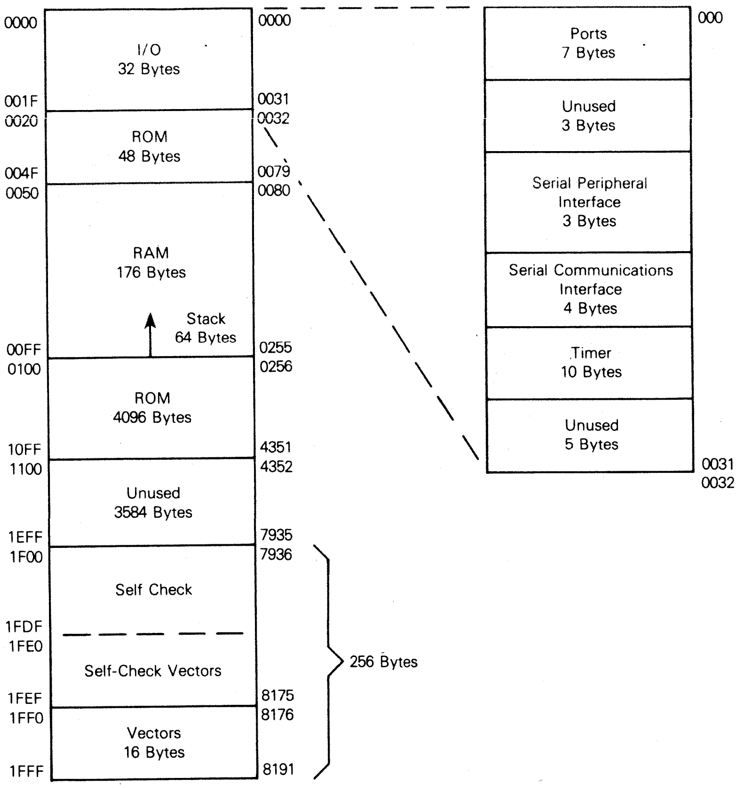


Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENTS



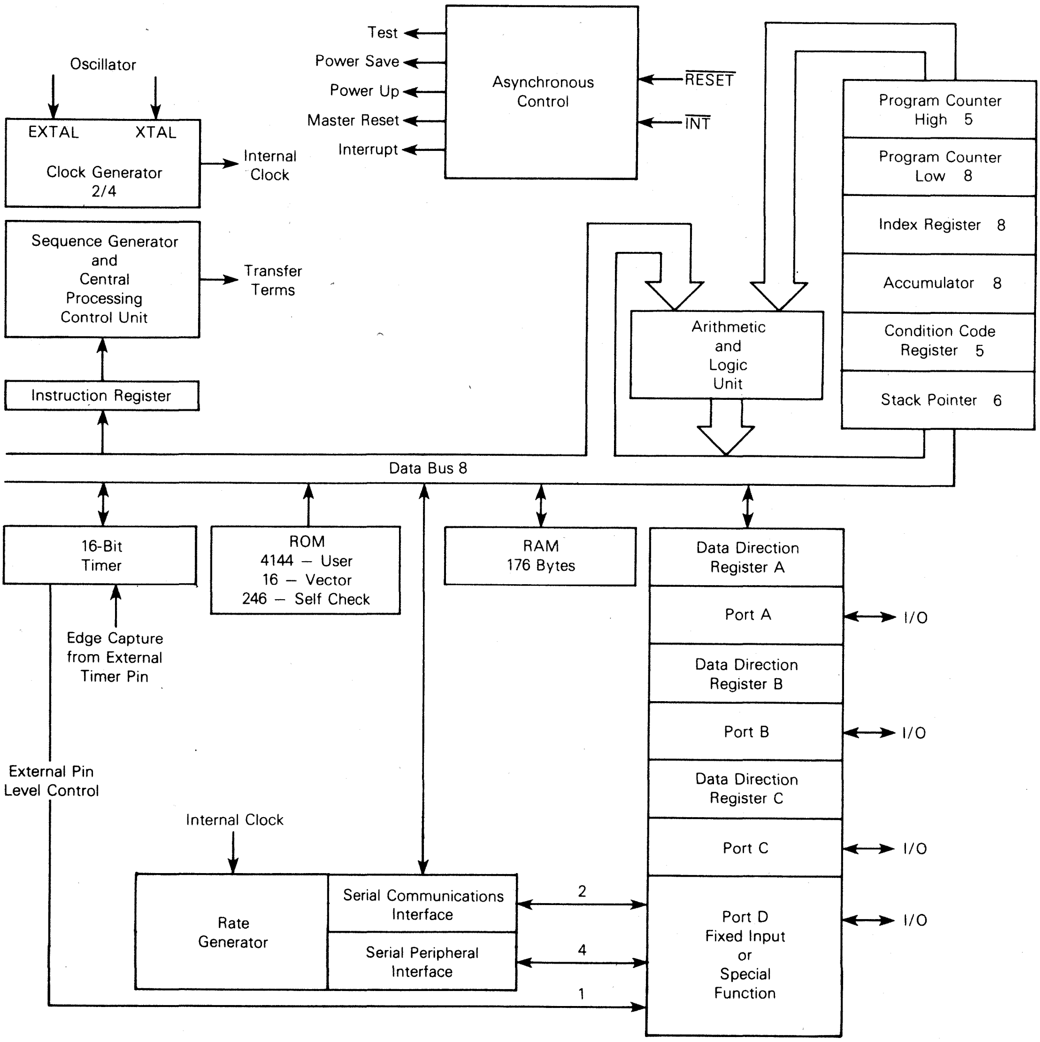
MEMORY MAP



NOTES:

1. Self check vector offset = (– 16). Address bit 4 forced to = 0 in self-check mode, when serving reset or interrupts.
2. ROM area added or deleted in 256 byte segments.

BLOCK DIAGRAM



MOTOROLA

MC68HC09E

Product Preview

8-BIT HCMOS MICROPROCESSING UNIT

The MC68HC09E is a revolutionary low-power high-performance 8-bit HCMOS microprocessor which supports modern programming techniques such as position independence, reentrancy, and modular programming.

This third-generation addition to the M6800 Family has major architectural improvements which include additional registers, instructions, and addressing modes.

The basic instructions of any computer are greatly enhanced by the presence of powerful addressing modes. The MC68HC09E has the most complete set of addressing modes available on any 8-bit microprocessor today.

The MC68HC09E has hardware and software features which make it an ideal processor for higher level language execution or standard controller applications. External clock inputs are provided to allow synchronization with peripherals, systems, or other MPUs.

HARDWARE FEATURES

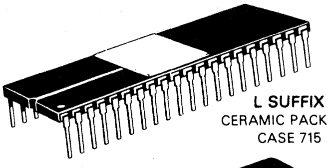
- Very Low-Power High-Density CMOS
- External Clock Inputs, E and Q, Allow Synchronization
- TSC Input Controls Internal Bus Buffers
- LIC Indicates Opcode Fetch
- AVMA Allows Efficient Use of Common Resources in a Multiprocessor System
- BUSY is a Status Line for Multiprocessing
- Fast Interrupt Request Input Stacks Only Condition Code Register and Program Counter
- Interrupt Acknowledge Output Allows Vectoring by Devices
- Sync Acknowledge Output Allows for Synchronization to External Event
- Single Bus-Cycle $\overline{\text{RESET}}$
- Single 5-Volt Supply Operation
- $\overline{\text{NMI}}$ Inhibited After $\overline{\text{RESET}}$ Until After First Load of Stack Pointer
- Early Address Valid Allows Use with Slower Memories
- Early Write Data for Dynamic Memories

SOFTWARE FEATURES

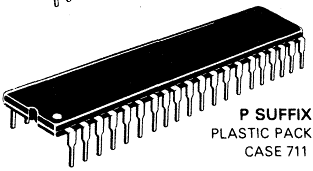
- 10 Addressing Modes
  - M6800 Upward Compatible Addressing Modes
  - Direct Addressing Anywhere in Memory Map
  - Long Relative Branches
  - Program Counter Relative
  - True Indirect Addressing
  - Expanded Indexed Addressing
    - 0-, 5-, 8-, or 16-Bit Constant Offsets
    - 8- or 16-Bit Accumulator Offsets
    - Auto-Increment/Decrement by 1 or 2
- Improved Stack Manipulation
- 1464 Instruction with Unique Addressing Modes
- 8 × 8 Unsigned Multiply
- 16-Bit Arithmetic
- Transfer/Exchange All Registers
- Push/Pull Any Registers or Any Set of Registers
- Load Effective Address

HCMOS
(HIGH-DENSITY CMOS
SILICON-GATE)

**8-BIT HCMOS
MICROPROCESSING
UNIT**

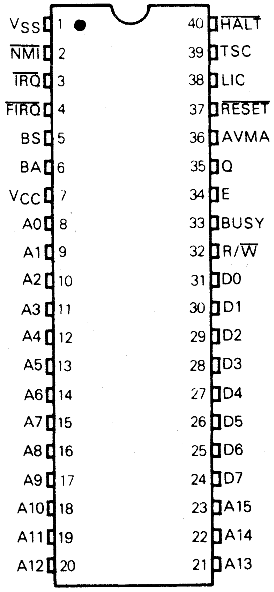


L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENTS



NP-367

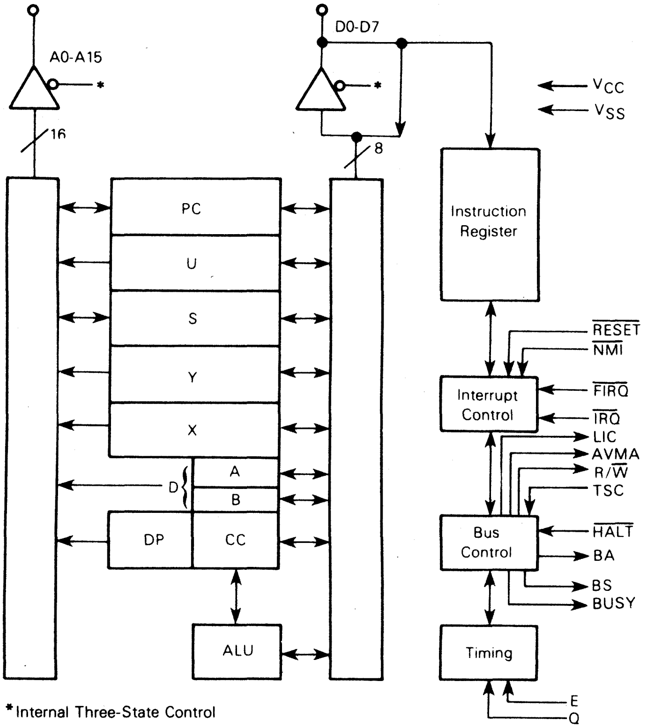
ARCHITECTURAL FEATURES

- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

MC6800 COMPATIBLE

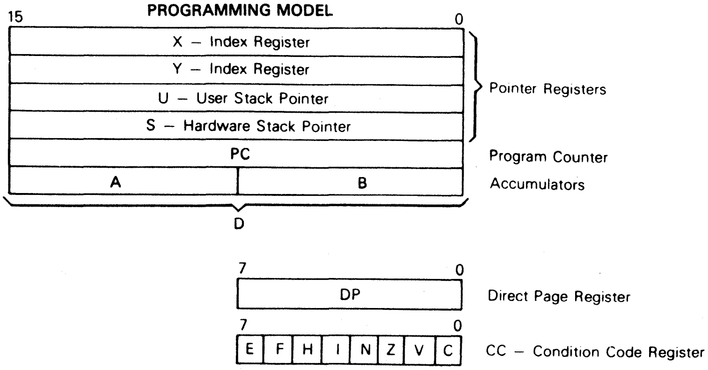
- Hardware — Interfaces with All M68010 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

EXPANDED BLOCK DIAGRAM

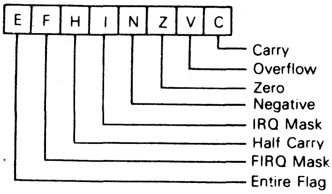


*Internal Three-State Control

PROGRAMMING MODEL



CONDITION CODE REGISTER FORMAT



MOTOROLA

MCM6810
1.0 MHz
MCM68A10
1.5 MHz
MCM68B10
2.0 MHz

MOS
(N-CHANNEL, SILICON-GATE)

128 X 8-BIT STATIC RANDOM ACCESS MEMORY

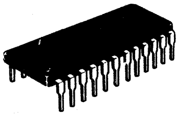
128 X 8-BIT STATIC RANDOM ACCESS MEMORY

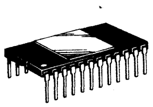
The MCM6810 is a byte-organized memory designed for use in bus-organized systems. It is fabricated with N-channel silicon-gate technology. For ease of use, the device operates from a single power supply, has compatibility with TTL and DTL, and needs no clocks or refreshing because of static operation.

The memory is compatible with the M6800 Microcomputer Family, providing random storage in byte increments. Memory expansion is provided through multiple Chip Select inputs.

- Organized as 128 Bytes of 8 Bits
- Static Operation
- Bidirectional Three-State Data Input/Output
- Six Chip Select Inputs (Four Active Low, Two Active High)
- Single 5-Volt Power Supply
- TTL Compatible
- Maximum Access Time = 450 ns — MCM6810
  360 ns — MCM68A10
  250 ns — MCM68B10

P SUFFIX
PLASTIC PACKAGE
CASE 709



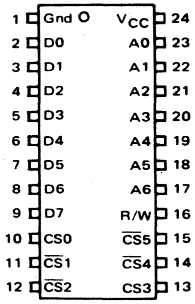


L SUFFIX
CERAMIC PACKAGE
CASE 716

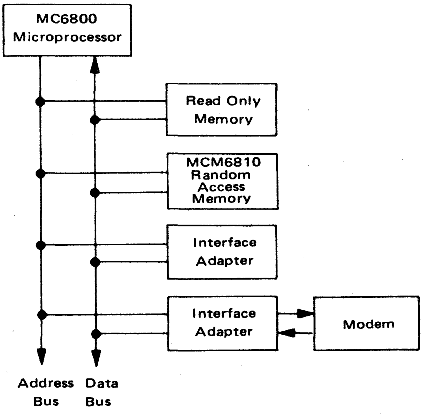
ORDERING INFORMATION

Speed	Device	Temperature Range
1.0 MHz	MC6810P, L	0 to 70°C
	MC6810CP, CL	-40 to +85°C
MIL-STD-883B	MC6810BJCS	-55 to +125°C
MIL-STD-883C	MC6810CJCS	
1.5 MHz	MC68A10P, L	0 to +70°C
	MC68A10CP, CL	-40 to +85°C
2.0 MHz	MC68B10P, L	0 to +70°C

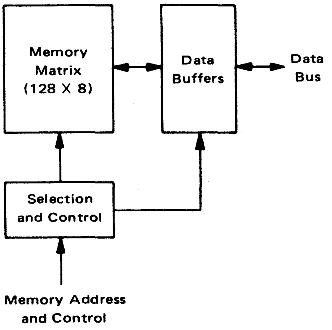
PIN ASSIGNMENT



M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM



MCM6810 — RANDOM ACCESS MEMORY BLOCK DIAGRAM



DS 9487

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 to +7.0	Vdc
Input Voltage	V_{in}	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	T_L to T_H 0 to 70 -40 to 85 -55 to 125	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Thermal Resistance	θ_{JA}	82.5	°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.

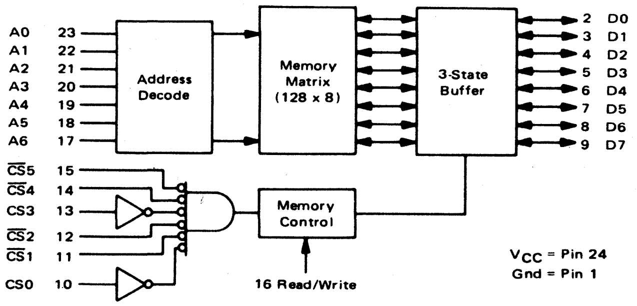
ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input Current (A_n, R/W, CS_n, $\overline{CS}_n$) (V_{in} = 0 to 5.25 V)		I_{in}	–	–	2.5	μAdc
Output High Voltage (I_{OH} = -205 μA)		V_{OH}	2.4	–	–	Vdc
Output Low Voltage (I_{OL} = 1.6 mA)		V_{OL}	–	–	0.4	Vdc
Output Leakage Current (Three-State) (CS = 0.8 V or $\overline{CS}$ = 2.0 V, V_{out} = 0.4 V to 2.4 V)		I_{TSI}	–	–	10	μAdc
Supply Current (V_{CC} = 5.25 V, all other pins grounded)	1.0 MHz 1.5, 2.0 MHz	I_{CC}	–	–	80 100	mAdc
Input Capacitance (A_n, R/$\overline{W}$, CS_n, $\overline{CS}_n$) (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)		C_{in}	–	–	7.5	pF
Output Capacitance (D_n) (V_{out} = 0, T_A = 25°C, f = 1.0 MHz, CS0 = 0)		C_{out}	–	–	12.5	pF

RECOMMENDED DC OPERATING CONDITIONS

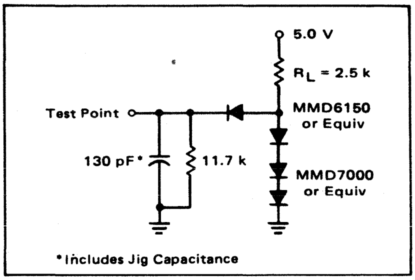
Parameter	Symbol	Min	Nom	Max	Unit
Input High Voltage	V_{IH}	2.0	–	5.25	Vdc
Input Low Voltage	V_{IL}	-0.3	–	0.8	Vdc

BLOCK DIAGRAM



V_{CC} = Pin 24
Gnd = Pin 1

FIGURE 1 – AC TEST LOAD



*Includes Jig Capacitance

AC TEST CONDITIONS

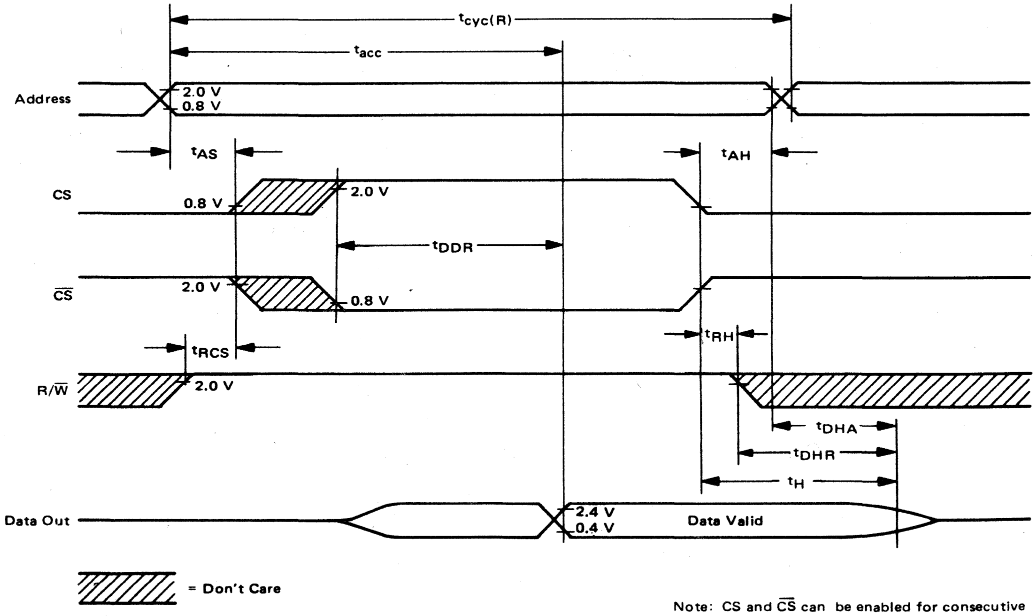
Condition	Value
Input Pulse Levels	0.8 V to 2.0 V
Input Rise and Fall Times	20 ns
Output Load	See Figure 1

AC OPERATING CONDITIONS AND CHARACTERISTICS

READ CYCLE (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic	Symbol	MCM6810		MCM68A10		MCM68B10		Unit
		Min	Max	Min	Max	Min	Max	
Read Cycle Time	$t_{cyc(R)}$	450	–	360	–	250	–	ns
Access Time	t_{acc}	–	450	–	360	–	250	ns
Address Setup Time	t_{AS}	20	–	20	–	20	–	ns
Address Hold Time	t_{AH}	0	–	0	–	0	–	ns
Data Delay Time (Read)	t_{DDR}	–	230	–	220	–	180	ns
Read to Select Delay Time	t_{RCS}	0	–	0	–	0	–	ns
Data Hold from Address	t_{DHA}	10	–	10	–	10	–	ns
Output Hold Time	t_H	10	–	10	–	10	–	ns
Data Hold from Read	t_{DHR}	10	80	10	60	10	60	ns
Read Hold from Chip Select	t_{RH}	0	–	0	–	0	–	ns

READ CYCLE TIMING

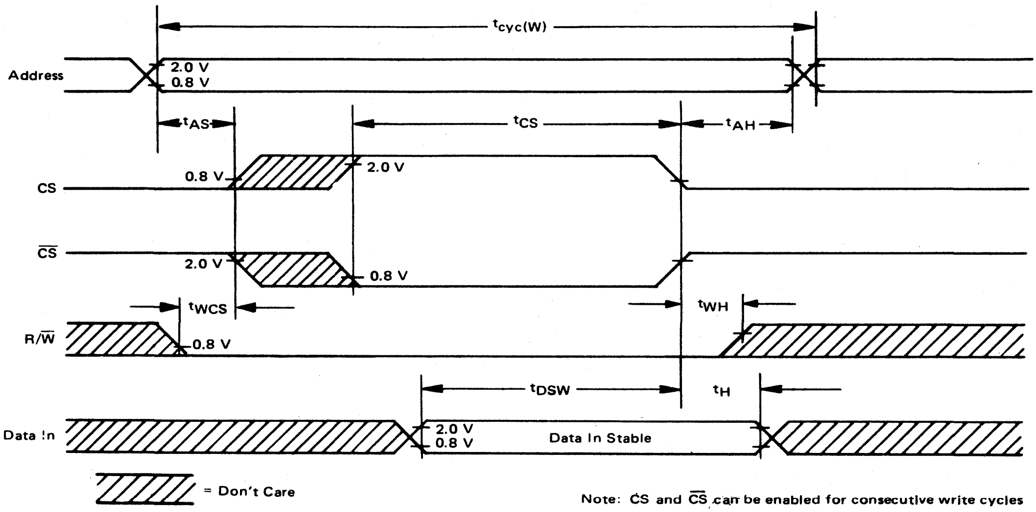


Note: CS and $\overline{CS}$ can be enabled for consecutive read cycles provided R/W remains at V_{IH}.

WRITE CYCLE (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

Characteristic	Symbol	MCM6810		MCM68A10		MCM68B10		Unit
		Min	Max	Min	Max	Min	Max	
Write Cycle Time	$t_{cyc(W)}$	450	–	360	–	250	–	ns
Address Setup Time	t_{AS}	20	–	20	–	20	–	ns
Address Hold Time	t_{AH}	0	–	0	–	0	–	ns
Chip Select Pulse Width	t_{CS}	300	–	250	–	210	–	ns
Write to Chip Select Delay Time	t_{WCS}	0	–	0	–	0	–	ns
Data Setup Time (Write)	t_{DSW}	190	–	80	–	60	–	ns
Input Hold Time	t_H	10	–	10	–	10	–	ns
Write Hold Time from Chip Select	t_{WH}	0	–					

WRITE CYCLE TIMING



Note: CS and $\overline{CS}$ can be enabled for consecutive write cycles provided R/W is strobed to V_{IH} before or coincident with the Address change, and remains high for time t_{AS}.

MOTOROLA

MC2661A/MC68661A
(Baud Rate Set A)
MC2661B/MC68661B
(Baud Rate Set B)
MC2661C/MC68661C
(Baud Rate Set C)

Advance Information

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI)

The MC2661/MC68661, Enhanced Programmable Communications Interface (EPCI), is a universal synchronous/asychronous data communications controller chip that is an enhanced version of the Signetics 2651. The EPCI directly interfaces to most 8-bit MPUs and easily to the MC68000 MPU and other 16-bit MPUs. It may be used in either a polled or interrupt driven system. Programmed instructions can be accepted from the host MPU while supporting many synchronous or asynchronous serial-data communication protocols in a full or half-duplex mode. Special support for BISYNC is provided.

The EPCI converts parallel data characters, accepted from the microprocessor data bus, into transmit-serial data. Simultaneously, the EPCI can convert receive-serial data to parallel data characters for input to the microprocessor.

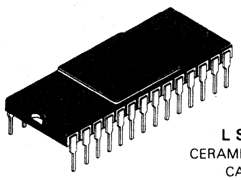
A baud rate generator in the EPCI can be programmed to either accept an external clock, or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (A, B, C) has a different set of baud rates.

FEATURES

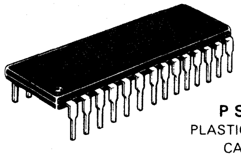
- **Synchronous Operation**
  - Single or Double SYN Operation
  - Internal or External Character Synchronization
  - Transparent or Non-transparent Mode
  - Transparent Mode DLE Stuffing (Tx) and Detection (Rx)
  - Automatic SYN or DLE-SYN Insertion
  - SYN, DLE, and DLE-SYN Stripping
  - Baud Rate: dc to 1M bps (1X Clock)
- **Asynchronous Operation**
  - 1, 1½, or 2 Stop Bits Transmitted
  - Parity, Overrun, and Framing Error Detection
  - Line Break Detection and Generation
  - False Start Bit Detection
  - Automatic Serial Echo Mode (Echoplex)
  - Baud Rate: dc 1M bps (1X Clock)
    dc to 62.5k bps (16X Clock)
    dc to 15.625k bps (64X Clock)
- **Common Features**
  - Internal or External Baud Rate Clock; No System Clock Required
  - 3 Baud Rate Sets (A, B, C); 16 Internal Rates for Each Set
  - 5- to 8-Bit Characters plus parity; Odd, Even, or No Parity
  - Double Buffered Transmitter and Receiver
  - Dynamic Character Length Switching
  - Full- or Half-Duplex Operation
  - Local or Remote Maintenance Loop-Back Mode
  - TTL-Compatible Inputs and Outputs
  - RxC and TxC Pins and Short Circuit Protected
  - 3 Open-Drain MOS Outputs can be Wire ORed
  - Single 5 V Power Supply
- **Applications**
  - Intelligent Terminals
  - Network Processors
  - Front End Processors
  - Remote Data Concentrators
  - Computer-to-Computer Links
  - Serial Peripherals
  - BISYNC Adaptors

MOS
(N-CHANNEL, SILICON-GATE)

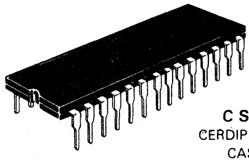
ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE (EPCI)



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



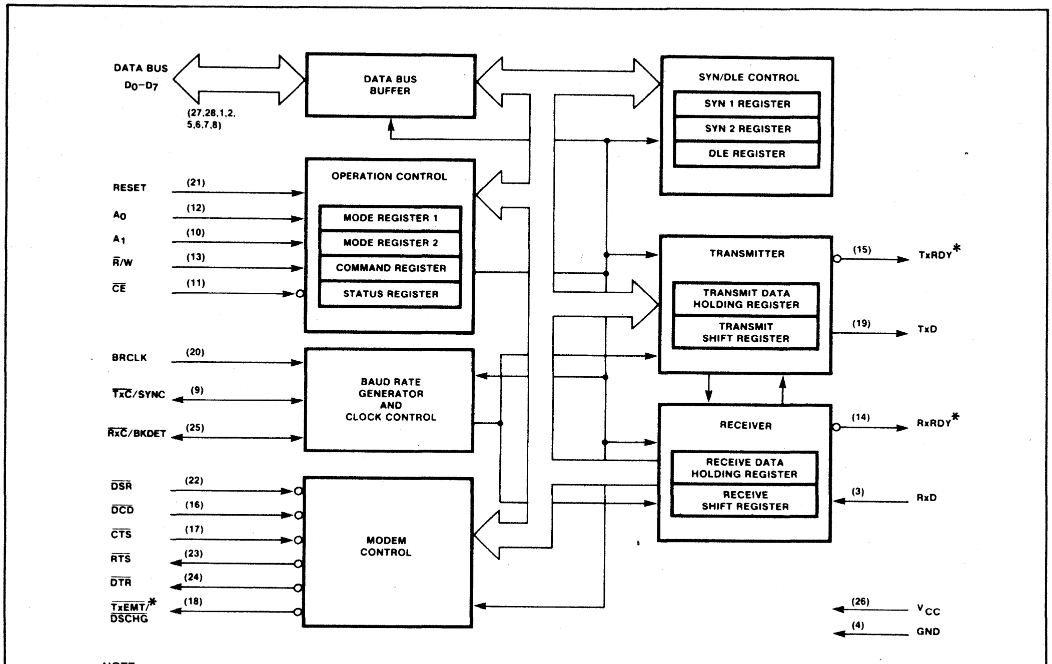
C SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
D2	1	28	D1
D3	2	27	D0
RxD	3	26	V_{CC}
GND	4	25	$\overline{\text{RxC}}$/BKDET
D4	5	24	$\overline{\text{DTR}}$
D5	6	23	$\overline{\text{RTS}}$
D6	7	22	$\overline{\text{DSR}}$
D7	8	21	RESET
$\overline{\text{TxC}}$/XSYNC	9	20	BRCLK
A1	10	19	TxD
$\overline{\text{CE}}$	11	18	$\overline{\text{TxEMT}}$/$\overline{\text{DSCHG}}$
A0	12	17	$\overline{\text{CTS}}$
$\overline{\text{R}}$/W	13	16	$\overline{\text{DCD}}$
RxRDY	14	15	$\overline{\text{TxRDY}}$

ADI-868-RI

BLOCK DIAGRAM



NOTE
* Open drain output pin.

BLOCK DIAGRAM

The EPCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS
Set A (BRCLK = 4.9152MHz)

MR23–20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS (Cont'd)

Set B (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

Set C (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC.

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
Ceramic DIP	MC2661A/MC68661A MC2661B/MC68661B MC2661C/MC68661C	See table 1 for baud rates
Plastic DIP	MC2661A/MC68661A MC2661B/MC68661B MC2661C/MC68661C	See table 1 for baud rates

Table 2 CPU-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal EPCI registers.
$\overline{R}$/W	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{R}$/W, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the three-state condition.
D_7-D_0	8,7,6,5, 2,1,28,17	I/O	8-bit, three-state data bus used to transfer commands, data and status between EPCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$	15	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$	14	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}$/ $\overline{DSCHG}$	18	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

OPERATION

The functional operation of the 2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2661 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{RxC}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27-MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the break detection timing diagram.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/ OUTPUT	FUNCTION
BRCLK	20	I	Clock input to the internal baud rate generator (see table 1). Not required if external receiver and transmitter clocks are used.
•$\overline{\text{RxC}}$/BKDET	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
•$\overline{\text{TxC}}$/XSYNC	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output or an external jam synchronization input.
RxD	3	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{\text{DSR}}$	22	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1.
$\overline{\text{DCD}}$	16	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ when its state changes if CR2 or CR0 = 1. If $\overline{\text{DCD}}$ goes high while receiving, the RxC is internally inhibited.
$\overline{\text{CTS}}$	17	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{\text{DTR}}$	24	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{\text{RTS}}$	23	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. If the transmit shift register is not empty when CR5 is reset (1 to 0), then $\overline{\text{RTS}}$ will go high one TxC time after the last serial bit is transmitted.

NOTE
•$\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ outputs have short circuit protection max. C_L = 100pF. Outputs become open circuited upon detection of a zero pulled high or a one pulled low.

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN(CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization.) When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{\text{RxRDY}}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27-MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{\text{CTS}}$ input is low and the TxEN command register bit is set. The 2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

In the synchronous mode, when the 2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the 2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the intialization process appears in figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{\text{CE}}$, $\overline{\text{R}}$/W, A_1 and A_0 inputs. The conditions necessary to address each register are shown in table 4.

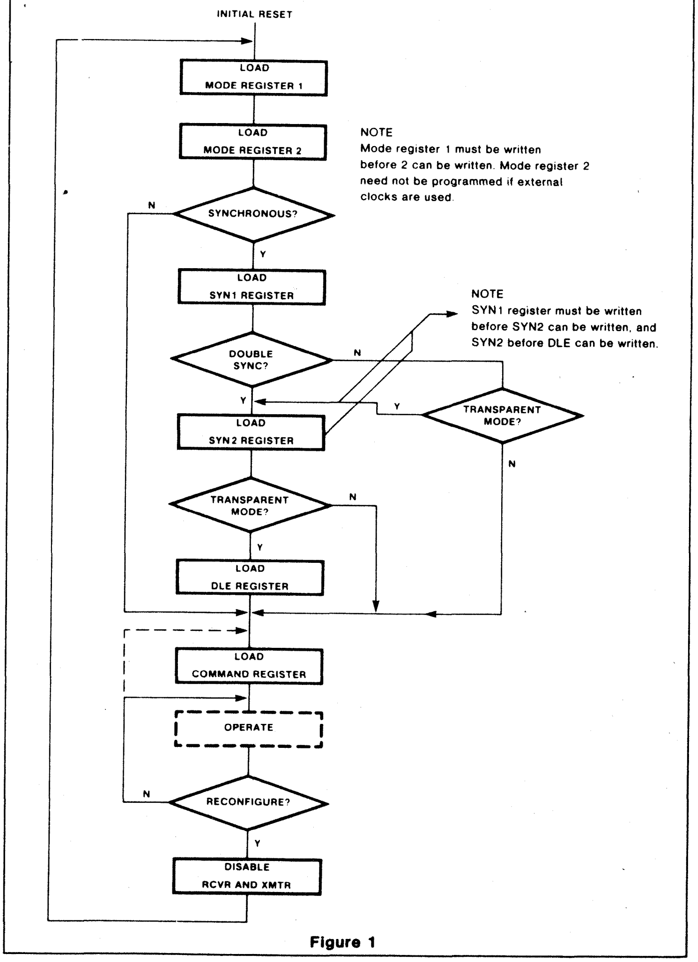
The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and

Table 4 MC2661/MC68661 REGISTER ADDRESSING

$\overline{\text{CE}}$	A_1	A_0	$\overline{\text{R}}$/W	FUNCTION
1	X	X	X	Three-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC characteristics section for timing requirements.

MC2661/MC68661 INITIALIZATION FLOW CHART



NOTE
Mode register 1 must be written before 2 can be written. Mode register 2 need not be programmed if external clocks are used.

NOTE
SYN1 register must be written before SYN2 can be written, and SYN2 before DLE can be written.

Figure 1

$\overline{\text{R}}$/W = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The 2661 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12-MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN=0 or when TxEN = 1 and the transmitter is marking in half duplex mode (RxEN = 0). To effect assembly/disassembly of the next received/transmitted character, MR12-15 must be changed within n bit times of the active going state of $\overline{\text{RxRDY}}$/$\overline{\text{TxRDY}}$. Transparent and non-transparent mode changes (MR16) must occur within n-1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (A, B, C). Versions A and B specify a 4.9152 MHz TTL input at BRCLK (pin 20); version C specifies a 5.0688 MHz input which is identical to the Signetics 2651. MR23-20 are don't cares if external clocks are selected (MR25-MR24 = 0). The individual rates are given in table 1.

MR24-MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 9 and 25. Refer to table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{\text{RxC}}$ rising edge. Disabling the receiver causes $\overline{\text{RxRDY}}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state

Table 5 MODE REGISTER 1 (MR 1)

MR17	MR16	MR15	MR14	MR13 MR12	MR11 MR10
Sync/Async		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor
Async: Stop Bit Length 00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate
Sync: Number of SYN char 0 = Double SYN 1 = Single SYN	**Sync: Transparency Control** 0 = Normal 1 = Transparent				

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6 MODE REGISTER 2 (MR2)

MR27–MR24											MR23–MR20
	TxC	RxC	Pin 9	Pin 25		TxC	RxC	Pin 9	Pin 25	Mode	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	XSYNC^1	RxC/TxC	sync	See baud rates in table 1
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	async	
0010	I	E	1X	RxC	1010	I	E	XSYNC^1	RxC	sync	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	async	
0100	E	E	TxC	RxC	1100	E	E	XSYNC^1	RxC/TxC	sync	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	async	
0110	I	E	16X	RxC	1110	I	E	XSYNC^1	RxC	sync	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	async	

NOTES

1. When pin 9 is programmed as XSYNC input, SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled.

E = External clock
I = Internal clock (BRG)
1X and 16X are clock outputs

Table 7 COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	Request To Send	Reset Error	Sync/Async	Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{RTS}$ output high one clock time after TxSR serialization 1 = Force $\overline{RTS}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect)	**Async:** **Force break** 0 = Normal 1 = Force break	0 = Disable 1 = Enable	0 = Force $\overline{DTR}$ output high 1 = Force $\overline{DTR}$ output low	0 = Disable 1 = Enable
			Sync: **Send DLE** 0 = Normal 1 = Send DLE			

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxEMT/DSCHG	RxRDY	TxRDY
0 = $\overline{DSR}$ input is high 1 = $\overline{DSR}$ input is low	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	Async: 0 = Normal 1 = Framing Error	0 = Normal 1 = Overrun Error	Async: 0 = Normal 1 = Parity error	0 = Normal 1 = Change in $\overline{DSR}$, or $\overline{DCD}$, or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty
		Sync: 0 = Normal 1 = SYN detected		Sync: 0 = Normal 1 = Parity error or DLE received			

(high) while $\overline{TxRDY}$ and $\overline{TxEMT}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{DTR}$ and $\overline{RTS}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE—non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

Table 9 MC2661/MC68661 EPCI vs SIGNETICS 2651 PCI

FEATURE	EPCI	PCI
1. MR2 Bit 6, 7	Control pin 9, 25	Not used
2. DLE detect-SR3	SR3 = 0 for DLE-DLE, DLE-SYNC1	SR3 = 1 for DLE-DLE, DLE-SYNC1
3. Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4. Send DLE-CR3	One time command	Reset via CR3 on next $\overline{\text{TxRDY}}$
5. DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6. SYNC1 stripping in double sync non-transparent mode	All SYNC1	First SYNC1 of pair
7. Baud rate versions	Three	One
8. Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{\text{TxRDY}}$ changing from 0 to 1	Reset CR0 when $\overline{\text{TxEMT}}$ goes from 1 to 0. Then reset CR5 when $\overline{\text{TxEMT}}$ goes from 0 to 1
9. Break detect	Pin 25[1]	FE and null character
10. Stop bit searched	One	Two
11. External jam sync	Pin 9[2]	No
12. Data bus timing	Improved over 2651	—
13. Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES

1. Internal BRG used for RxC.
2. Internal BRG used for TxC.

When CR5 (RTS) is set, the $\overline{\text{RTS}}$ pin is forced low and the transmit serial logic is enabled. A 1 to 0 transition of CR5 will cause $\overline{\text{RTS}}$ to go high (inactive) one TxC time after the last serial bit has been transmitted (if the transmit shift register was not empty).

The EPCI can operate in one of four sub-modes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7-CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{\text{TxRDY}}$ output = 1.
4. The $\overline{\text{TxEMT/DSCHG}}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loop back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{\text{DTR}}$ is connected to $\overline{\text{DCD}}$ and $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{\text{DTR}}$, $\overline{\text{RTS}}$ and $\overline{\text{TxD}}$ outputs are held high.
5. The $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{DSR}}$ and RxD inputs are ignored.

Additional requirements to operate in the local loop back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loop back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{\text{RxRDY}}$, $\overline{\text{TxRDY}}$, and $\overline{\text{TxEMT/DSCHG}}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{\text{TxRDY}}$ output pin is low. In the automatic echo and remote loop back modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\text{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit, i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{\text{DCD}}$ and $\overline{\text{DSR}}$ inputs respectively. A low input sets its corresponding status bit, and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−55 to +150	°C
All voltages with respect to ground[3]	−0.3 to +7.0	V

THERMAL CHARACTERISTICS

CHARACTERISTIC	SYMBOL	VALUE	UNIT
Thermal Resistance			
Ceramic		50	
Plastic	θ_{JA}	100	°C/W
Cerdip		60	

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5% [4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low		−0.3		0.8	
V_{IH}	High		2.0		V_{CC}	
	Output voltage					V
V_{OL}	Low	I_{OL} = 2.2mA			0.4	
V_{OH}[7]	High	I_{OH} = −400μA	2.4			
I_{IL}	Input leakage current	V_{IN} = 0 to 5.5 V			10	μA
	3-state output leakage current					μA
I_{LH}	Data bus high	V_{OUT} = 0 to 5.25 V			10	
I_{LL}	Data bus low				10	
I_{CC}	Power supply current				150	mA

CAPACITANCE T_A = 25°C, V_{CC} = 0V

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance	V_{IN} = V_{OUT} = 0 V				pF
C_{IN}	Input				20	
C_{OUT}	Output	fc = 1MHz			20	
$C_{I/O}$	Input/Output	Unmeasured pins tied to ground			20	

Notes on following page.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C \text{ to } +70°C, V_{CC} = 5.0V \pm 5\%$ [4,5,6]

PARAMETER		TEST CONDITIONS	Min	Typ	Max	UNIT
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		250			
	Setup and hold time					ns
t_{AS}	Address setup		10			
t_{AH}	Address hold		10			
t_{CS}	R/W control setup		10			
t_{CH}	R/W control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	$C_L = 150pF$			200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$			100	
t_{CED}	CE to CE delay		600			
	Input clock frequency					MHz
f_{BRG}	Baud rate generator (MC2661A,B/MC68661A,B)		1.0	4.9152	4.9202	
f_{BRG}	Baud rate generator (MC2661C/MC68661C)		1.0	5.0688	5.0738	
$f_{R/T}$	TxC or RxC		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high (MC2661A,B/MC68661A,B)		75			
t_{BRH}[9]	Baud rate high (MC2661C/MC68661C)		70			
t_{BRL}[9]	Baud rate low (MC2661A,B/MC68661A,B)		75			
t_{BRL}[9]	Baud rate low (MC2661C/MC68661C)		70			
$t_{R/TH}$	TxC or RxC high		480			
$t_{R/TL}$	TxC or RxC low		480			
t_{TXD}	TxD delay from falling edge of TxC	$C_L = 150pF$			650	ns
t_{TCS}	Skew between TxD changing and falling edge of TxC output[8]	$C_L = 150pF$	TBD		TBD	

NOTES
1. 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 at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (IQ ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8 V and 2.0 V for outputs. Input levels swing between 0.4 V and 2.4 V, with a transition time of 20ns maximum.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. TxRDY, RxRDY and TxEMT/DSCHG outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688 MHz f_{BRG} (MC2661C/MC68661C) and 4.9152 MHz f_{BRG} (MC2661A,B/MC68661A,B), f_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.

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:

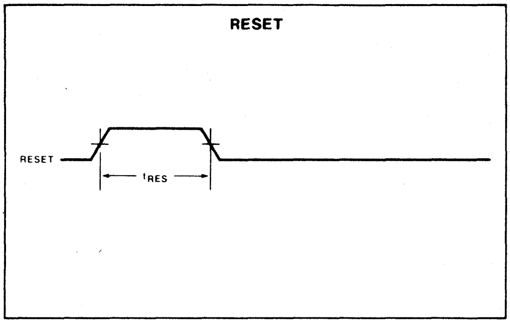
$$P_D = K \div (T_J + 273°C) \quad (2)$$

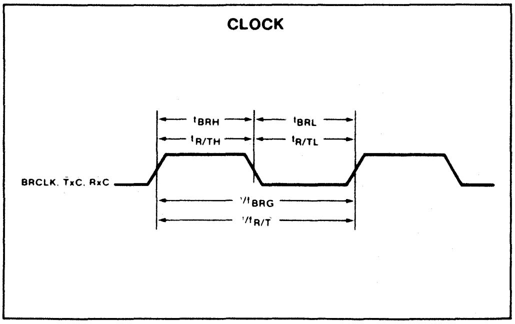
Solving equations 1 and 2 for K gives:

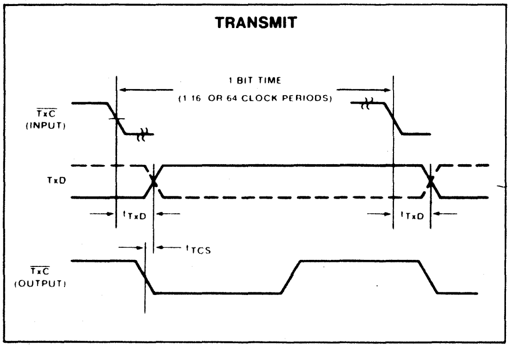
$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

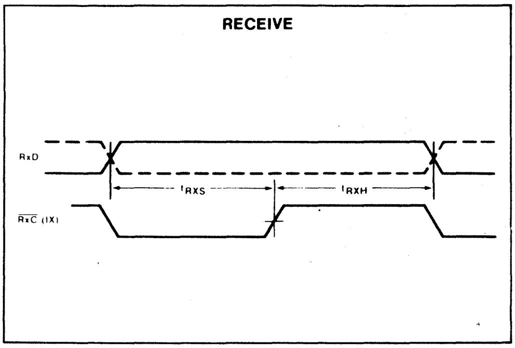
Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

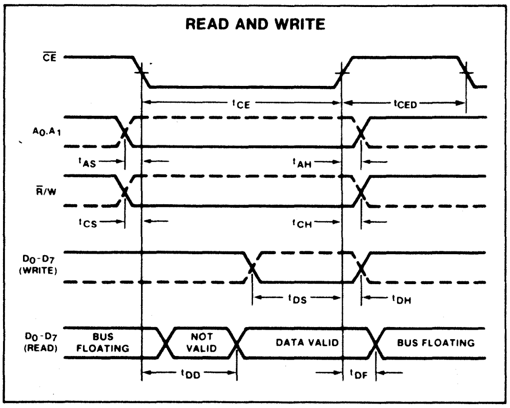
TIMING DIAGRAMS


RESET


CLOCK

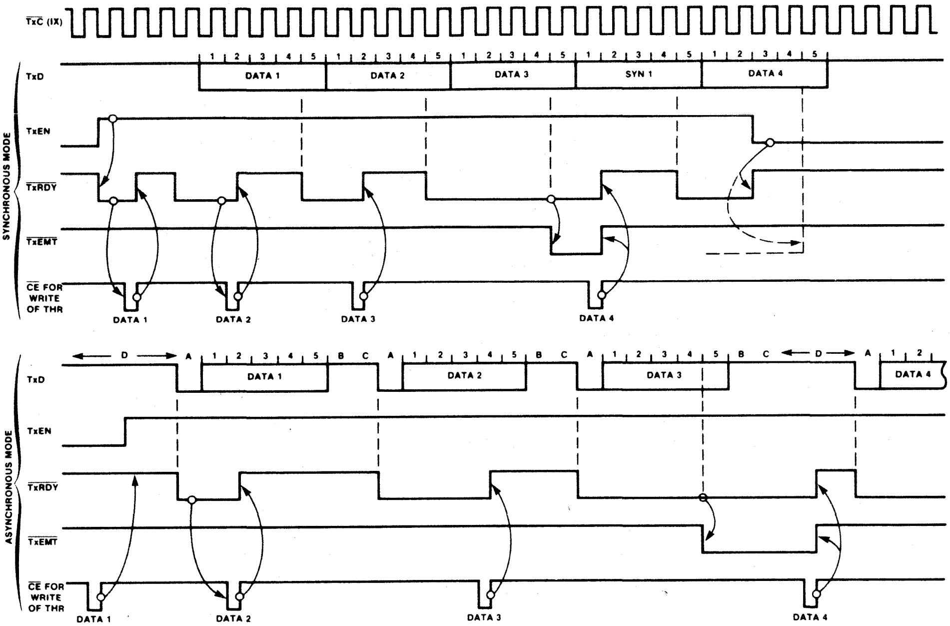

TRANSMIT


RECEIVE


READ AND WRITE

TIMING DIAGRAMS (Cont'd)

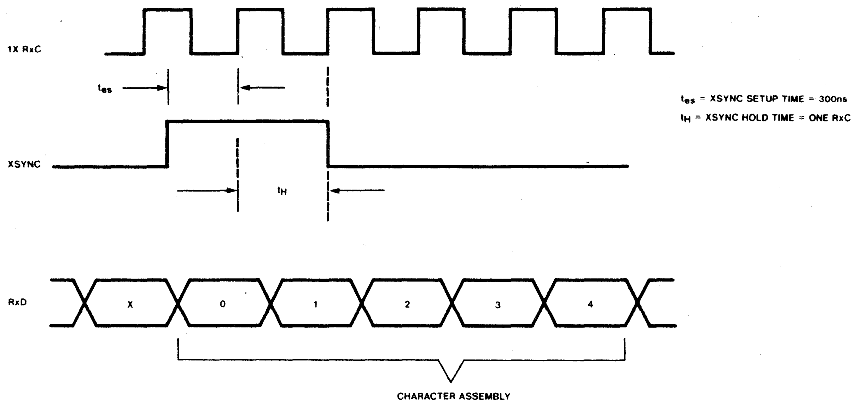
TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])



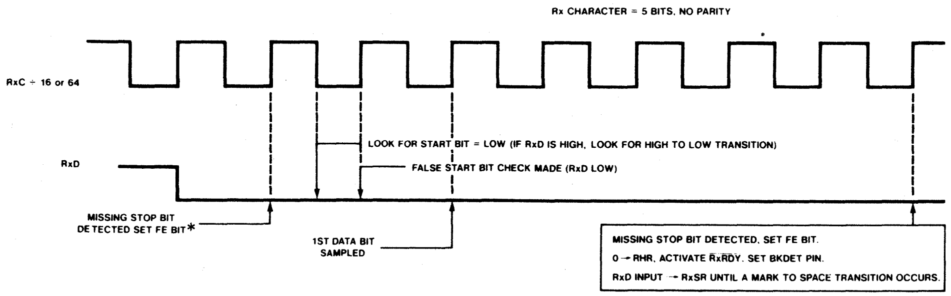
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

TIMING DIAGRAMS (Cont'd)

EXTERNAL SYNCHRONIZATION WITH XSYNC



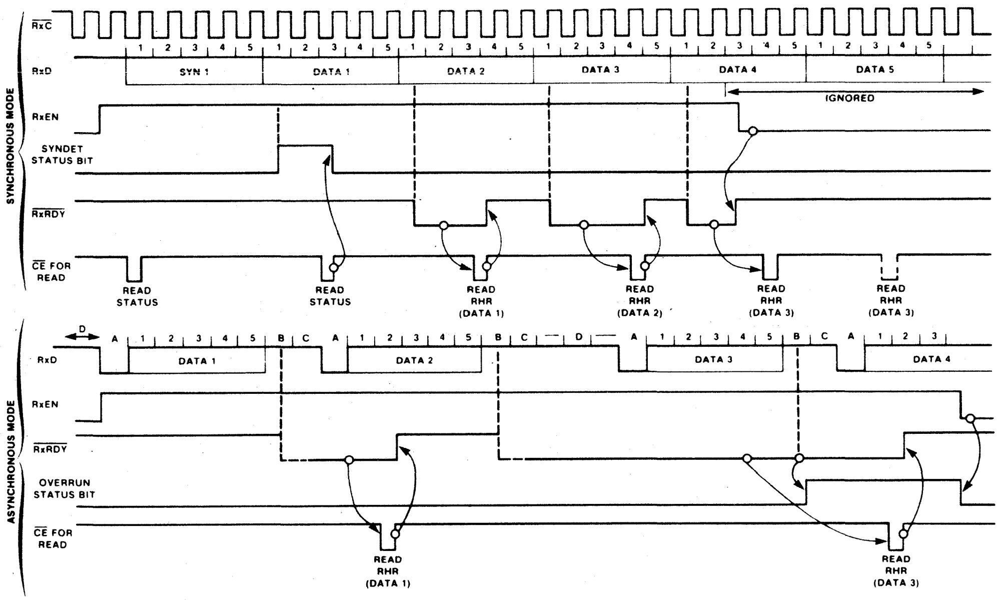
BREAK DETECTION TIMING



NOTE
* If the stop bit is present, the start bit search will commence immediately.

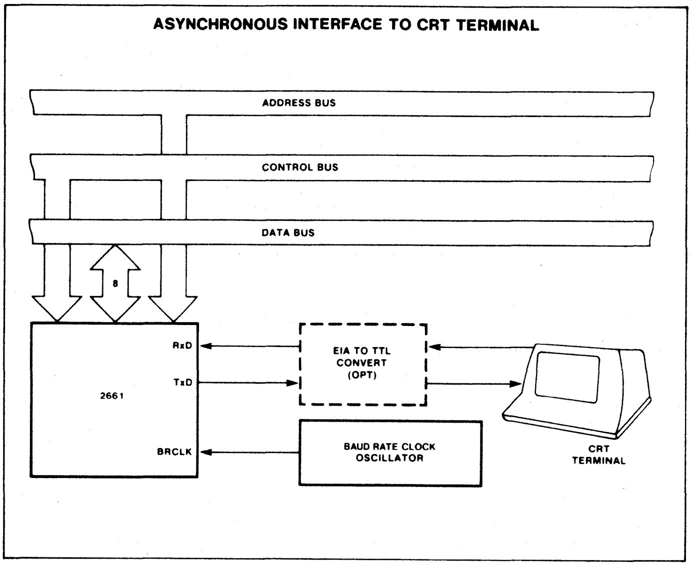
TIMING DIAGRAMS (Cont'd)

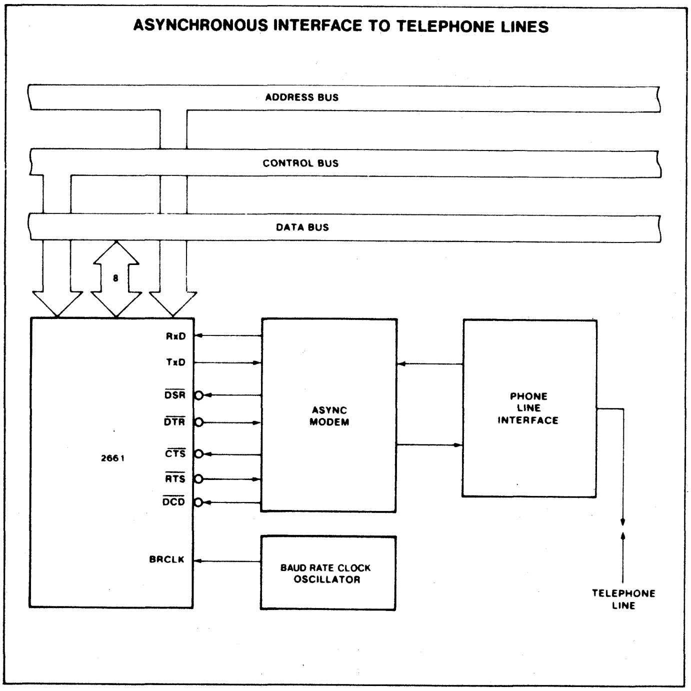
RxRDY (Shown for 5-bit characters, no parity, 2 stops bits [in asynchronous mode])



NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected.

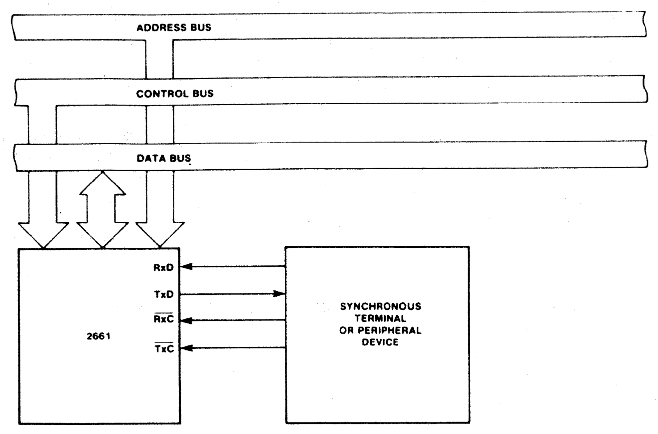
TYPICAL APPLICATIONS



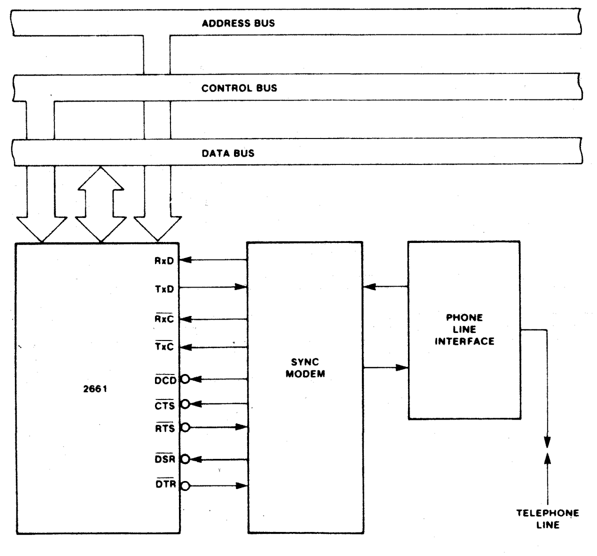


TYPICAL APPLICATIONS (Cont'd)

SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES



MC68000 MPU-TO-EPCI INTERFACE REQUIREMENTS

The circuit shown in Figure 2 interfaces the EPCI to the MC68000 MPU. The 8-bit data bus of the EPCI is connected to the low order 8 bits of the MPU data bus (D0-D7). Due to this, the EPCI's registers are addressed on word (even byte) boundaries and so address line A1 of the MPU is connected to the A0 address line of the EPCI. Similarly, A2 of the MPU is connected to A1 of the EPCI. R/$\overline{\text{W}}$ on the MC68000 is inverted and connected to $\overline{\text{R}}$/W of the EPCI.

The $\overline{\text{CE}}$ signal must be generated for the EPCI and the $\overline{\text{DTACK}}$ signal must be supplied to the MPU. To allow for the data setup time on a read of the EPCI, $\overline{\text{CE}}$ must be delayed one-half clock cycle and $\overline{\text{DTACK}}$ generated on the next rising edge of the system clock. This causes the processor to insert one wait state in the bus cycle. In addition to this, $\overline{\text{CE}}$ must not be reasserted until the chip enable period t_{CE} has expired. Since some instructions on the MC68000 can cause access to consecutive addresses on consecutive bus cycles (e.g., MOVEP), an INHIBIT signal must be generated to hold-off an access during this period. A state machine consisting of a 74LS161 binary counter and a 74LS74 D flip-flop is configured as a digital "one shot." The rising edge of $\overline{\text{CE}}$ starts the counter which times out after given number of clock cycles. Since t_{CE} is 600 ns, a minimum of 5 clock cycles at 8 MHz (625 ns) is required. The timing for two consecutive read bus cycles is shown in Figure 3. The INHIBIT signal prevents $\overline{\text{CE}}$ from being generated and $\overline{\text{DTACK}}$ from being asserted, causing the processor to generate wait states until INHIBIT is negated.

M6809 FAMILY MPU-TO-EPCI INTERFACE REQUIREMENTS

The M6809 family of microprocessors can be easily interfaced to the EPCI as shown in Figure 4.

FIGURE 2 — MC68000-TO-EPCI INTERFACE — EXAMPLE CIRCUIT

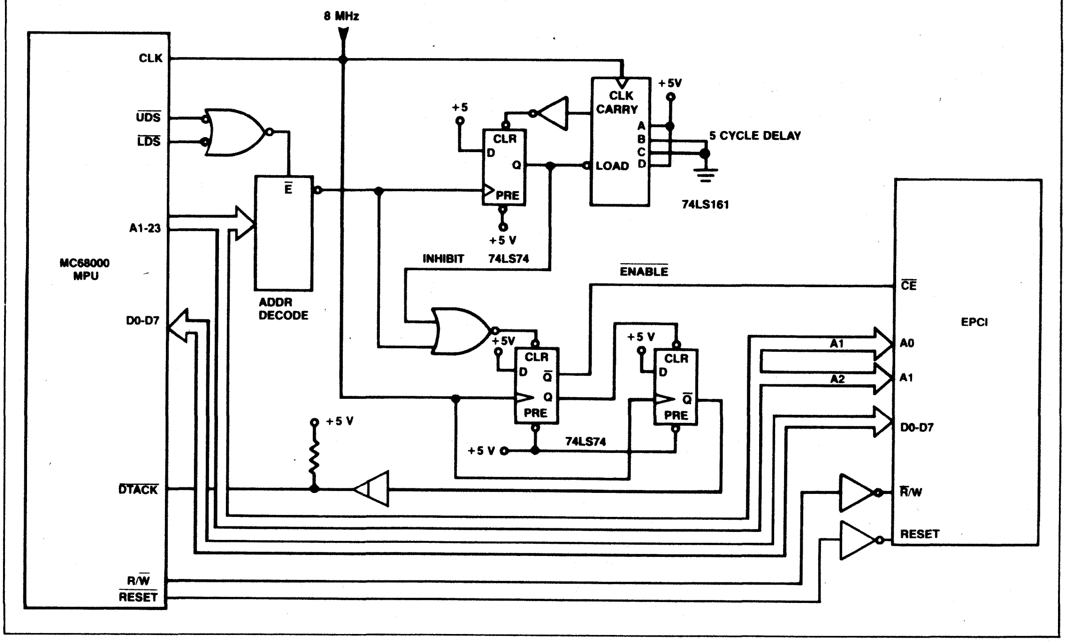
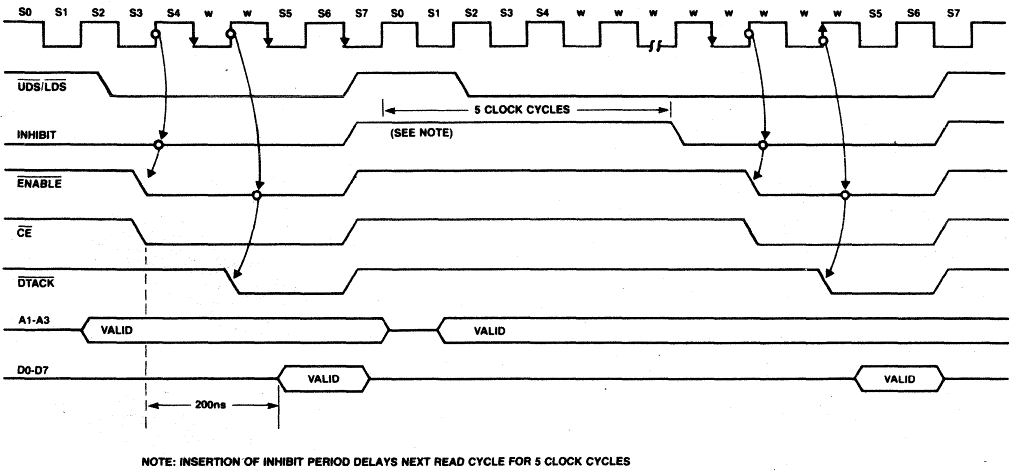
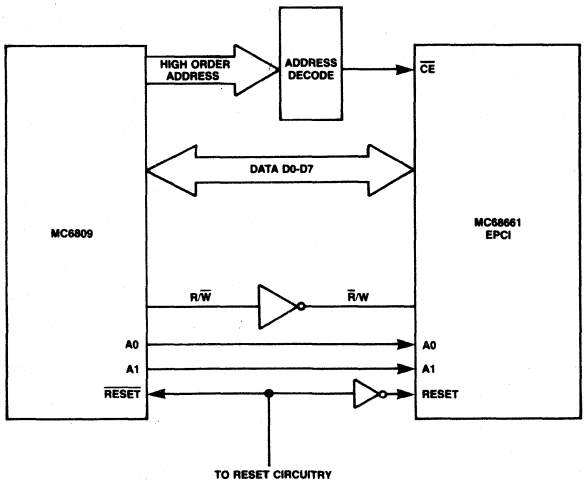


FIGURE 3 — MC68000-TO-EPCI READ BUS CYCLE TIMING



NOTE: INSERTION OF INHIBIT PERIOD DELAYS NEXT READ CYCLE FOR 5 CLOCK CYCLES

FIGURE 4 — INTERFACE CONNECTIONS TO MC6809



MOTOROLA

MC6805P6

Advance Information

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. A comparison of the key features of several members of the M6805 Family is shown on the last page of this data sheet. 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

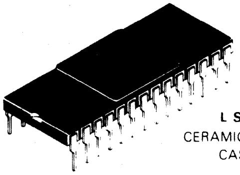
USER SELECTABLE OPTIONS

- 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

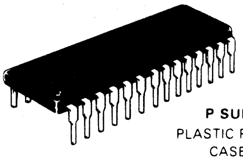
HMOS

(HIGH DENSITY
N CHANNEL, SILICON-GATE
DEPLETION LOAD)

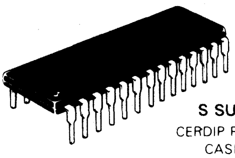
8-BIT MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

FIGURE 1 — PIN ASSIGNMENTS

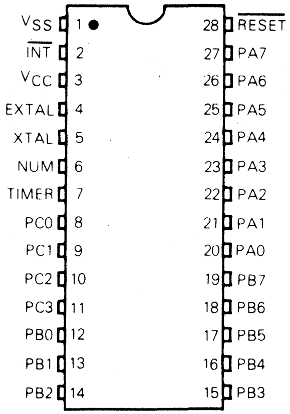
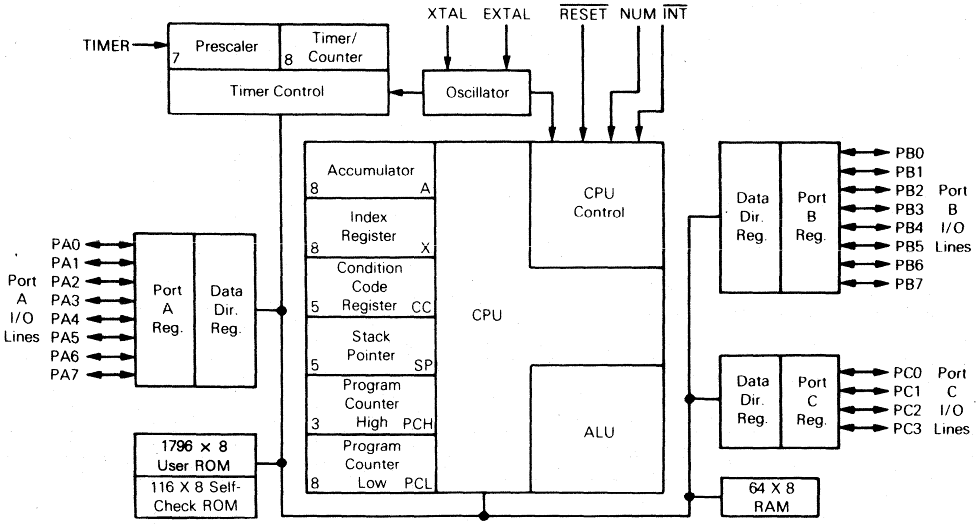


FIGURE 2 — MC6805P6 HMOS MICROCOMPUTER BLOCK DIAGRAM



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 Ceramic Cerdip	T_J	150 175 175	°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 $V_{out}) \leq V_{CC}$. 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 Ceramic Cerdip	θ_{JA}	120 50 60	°C/W

POWER CONSIDERATIONS

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

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

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:

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

Solving equations 1 and 2 for K gives:

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

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

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					
$\overline{\text{RESET}}$ ($4.75 \leq V_{CC} \leq 5.75$)	V_{IH}	4.0	–	V_{CC}	V
($V_{CC} < 4.75$)		$V_{CC}-0.5$	–	V_{CC}	
$\overline{\text{INT}}$ ($4.75 \leq V_{CC} \leq 5.75$)		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}	V
Self-Check Mode		–	9.0	15.0	
Input Low Voltage					
$\overline{\text{RESET}}$	V_{IL}	−0.3	–	0.8	V
$\overline{\text{INT}}$		−0.3	*	1.5	
All Other		−0.3	–	0.8	
$\overline{\text{RESET}}$ Hysteresis Voltage (See Figures 11, 12, and 13)					
"Out of Reset"	V_{IRES+}	2.1	–	4.0	V
"Into Reset"	V_{IRES-}	0.8	–	2.0	
$\overline{\text{INT}}$ 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	V_{LVI}	–	3.5	–	V
Input Current					
TIMER (V_{in} = 0.4 V)	I_{in}	–	–	20	μA
$\overline{\text{INT}}$ (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	
$\overline{\text{RESET}}$ (V_{in} = 0.8 V)		−4.0	–	−50	
(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° 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}	3.5	–	–	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}	0.3	–	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	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}	−0.3	–	0.8	V
Hi-Z State Input Current	I_{TSI}	–	2	20	μA

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
$\overline{\text{INT}}$ and TIMER Pulse Width (See Interrupt Section)	t_{WL}, t_{WH}	t_{cyc} + 250			ns
$\overline{\text{RESET}}$ Pulse Width	t_{RWL}	t_{cyc} + 250			ns
$\overline{\text{RESET}}$ Delay Time (External Capacitance = 1.0 µF)	t_{RHL}		100		ms
$\overline{\text{INT}}$ Zero Crossing Detection Input Frequency (±5° Accuracy)	f_{INT}	0.03		1.0	kHz
External Clock Input Duty Cycle (EXTAL)		40	50	60	%

FIGURE 3 — TTL EQUIVALENT TEST LOAD (PORT B)

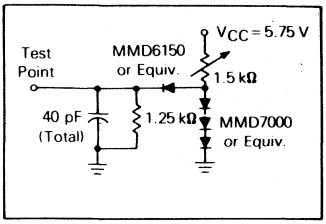


FIGURE 4 — CMOS EQUIVALENT TEST LOAD (PORT A)

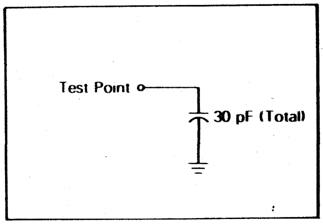
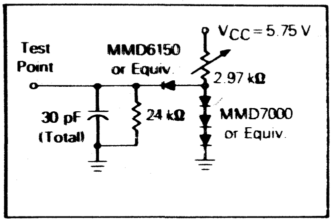


FIGURE 5 — TTL EQUIVALENT TEST LOAD (PORTS A AND C)



SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in Figure 1, 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.

$\overline{\text{INT}}$

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.

$\overline{\text{RESET}}$

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 6, 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 7. 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.

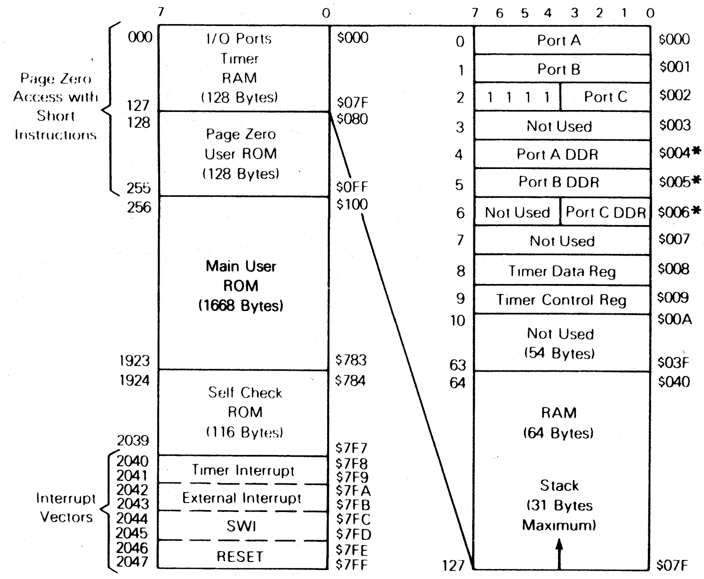
CENTRAL PROCESSING UNIT

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

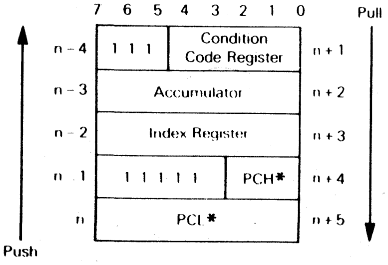
The M6805 Family CPU has five registers available to the programmer. They are shown in Figure 8 and are explained in the following paragraphs.

FIGURE 6 — MCU ADDRESS MAP



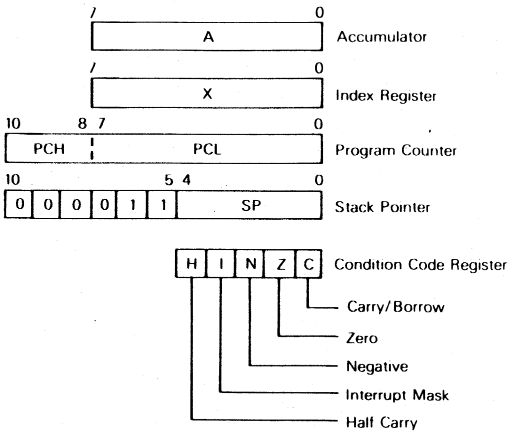
*Caution: Data direction registers (DDRs) are write-only; they read as $FF.

FIGURE 7 — INTERRUPT STACKING ORDER



*For subroutine calls, only PCL and PCH are stacked.

FIGURE 8 — 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 ($\overline{\text{INT}}$). 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 9. 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.

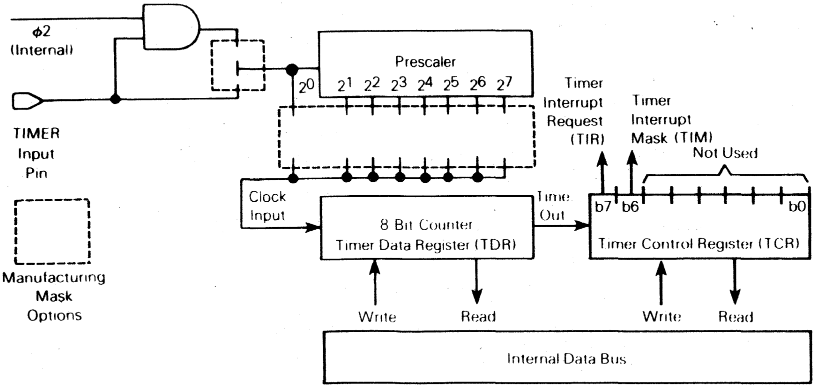
The clock input to the timer can be from an external source (decrementing of timer counter occurs on a positive transition of the external source) applied to the TIMER input pin or it can be the internal $\phi 2$ signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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 9 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 10 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 (RESET), and by an optional internal low voltage detect circuit; see Figure 11. The internal circuit connected to the RESET pin consists of a Schmitt trigger which senses the RESET line logic level. The Schmitt trigger provides an internal reset voltage if it senses a logic "0" on the RESET pin. During power-up, the Schmitt trigger switches on (removes reset) when the RESET pin voltage rises to V_{IRES+}. When the 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 12.

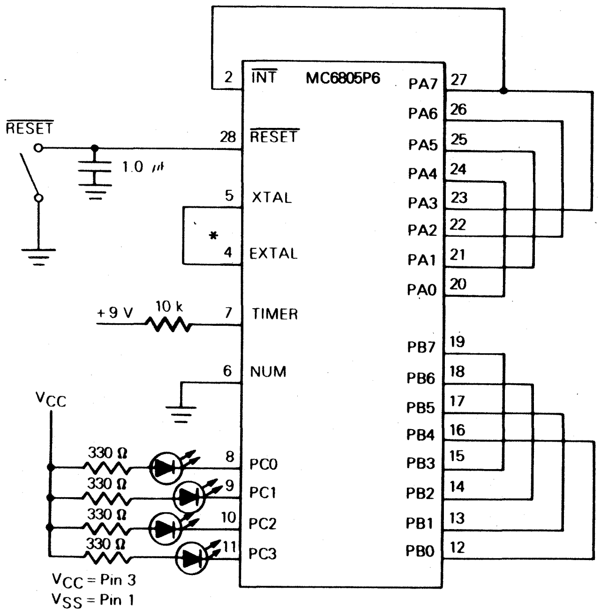
During power-up, a delay of t_{RHL} is needed before allowing the RESET input to go high. This time allows the internal clock generator to stabilize. Connecting a capacitor to the RESET input, as shown in Figure 13, typically provides sufficient delay. See Figure 17 under Interrupts section for the complete 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 14. The crystal specifications and suggested PC board layouts

FIGURE 10 — SELF-CHECK CONNECTIONS



*This connection depends on the clock oscillator user selectable mask option. Use crystal if that 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 11 — POWER AND $\overline{\text{RESET}}$ TIMING

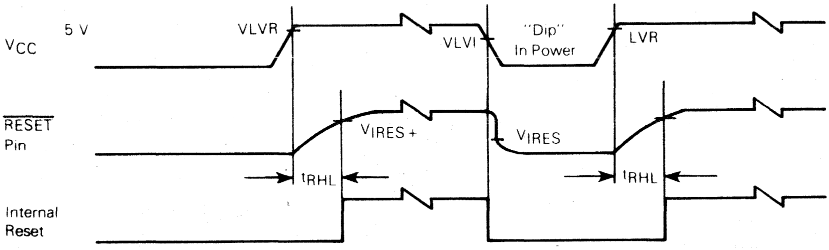


FIGURE 12 — TYPICAL RESET SCHMITT TRIGGER HYSTERESIS

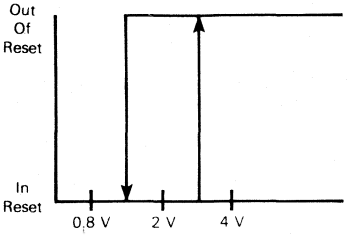


FIGURE 13 — POWER-UP RESET DELAY CIRCUIT

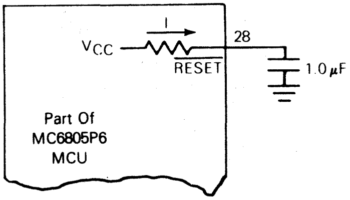
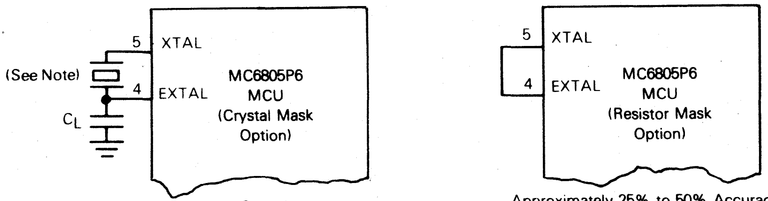
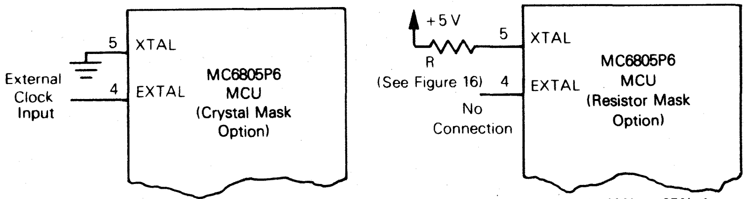


FIGURE 14 — CLOCK GENERATOR OPTIONS



Crystal

Approximately 25% to 50% Accuracy
Typical $t_{cyc} = 1.25\ \mu s$
External Jumper

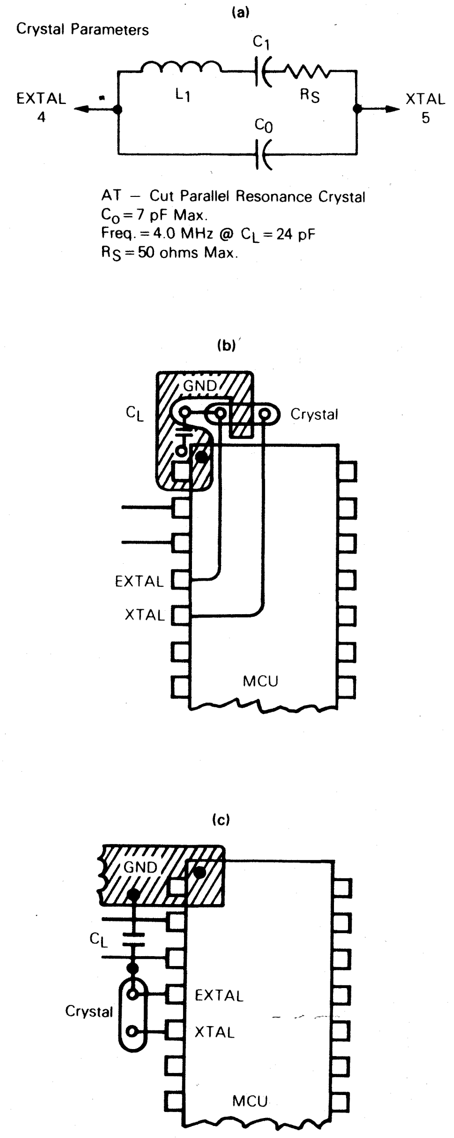


External Clock

Approximately 10% to 25% Accuracy
External Resistor
(Excludes Resistor Tolerence)

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.

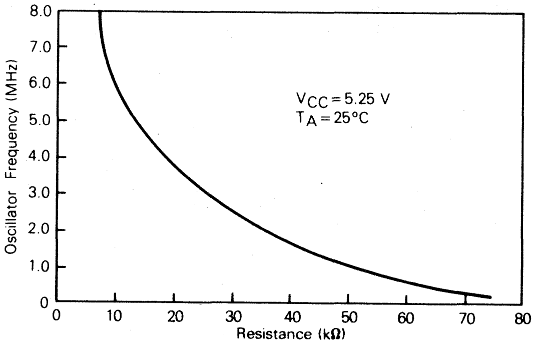
FIGURE 15 — CRYSTAL MOTIONAL ARM PARAMETERS AND SUGGESTED PC BOARD LAYOUT



AT — Cut Parallel Resonance Crystal
$C_0 = 7$ 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 16 — TYPICAL FREQUENCY SELECTION FOR RESISTOR OSCILLATOR OPTION



INTERRUPTS

The MC6805P6 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 17. 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 $\overline{\text{RESET}}$, 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 (f_{INT} maximum) can be used to generate an external interrupt, as shown in Figure 18(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. 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 $\overline{\text{INT}}$ pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or $\overline{\text{INT}}$ 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 17 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

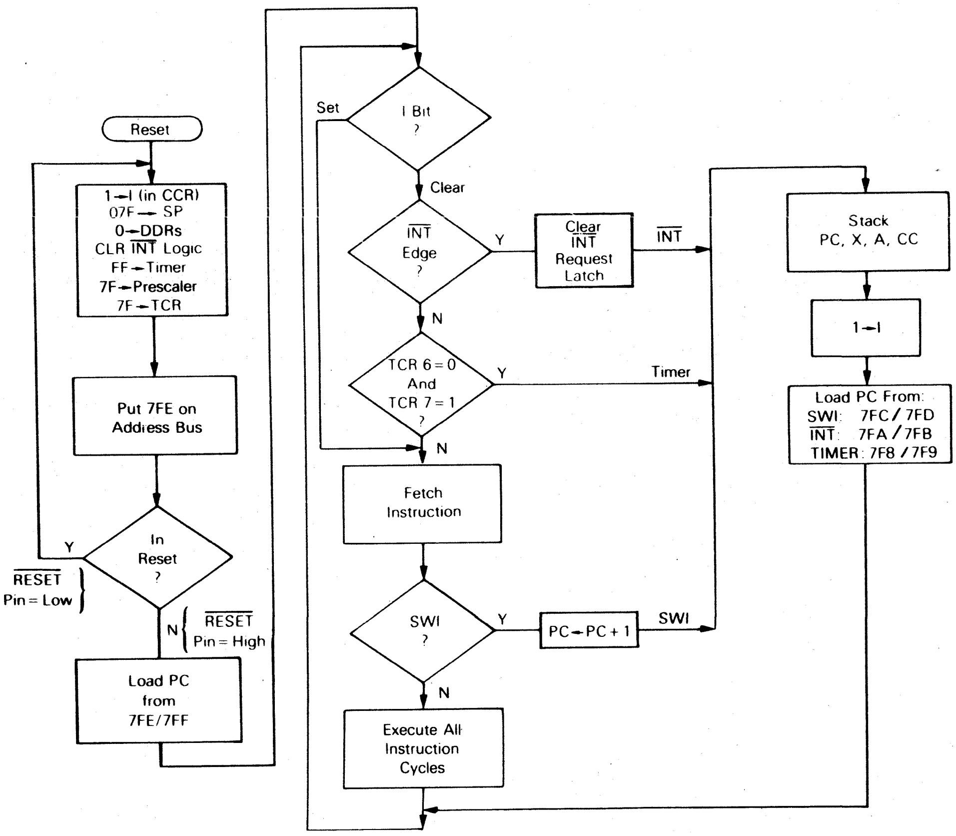
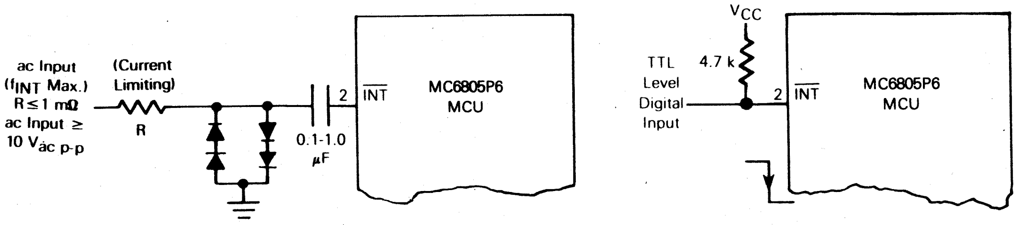


FIGURE 18 — TYPICAL INTERRUPT CIRCUITS



(a) Zero-Crossing Interrupt

(b) Digital-Signal Interrupt

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 18(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. SWIs are usually used as breakpoints for debugging or as system calls.

INPUT/OUTPUT

There are 20 input/output pins. The $\overline{\text{INT}}$ 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 19. When port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

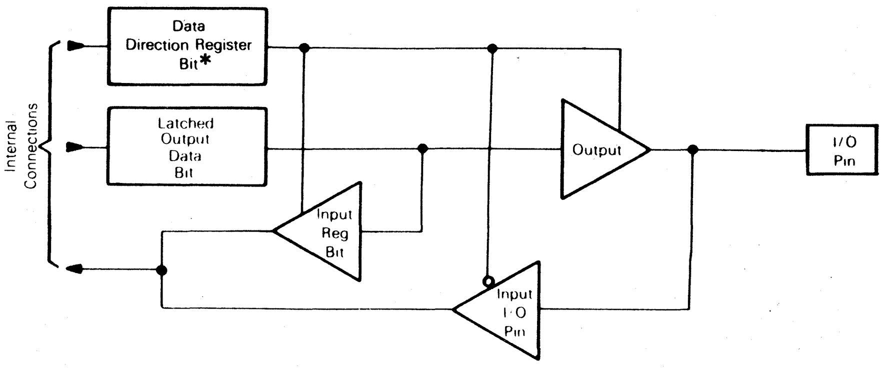
All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A may be made CMOS compatible as outputs with a mask option. The address map in Figure 6 gives the address of data registers and DDRs. The register configuration is provided in Figure 20 and Figure 21 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 19) 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 19 – TYPICAL PORT I/O CIRCUITRY

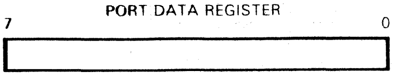


Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z **	Pin

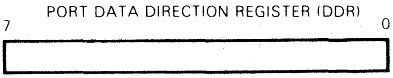
*DDR is a write-only register and reads as all "1s".

**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.

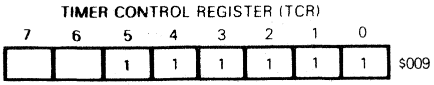
FIGURE 20 — MCU REGISTER CONFIGURATION



Port A Addr = $000
Port B Addr = $001
Port C Addr = $002 (Bits 0→3)



(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 (Bits 0→3)



TCR7 – Timer Interrupt Status 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.

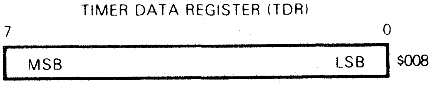
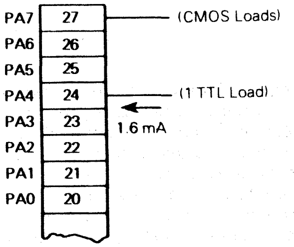
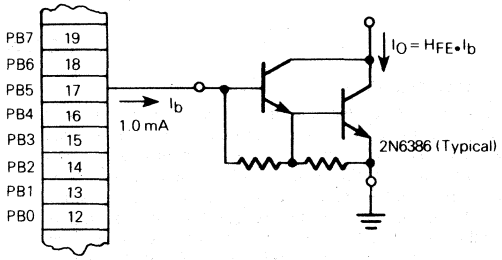


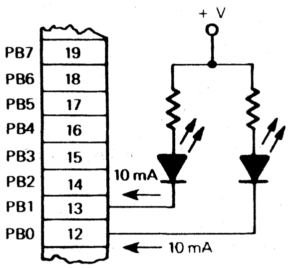
FIGURE 21(a) — TYPICAL OUTPUT MODE PORT CONNECTIONS



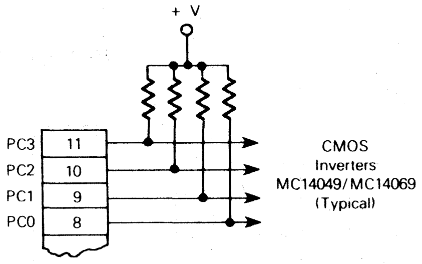
Port A, bit 7 and bit 4 programmed as output. Bit 7 driving CMOS loads and bit 4 driving one TTL load directly using CMOS output option.



Port B, bit 5 programmed as output, driving Darlington-base directly.

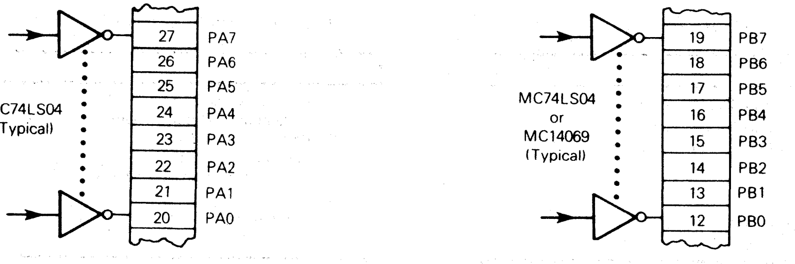


Port B, bit 0 and bit 1 programmed as output, driving LEDs directly.



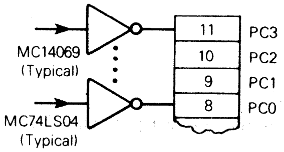
Port C, bits 0-3 programmed as output, driving CMOS loads, using external pullup resistors.

FIGURE 21(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.

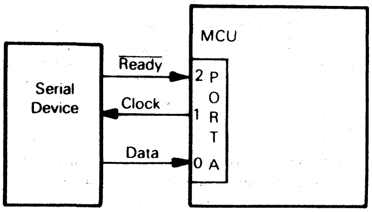
SOFTWARE

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 22 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 22 — BIT MANIPULATION EXAMPLE



```
                      .
                      .
                      .
SELF    BRSET     2, PORTA, SELF
                      .
                      .
                      .
        BSET      1, PORTA
        BRCLR     0, PORTA, CONT
CONT    BCLR      1, PORTA
        ASR       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			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						●	●	●	●	●
CLL	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
NEO	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										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	Λ	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero		

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

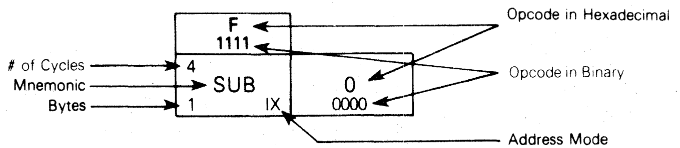
TABLE 7 — M6805 HMOS FAMILY OPCODE MAP

	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



ORDERING INFORMATION

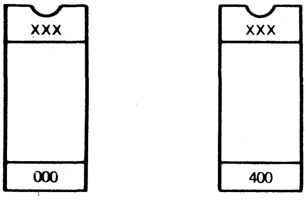
The information required when ordering a custom MCU is listed below. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

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 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.

EXORset is a trademark of Motorola Inc

MC6805P6 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 φ2 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 Oscillator Input
- ☐ Crystal
- ☐ Resistor

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 ______________________

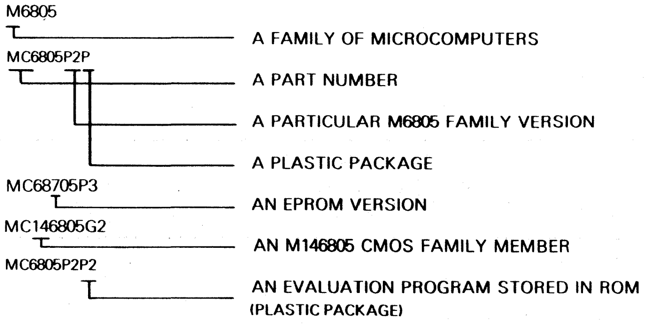
Signature ______________________

M6805 HMOS/146805 CMOS FAMILY

Features	MC6805P2	MC6805P4	MC6805P6	MC6805R2	MC6805R3	MC6805T2	MC6805U2	MC6805U3
Technology	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS	HMOS
Number of Pins	28	28	28	40	40	28	40	40
On-Chip RAM (Bytes)	64	112	64	64	112	64	64	112
On-Chip User ROM (Bytes)	1100	1100	1796	2048	3776	2508	2048	3776
External Bus	None	None	None	None	None	None	None	None
Bidirectional I/O Lines	20	20	20	24	24	19	24	24
Unidirectional I/O Lines	None	None	None	6 Inputs	6 Inputs	None	8 Inputs	8 Inputs
Other I/O Features	Timer	Timer	Timer	Timer, A/D	Timer, A/D	Timer, PLL	Timer	Timer
External Interrupt Inputs	1	1	1	2	2	2	2	2
EPROM Version	MC68705P3	MC68705P3	MC68705P3	MC68705R3	MC68705R3	None	MC68705U3	MC68705U3
STOP and WAIT	No	No	No	No	No	No	No	No

Features	MC68705P3	MC68705P5	MC68705R3	MC68705U3	MC146805E2	MC146805F2	MC146805G2	MC1468705G2
Technology	HMOS	HMOS	HMOS	HMOS	CMOS	CMOS	CMOS	CMOS
Number of Pins	28	28	40	40	40	28	40	40
On-Chip RAM (Bytes)	112	112	112	112	112	64	112	112
On-Chip User ROM (Bytes)	1804 EPROM	1804 EPROM	3776 EPROM	3776 EPROM	None	1080	2106	2106 EPROM
External Bus	None	None	None	None	Yes	None	None	None
Bidirectional I/O Lines	20	20	24	24	16	16	32	32
Unidirectional I/O Lines	None	None	6 Inputs	8 Inputs	None	4 Inputs	None	None
Other I/O Features	Timer	Timer	Timer,A/D	Timer	Timer	Timer	Timer	Timer
External Interrupt Inputs	1	1	2	2	1	1	1	1
EPROM Version	–	–	–	–	None	None	None	–
STOP and WAIT	No	No	No	No	Yes	Yes	Yes	Yes

M6805 FAMILY NOMENCLATURE



M6805 — A FAMILY OF MICROCOMPUTERS

MC6805P2P — A PART NUMBER

A PARTICULAR M6805 FAMILY VERSION

A PLASTIC PACKAGE

MC68705P3 — AN EPROM VERSION

MC146805G2 — AN M146805 CMOS FAMILY MEMBER

MC6805P2P2 — AN EVALUATION PROGRAM STORED IN ROM (PLASTIC PACKAGE)

MOTOROLA

MC3870

Advance Information

SINGLE-CHIP MICROCONTROLLER

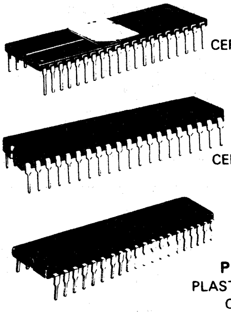
The MC3870 is a monolithic 8-bit microcomputer utilizing ion-implanted, N-channel silicon-gate technology and advanced circuit design techniques. The single-chip 3870 offers maximum cost effectiveness in a wide range of control and logic replacement applications.

- 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

MOS

(N-CHANNEL, SILICON-GATE
DEPLETION LOAD)

SINGLE-CHIP MICROCONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

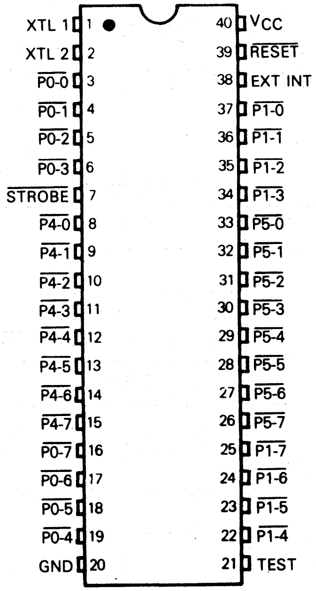
P SUFFIX
PLASTIC PACKAGE
CASE 711

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

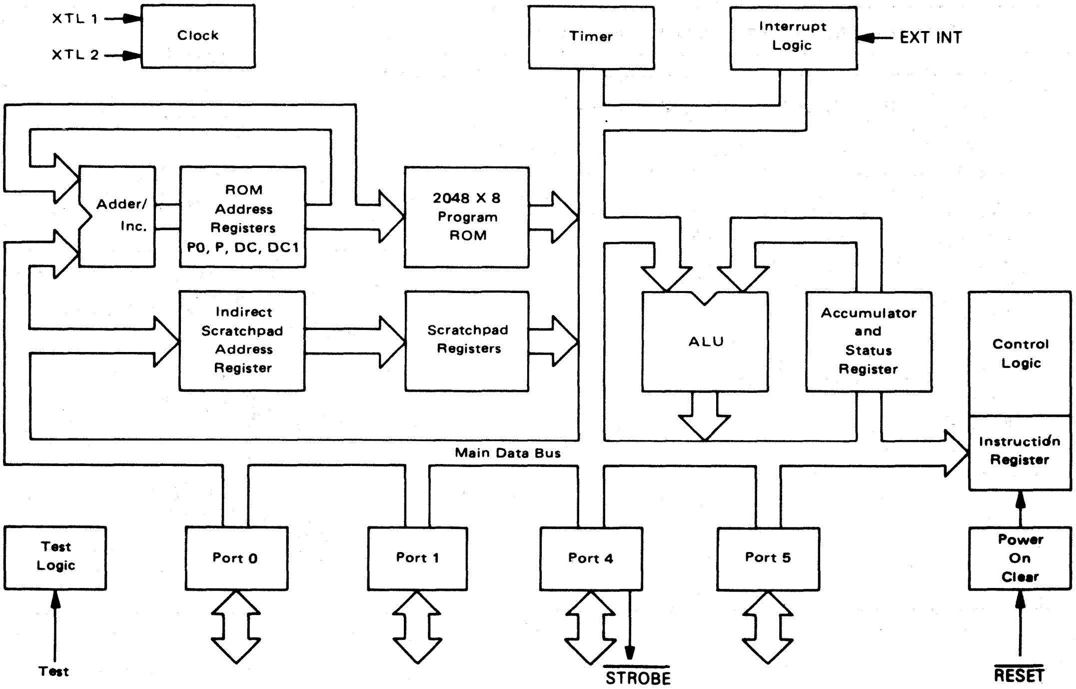
*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure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

FIGURE 1 — PIN ASSIGNMENT



ADI-445 R1

FIGURE 2 — BLOCK DIAGRAM



AC CHARACTERISTICS

Signal	Symbol	Parameter	0 to 70°C		−40° to +85°C		Unit	Notes
			Min	Max	Min	Max		
XTL1 XTL2	t_o	Time Base Period, all clock modes	250	1000	250	500	ns	
	$t_{ex}(H)$	External clock pulse width high	90	700	100	390	ns	
	$t_{ex}(L)$	External clock pulse width low	100	700	110	390	ns	
ϕ	t_ϕ	Internal ϕ clock	$2t_0$			$2t_0$		
WRITE	t_w	Internal WRITE Clock period	$4t_\phi$ $6t_\phi$			$4t_\phi$ $6t_\phi$		Short Cycle Long Cycle
I/O	$t_{dI/O}$	Output delay from internal WRITE clock	0	1000	0	1200	ns	50 pF plus one TTL load
	$t_{sI/O}$	Input setup time to internal WRITE clock	1000		1200		ns	
$\overline{\text{STROBE}}$	$t_{I/O\text{-}s}$	Output valid to $\overline{\text{STROBE}}$ delay	$3t_\phi - 1000$	$3t_\phi + 250$	$3t_\phi - 1200$	$3t_\phi + 300$	ns	I/O load = 50 pF + 1 TTL load
	t_{sL}	$\overline{\text{STROBE}}$ low time	$8t_\phi - 250$	$12t_\phi + 250$	$8t_\phi - 300$	$12t_\phi + 300$	ns	$\overline{\text{STROBE}}$ load = 50 pF + 3 TTL loads
$\overline{\text{RESET}}$	t_{RH}	$\overline{\text{RESET}}$ hold time, low	$6t_\phi + 750$		$6t_\phi + 1000$		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	$6t_\phi + 750$		$6t_\phi + 1000$		ns	To trigger interrupt
			$2t_\phi$		$2t_\phi$		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	$V_{IHEX} = V_{CC}$
I_{ILEX}	External Clock Input Low Current	–	−100	–	−130	μA	$V_{ILEX} = V_{SS}$
$V_{IHI/O}$	Input High Level, I/O Pins	2.0 2.0	V_{CC} 13.2	2.2 2.2	V_{CC} 13.2	V V	Standard Pullup Open Drain (1)
V_{IHR}	Input High Level, $\overline{\text{RESET}}$	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, and Inputs with No Pullup Resistor	– –	+10 −5	– –	+18 −8	μA	$V_{IN} = 13.2$ V $V_{IN} = 0.2$ 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 –	– −8.5	−1.3 –	– −11	mA	$V_{OH} = 1.5$ V $V_{OH} = 0.7$ V
I_{OHS}	STROBE Output High Current	−300	–	−270	–	μA	$V_{OL} = 2.4$ 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. $\overline{\text{RESET}}$ 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)	$-5t\pm$ 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 3 — $\overline{\text{STROBE}}$ SOURCE CAPABILITY (TYPICAL AT $V_{CC}=5$ V, $T_A=25°C$)

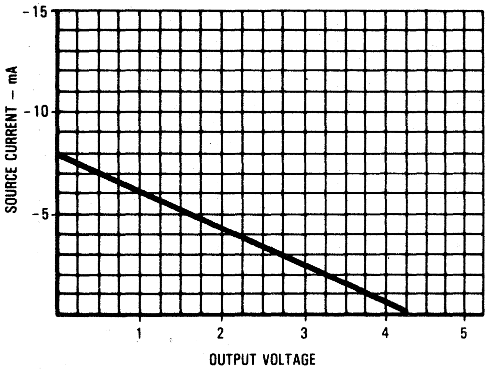


FIGURE 4 — $\overline{\text{STROBE}}$ SINK CAPABILITY (TYPICAL AT $V_{CC}=5$ V, $T_A=25°C$)

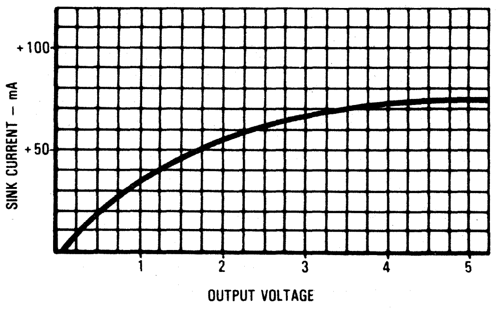


FIGURE 5 — STANDARD I/O PORT SOURCE CAPABILITY (TYPICAL AT $V_{CC}=5$ V, $T_A=25°C$)

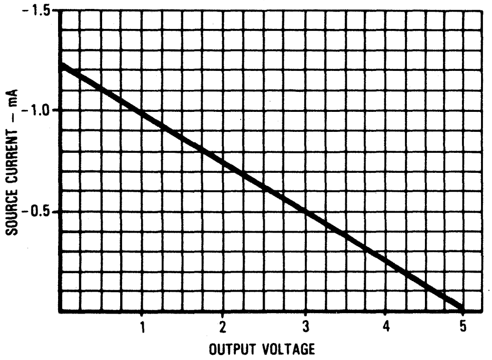


FIGURE 6 — DIRECT DRIVE I/O PORT SOURCE CAPABILITY (TYPICAL AT $V_{CC}=5$ V, $T_A=25°C$)

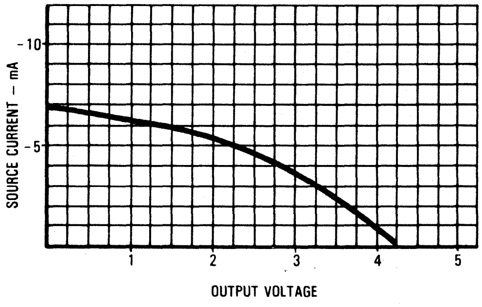


FIGURE 7 — I/O PORT SINK CAPABILITY (TYPICAL AT $V_{CC}=5$ V, $T_A=25°C$)

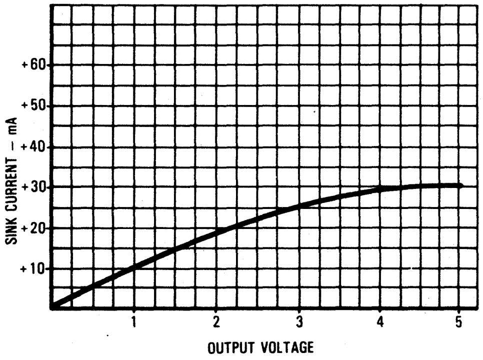


FIGURE 8 — MAXIMUM OPERATING TEMPERATURE vs I/O POWER DISSIPATION

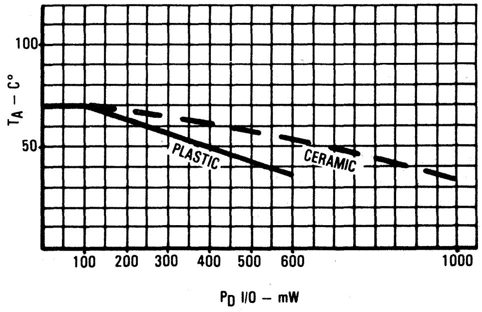


FIGURE 9 — MC3870 IDD vs TEMPERATURE (V_{CC} = 5 V)

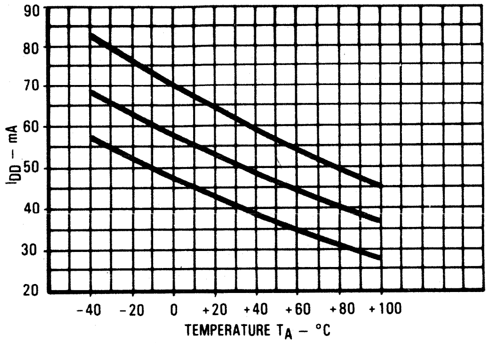
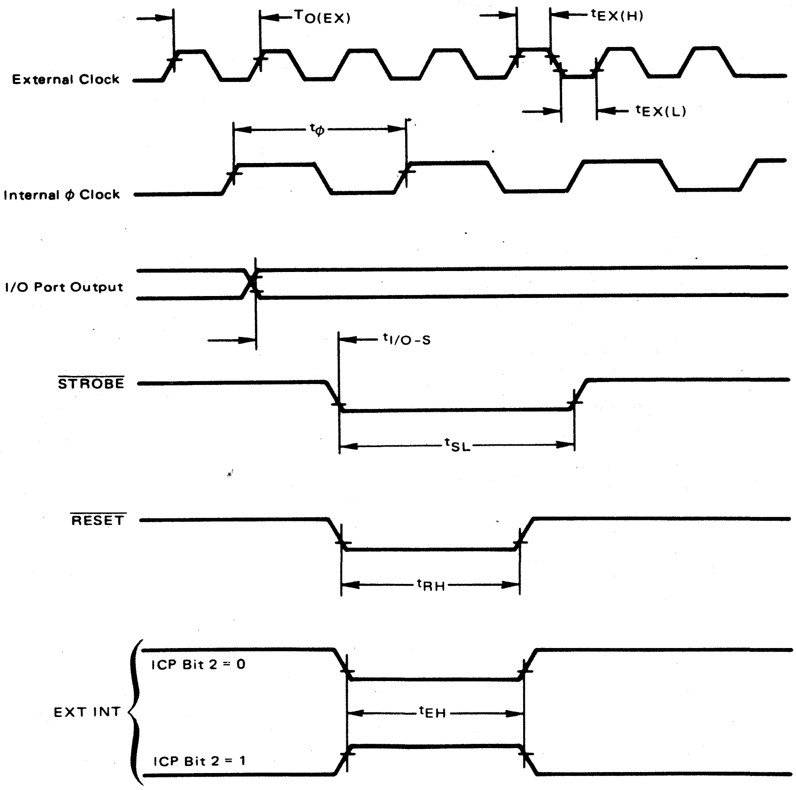
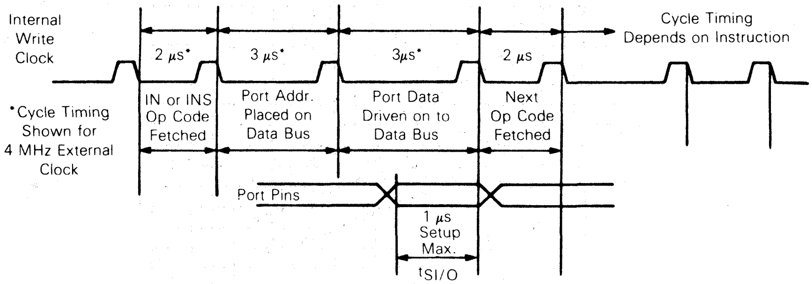


FIGURE 10 — AC TIMING DIAGRAM

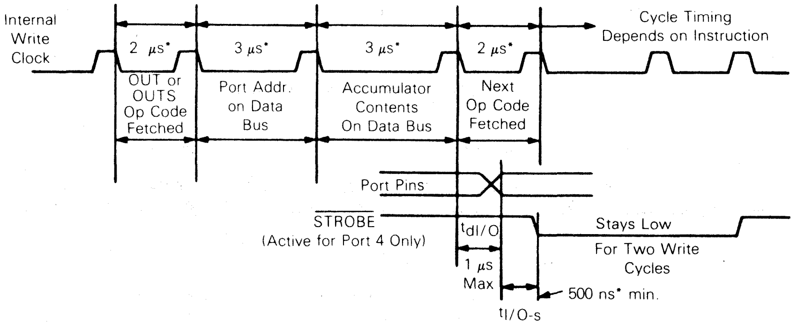


NOTE: All measurements are referenced to V_{IL} max., V_{IH} min., V_{OL} max., or V_{OH} min.

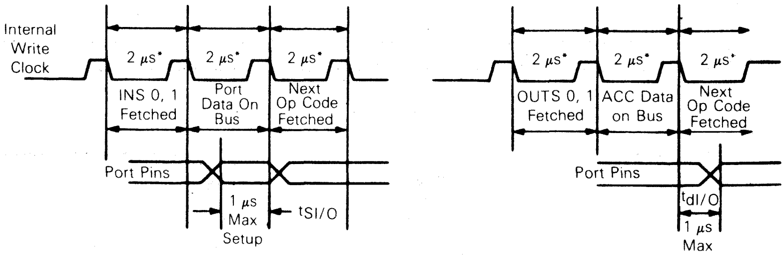
FIGURE 11 — INPUT/OUTPUT AC TIMING



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

3870 CLOCKS

The time base for the 3870 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 nanoseconds 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 reproducability 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

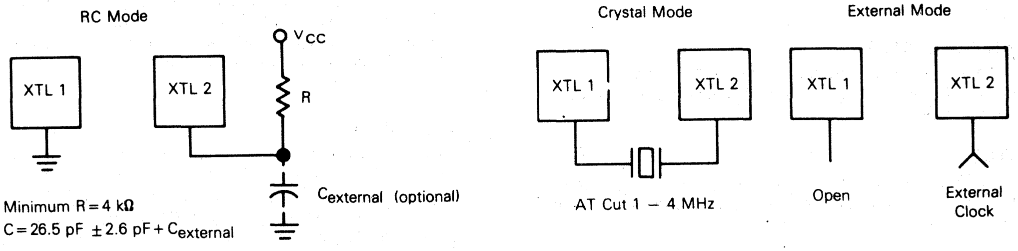
b) Frequency Tolerance measured with 18 pF load (0.1% accuracy) – drive level 10 mW

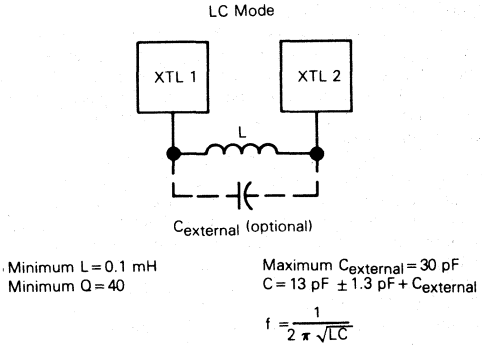
c) Shunt capacitance (Co) = 7 pF max

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

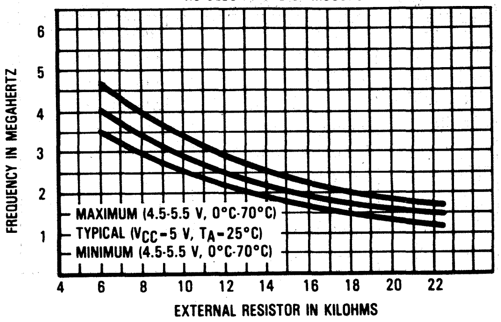
FIGURE 12 — CLOCK CONFIGURATION





$$f = \frac{1}{2\pi\sqrt{LC}}$$

RC CLOCK MODE OF MC3870



FUNCTIONAL PIN DESCRIPTION

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

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

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

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 15 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 ($\overline{\text{P4-0}}$ - $\overline{\text{P4-7}}$) 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
$\overline{\text{P0-0}}$ - $\overline{\text{P0-7}}$	I/O Port 0	Bidirectional
$\overline{\text{P1-0}}$ - $\overline{\text{P1-7}}$	I/O Port 1	Bidirectional
$\overline{\text{P4-0}}$ - $\overline{\text{P4-7}}$	I/O Port 4	Bidirectional
$\overline{\text{P5-0}}$ - $\overline{\text{P5-7}}$	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

3870 ARCHITECTURE

This section describes the basic functional elements of the 3870 as shown in the block diagram of Figure 2. A programming model is shown in Figure 13.

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 (0'28') 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).

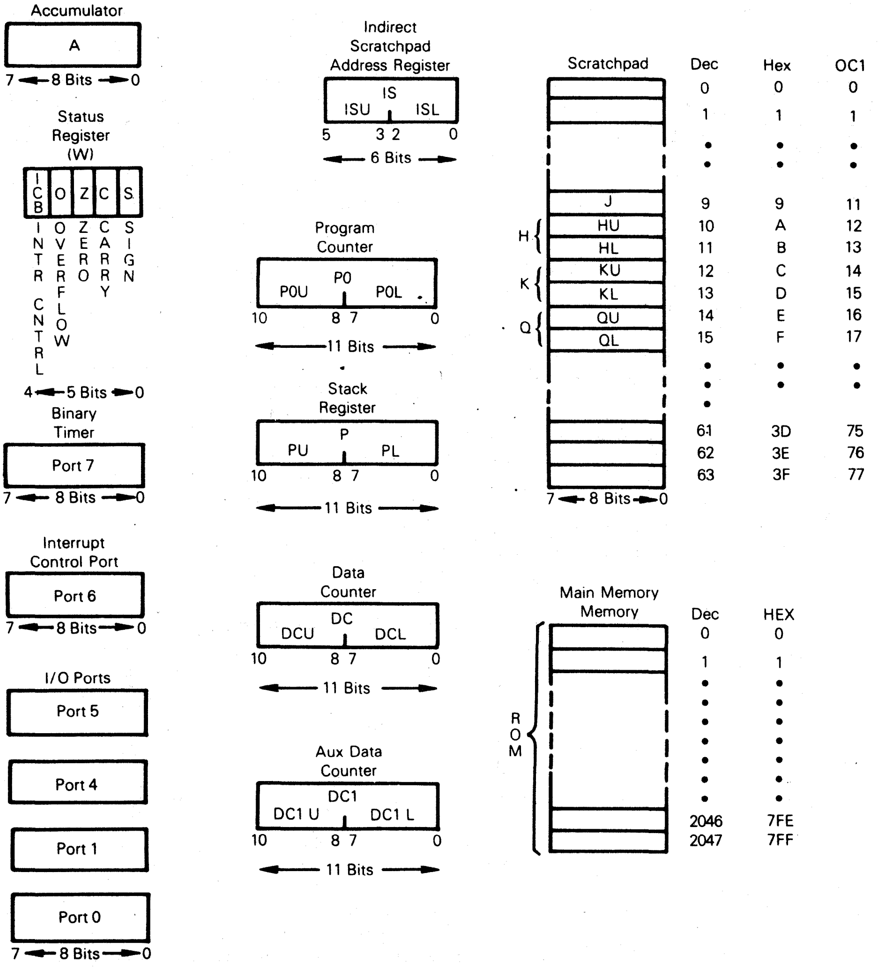
ARITHMETIC AND LOGIC UNIT (ALU)

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.

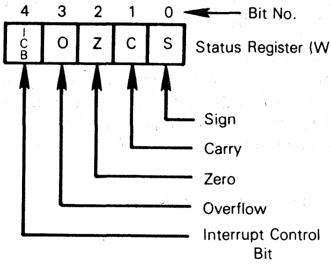
FIGURE 13 — MC3870 PROGRAMMABLE REGISTERS, PORTS, AND MEMORY MAP



THE STATUS REGISTER (W)

The Status Register (also called the W register) holds five status flags as shown in Figure 14.

FIGURE 14 — STATUS REGISTER (W)



Summary of Status Bits

OVERFLOW = $\text{Carry}_7 \oplus \text{CARRY}_6$

ZERO = $\overline{\text{ALU}_7} \land \overline{\text{ALU}_6} \land \overline{\text{ALU}_5} \land \overline{\text{ALU}_4} \land \overline{\text{ALU}_3} \land \overline{\text{ALU}_2} \land \overline{\text{ALU}_1} \land \overline{\text{ALU}_0}$

CARRY = CARRY_7

SIGN = $\overline{\text{ALU}_7}$

INTERRUPT CONTROL BIT (ICB)

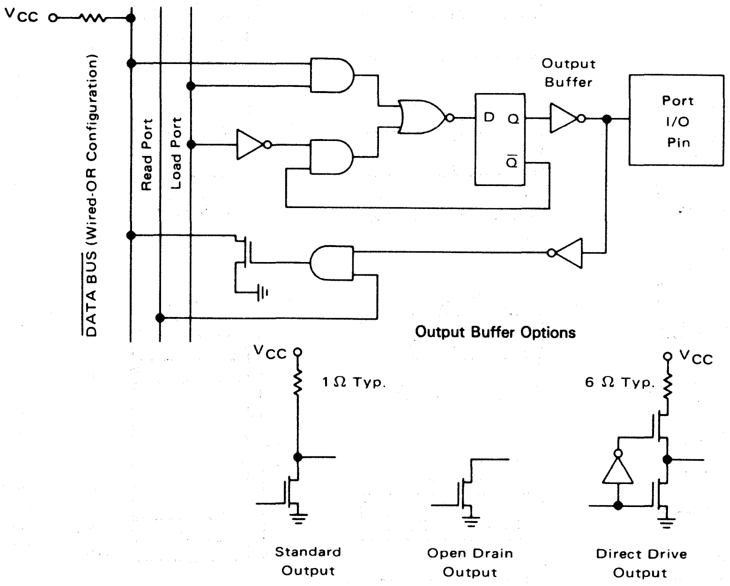
The ICB may be used to allow or disallow interrupts in the MC3870. This bit is not the same as the two interrupt enable bits in the Interrupt Control Port (ICP). If the ICB is set and the MC3870 interrupt logic communicates an interrupt request to the CPU section, the interrupt will be acknowledged and 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 15.

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. $\overline{\text{STROBE}}$ 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 15 — 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 $\overline{\text{STROBE}}$ output is always configured similar to a Direct Drive Output except that it is capable of driving 3 TTL loads.

$\overline{\text{RESET}}$ 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 16, 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 17.

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:

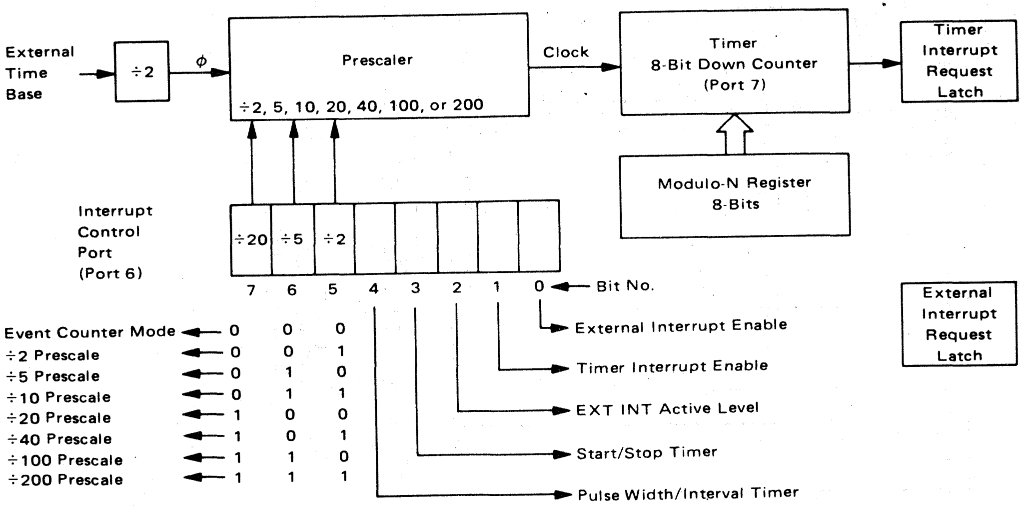
Bit 0 — External Interrupt Enable
Bit 1 — Timer Interrupt Enable
Bit 2 — EXT INT Active Level
Bit 3 — Start/Stop Timer
Bit 4 — Pulse Width/Interval Timer
Bit 5 — ÷2 Prescale
Bit 6 — ÷5 Prescale
Bit 7 — ÷20 Prescale

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 Internal 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.

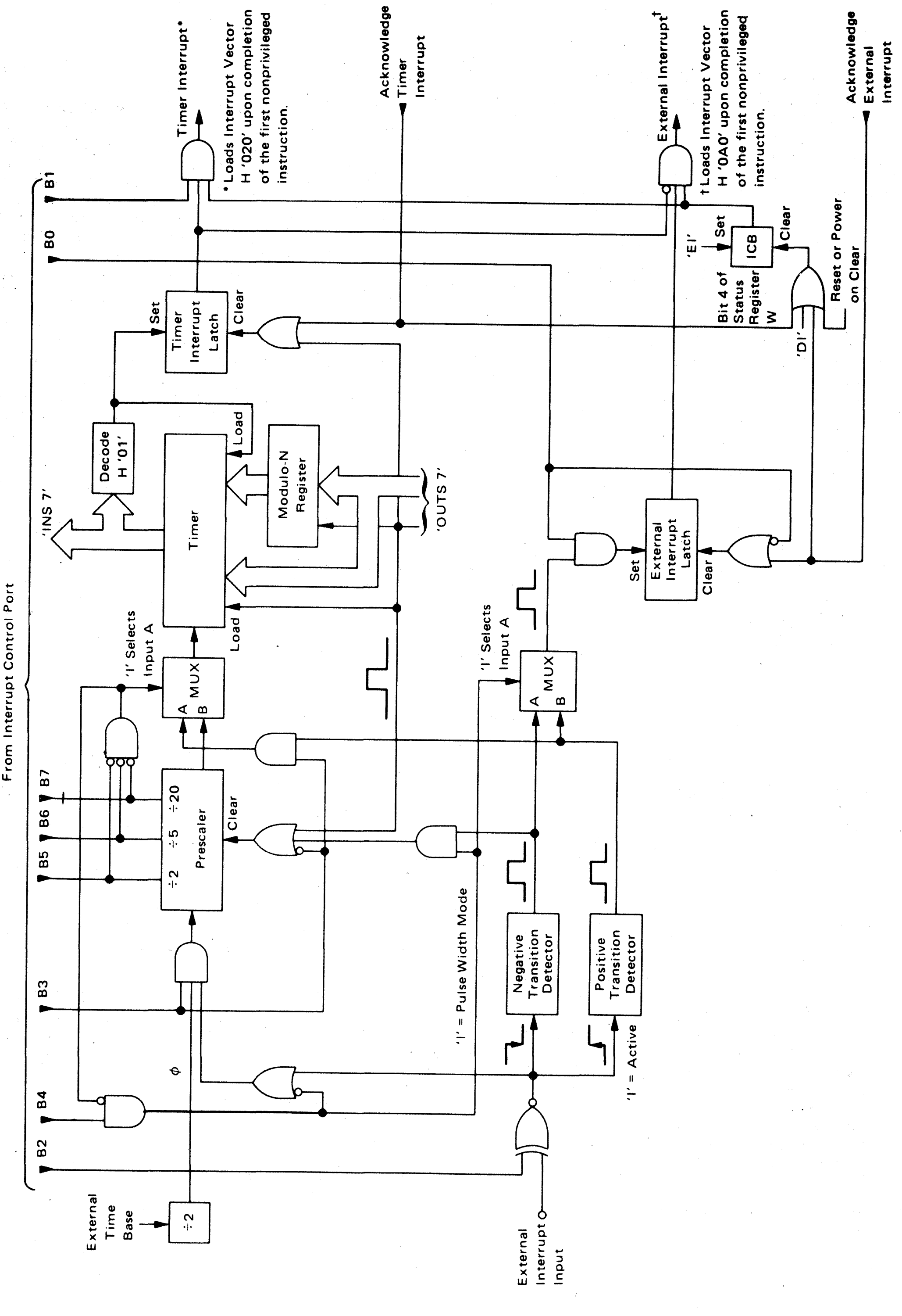
Any of three conditions will cause the prescaler to be reset: whenever the timer is stopped by clearing ICP bit 3, execution of an output instruction to Port 7, (the timer is assigned port address 7), or on the trailing edge transition of the EXT INT pin when in the Pulse Width Measurement Mode. These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

FIGURE 16 — TIMER AND CONTROL PORT BLOCK DIAGRAM



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

FIGURE 17 — 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 ÷40 the timer interrupt request latch will be set every 400 ϕ clock periods. For a 2 MHz ϕ clock (4 MHz time-base freuqency) 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 ±6 ϕ 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 ±6 ϕ 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.

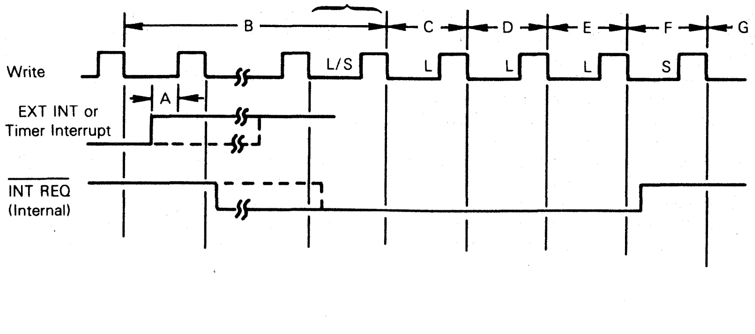
Figure 18 details the interrupt sequence which occurs whether the interrupt request is from an external source via EXT INT or from the MC3870's internal timer. Events are labeled with the letters A through G and are described below.

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 18 — 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 18, 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).

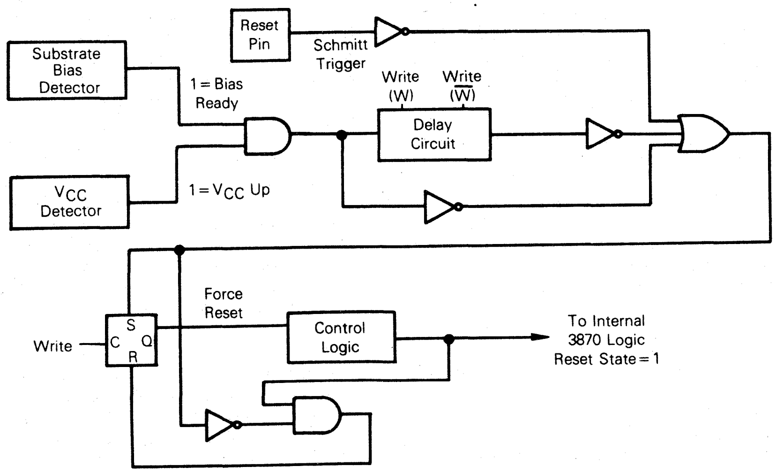
POWER-ON RESET

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 19 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 19 will stay low until the clocks

FIGURE 19 — 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, $\overline{\text{STROBE}}$ 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

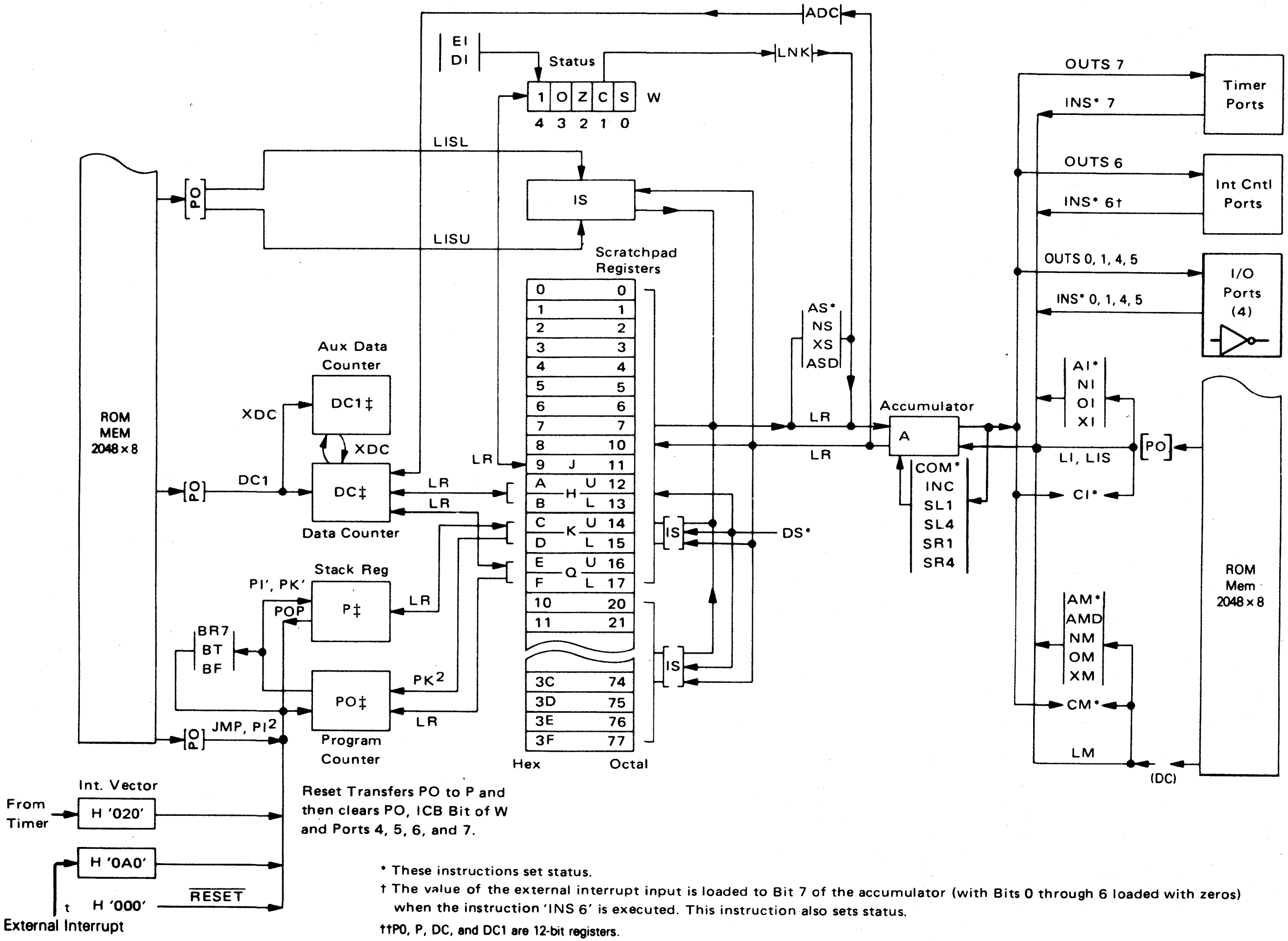
The nomenclature is used in order to be consistent with the assembly language mnemonics.

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 20 for a MC3870 Programming Model.

FIGURE 20 — 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.

††PO, 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				Status Bits		
						Short	Long	(2 MHzϕ)	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)ΛH'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)vH'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				Status Bits		
						Short	Long	(2 MHz ϕ)	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 TEST CONDITION 2^2 2 2^0 ZERO CRY SIGN	taa	P0←(P0)+1+'Haa' if any test is true	8taa	2	2	1	7	–	–	–	–
Branch if Negative	8M	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 TEST CONDITION 2' 2' 2' 2' OVF ZERO CRY SIGN	taa	P0←(P0)+1+H'aa' if all false test bits	9taa	2	2	1	7	–	–	–	–
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	PO←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				Status Bits		
						Short	Long	(2 MHz φ)	OVR	ZERO	CRY	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)∧[(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				Status Bits		
						Short	Long	(2 MHz φ)	OVR	ZERO	CRY	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)	0F	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)	0D	1	1	2	8	–	–	–	–
Load Stack Register	LR	P, K	PU ← (r12), PL←r13)	09	1	1	2	8	–	–	–	–
Return from Subroutine*	POP		P0⇄(P)	1C	1	2		4	–	–	–	–
Store Data Counter	LR	Q, DC	r14←(DCU), r15←(DCL)	0E	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				Status Bits		
						Short	Long	(2 MHz φ)	OVR	ZERO	CRY	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)∧(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			Status Bits			
						Short	Long	(2 MHz φ)	OVR	ZERO	CRY	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(0bbb)**	1	1		2	–	–	–	–
Load ISAR Upper	LISU	bbb	ISU←bbb	6(1bbb)**	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

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

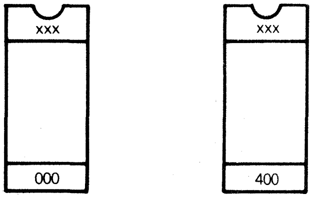
MDOS disk file

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.

® Registered trademark of Motorola Inc.

EXORsets is a trademark of Motorola Inc.

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.

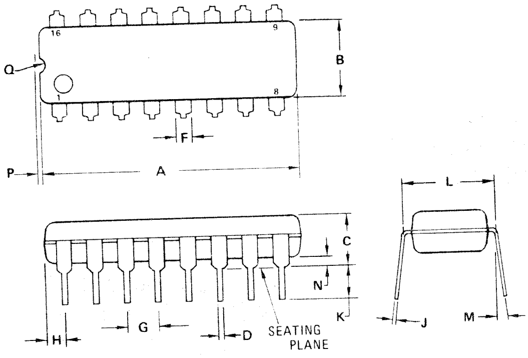
Mechanical Data

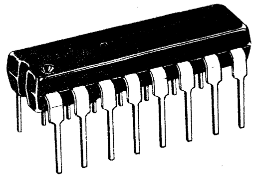
Mechanical Data

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.

16-PIN PACKAGES

P SUFFIX
PLASTIC PACKAGE
CASE 648-03





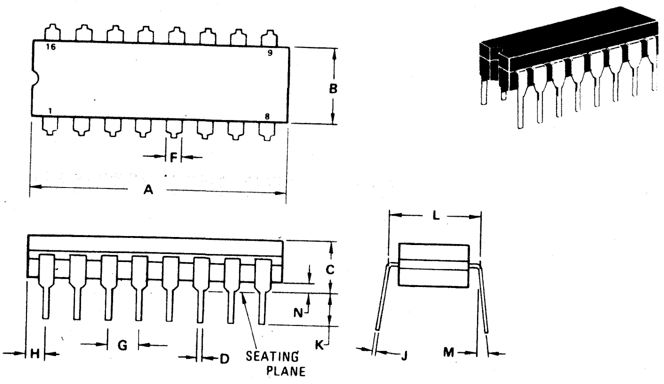
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	20.70	21.34	0.815	0.840
B	6.10	6.60	0.240	0.260
C	4.06	4.57	0.160	0.180
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.32	1.83	0.052	0.072
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.37	7.87	0.290	0.310
M	–	10°	–	10°
N	0.51	1.02	0.020	0.040
P	0.13	0.38	0.005	0.015
Q	0.51	0.76	0.020	0.030

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

16-PIN PACKAGES (CONTINUED)

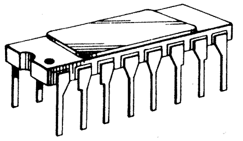
L SUFFIX
CERAMIC PACKAGE
CASE 620-02

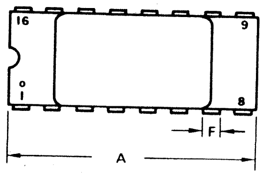


NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

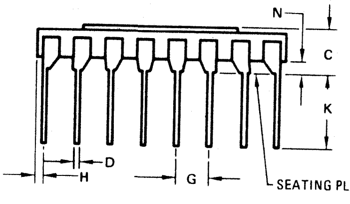
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	19.05	19.81	0.750	0.780
B	6.22	6.98	0.245	0.275
C	4.06	5.08	0.160	0.200
D	0.38	0.51	0.015	0.020
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.14	0.020	0.045
J	0.20	0.31	0.008	0.012
K	3.18	0.30	0.125	0.160
L	7.37	7.87	0.290	0.310
M	–	15°	–	15°
N	0.51	1.02	0.020	0.040

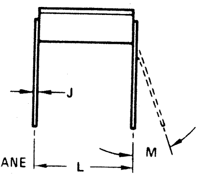
L SUFFIX
CERAMIC PACKAGE
CASE 690





NOTE:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.

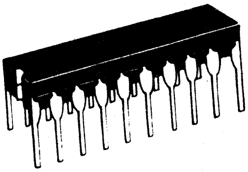




DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	20.07	20.57	0.790	0.810
C	2.67	3.94	0.105	0.155
D	0.38	0.53	0.015	0.021
F	1.22	1.52	0.048	0.060
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.31	0.008	0.012
K	3.05	4.83	0.120	0.190
L	7.62 BSC		0.300 BSC	
M		10°		10°
N	0.64	1.52	0.025	0.060

20-PIN PACKAGES

L SUFFIX
CASE 732-03

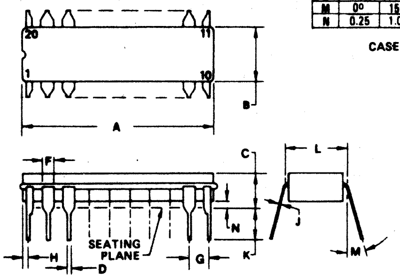


NOTES:

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.

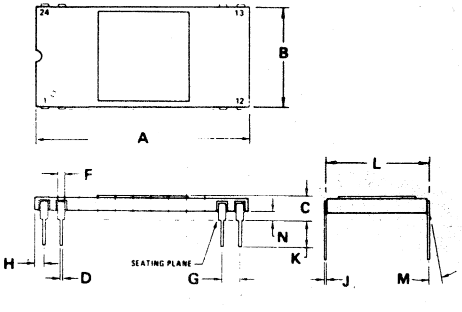
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	23.88	25.15	0.940	0.990
B	6.60	7.49	0.260	0.295
C	3.81	5.08	0.150	0.200
D	0.38	0.56	0.015	0.022
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.27	0.020	0.050
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.25	1.02	0.010	0.040

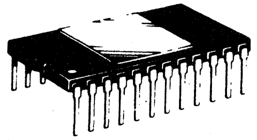
CASE 732-03



24-PIN PACKAGES

CERAMIC PACKAGE
CASE 716



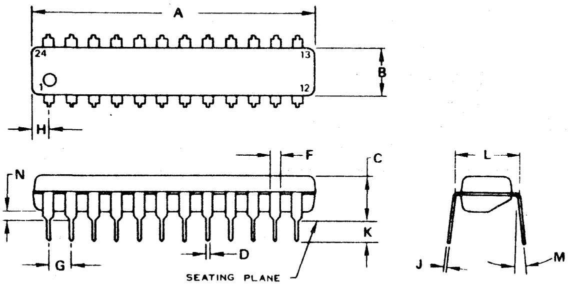


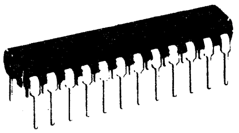
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	27.64	30.99	1.088	1.220
B	14.73	15.34	0.580	0.604
C	2.67	4.32	0.105	0.170
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.57	0.100	0.180
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	1.02	1.52	0.040	0.060

NOTE:

1. LEADS TRUE POSITIONED WITHIN 0.25mm (0.010) DIA (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

P3 SUFFIX
PLASTIC PACKAGE
CASE 724-02





θ_{JA}(typ) = 90°C/W

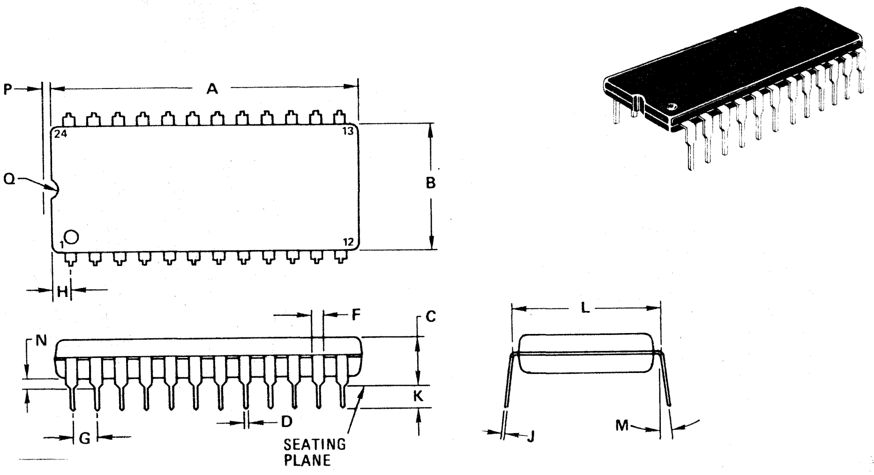
NOTE:

1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION (DIM D).

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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°	–	10°
N	0.51	1.02	0.020	0.040

24-PIN PACKAGES (CONTINUED)

P SUFFIX
PLASTIC PACKAGE
CASE 649-03

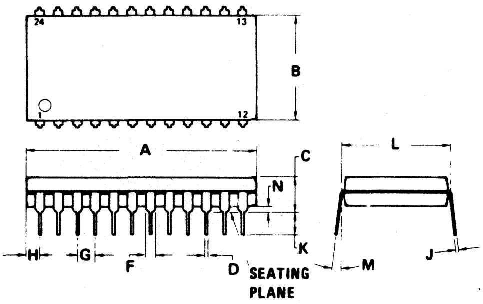


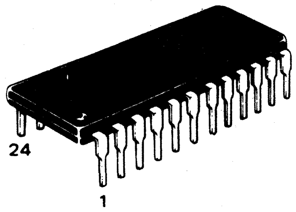
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	31.50	32.13	1.240	1.265
B	13.21	13.72	0.520	0.540
C	4.70	5.21	0.185	0.205
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	0.51	1.02	0.020	0.040
P	0.13	0.38	0.005	0.015
Q	0.51	0.76	0.020	0.030

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.

24-PIN PACKAGES (CONTINUED)

PLASTIC PACKAGE
CASE 709

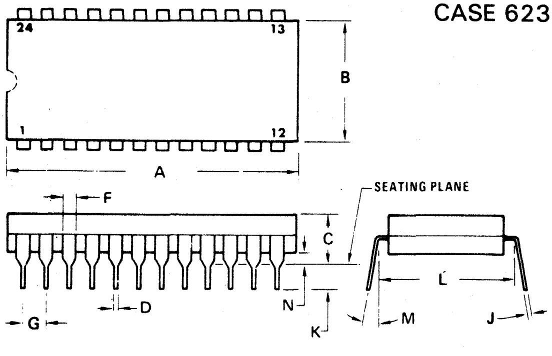


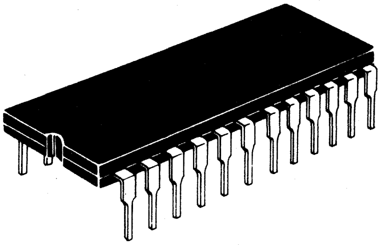


NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	31.37	32.13	1.235	1.265
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	1.78	2.03	0.070	0.080
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

CERDIP PACKAGE
CASE 623



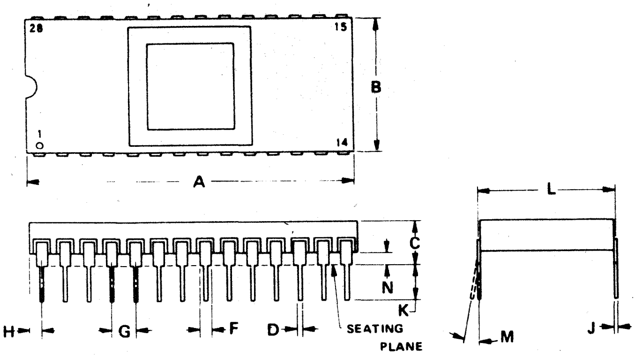


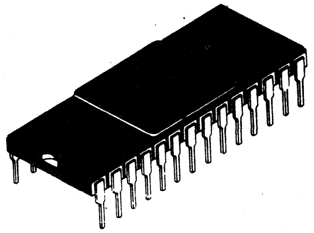
NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL)

DIM	MILLIMETERS		INCHES	
	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	2.29	4.06	0.090	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

28-PIN PACKAGES

CERAMIC PACKAGE
CASE 719

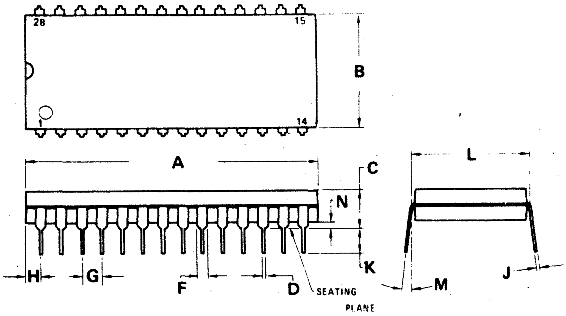


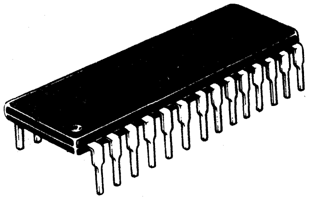


NOTES:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIAMETER (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

	MILLIMETERS		INCHES	
DIM	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

PLASTIC PACKAGE
CASE 710



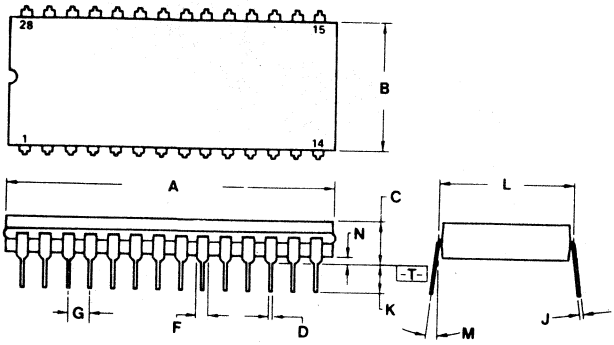


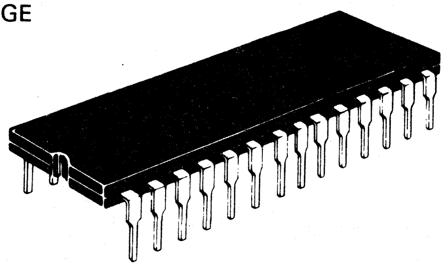
NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

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

28-PIN PACKAGES (CONTINUED)

CERDIP PACKAGE
CASE 733



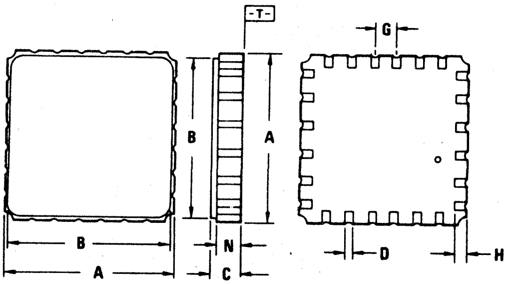


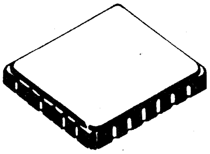
NOTES:
1. DIM [-A-] IS DATUM.
2. POSITIONAL TOL FOR LEADS:

   [⊕ | 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.

	MILLIMETERS		INCHES	
DIM	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	2.54	4.06	0.100	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

CHIP CARRIER
CASE 753





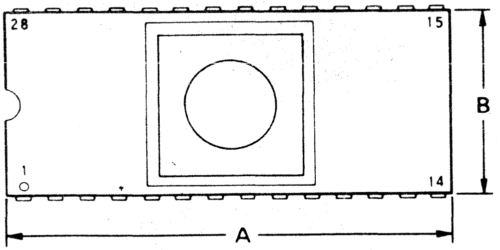
NOTES:
1. DIMENSION A IS DATUM. (2 PLACES)
2. [-T-] IS GUAGE PLANE.
3. POSITIONAL TOLERANCE FOR TERMINALS (D): 24 PLACES

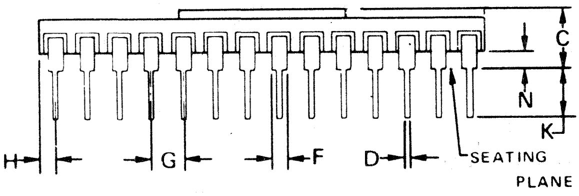
   [⊕ | 0.25 (0.010) Ⓜ | T | A Ⓢ]
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.91	10.41	0.390	0.410
B	9.78	9.90	0.385	0.390
C	1.63	1.93	0.064	0.076
D	0.39	0.63	0.015	0.025
G	1.27 BSC		0.050 BSC	
H	0.77	1.01	0.030	0.040
N	1.40	1.65	0.055	0.065

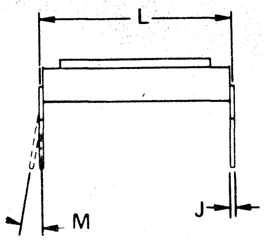
40-PIN PACKAGES

CERAMIC PACKAGE
CASE 715-04





	MILLIMETERS		INCHES	
DIM	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

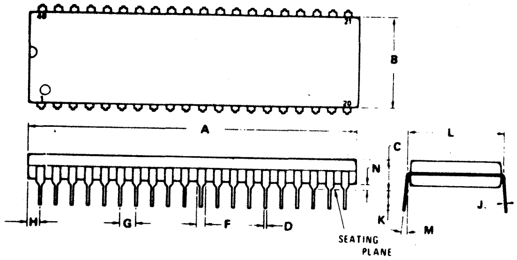


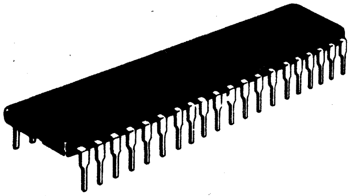
NOTES:

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.

40-PIN PACKAGES (CONTINUED)

PLASTIC PACKAGE
CASE 711
CASE 711-01

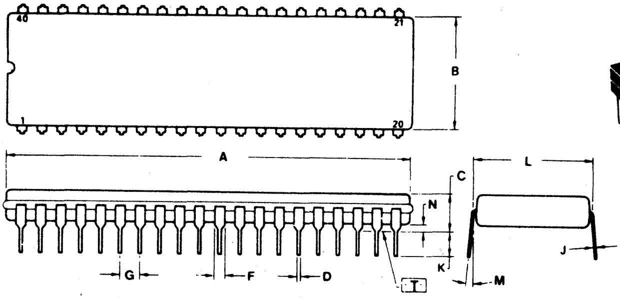


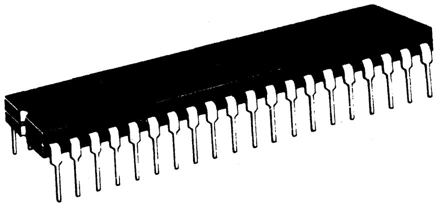


NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

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

L SUFFIX
CERDIP PACKAGE
CASE 734





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

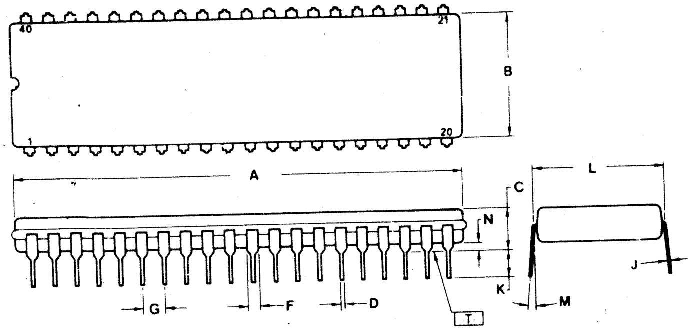
⌖ | Ø 0.25(0.010) Ⓜ | T | A Ⓜ

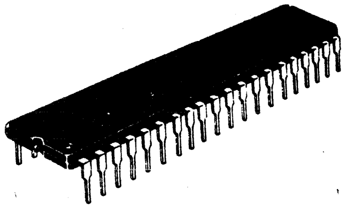
3. -T- IS SEATING PLANE.
4. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONS A AND B INCLUDE MENISCUS.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	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°	15°	5°	15°
N	0.51	1.27	0.020	0.050

40-PIN PACKAGES (CONTINUED)

S SUFFIX
CERDIP PACKAGE
CASE 734-03

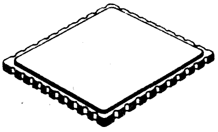




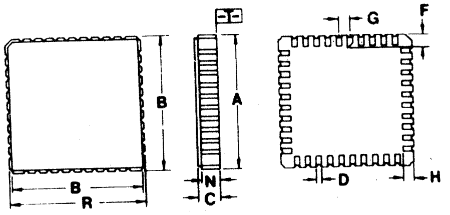
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	51.31	53.24	2.020	2.096
B	12.70	15.49	0.500	0.610
C	4.06	5.84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

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. DIMENSION A AND B INCLUDES MENISCUS.

Z SUFFIX
40-PIN
CHIP CARRIER
CASE 761



DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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.

Technical Training

Technical Training System Design

Since 1974 when Motorola first introduced the M6800 Family course, Motorola technical training courses have been among the most popular and affective 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 Europe. The schedule is advertised periodically, and information is always available from the regional Motorola Sales Offices.

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 country, or enrollment procedures, please call the Motorola Technical Training group at your nearest Motorola Sales Office.

COURSE OFFERINGS

Basic MC6800 Course – 4 Days (MTT1):

Course covers the MC6800 Microprocessor, Instruction Set, RAMs, ROMs, Addressing Modes, Assembler, PIA and ACIA, and examples programs for the MC6800.

Basic M6801 Familiy – 4 Days (MTT2):

Course covers the MC6801 Microcomputer, Instruction Set, RAMs, ROMs, Addressing Modes, Assembler and Input/Output Techniques, and examples programs for the MC6801.

MC6809 Update – 2 Days (MTT3):

Course covers MC6809 Microprocessor, Instruction Set, Addressing Modes, Relocatable Macro Assembler, and example programs for the MC6809.

MC6801 Update – 2 Days (MTT5):

This course is an update course for students who are familiar with the MC6800 and need to be knowledgeable of the MC6801 Microcomputer.

M6805 Family Basic Course – 3 Days (MTT6):

This course covers the M6805 Family, the Instruction Set, RAMs, ROMs, Addressing Modes, Assembler, and example programs for the M6805 Family.

Understanding Microprocessor Basics – 1 Day (MTT7):

This course is a one-day non-technical course designed to acquaint Managers, Secretaries, Buyers, Salesman, 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.

Micromodules – 2 Days (MTT9):

This 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 – 3 Days (MTT10):

This course is designed to prepare the student to understand and use the basic functions of the MC6800 and MC6809 development systems. Topics covered are EXORterm, EXORciser II, EXORdisk, System Analyser, Usec, PROM Programmer, CRT Editor, Macro Assembler, Linking Loader, and Application Examples.

Basic MC6809 Course – 4 Days (MTT11):

This 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 stadard Pascal and structured programming as taught in a college-level course. The fourth day includes Motorola extensions and implementations for the MC6809. Each student has the opportunity to complete and execute several programs. Each receives, and keeps, a home lab diskette with sample exercises.

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.

Also

MACROCELL Array Course – 3 Days:

Covering logic simulation, fault grading, auto place and route, graphics, design rule violation check, AC simulation, hands-on activity including simple design.

Application Notes

Application Notes – Reference Index

The following application notes are available for free at your nearest Motorola distributor or sales office.

A collection of application notes for 8/16-bits microcomponent designs is also available under the reference "MCU/MPU Applications Manual".

AN-747	Low-Speed Modem System Design Using the M6800
AN-754	Device Operation and System Implementation of the Asynchronous Communications Interface Adapter (MC6850)
AN-764	A Floppy Disk Controller Using the MC6852 SSDA and other M6800 Microprocessor Family Parts
AN-775	M6800 Systems Utilizing the MC6875 Clock Generator/Driver
AN-787	An M6800 Clock System that Handles DMA and Memory Refresh Cycle Stealing
AN-788	MC6802/MC6846 with Mikbuy 2.0 A Powerful Software Development Team
AN-797	MC6801/03 Port Expansion
AN-798A	A Low-cost Terminal Using the MC6801
AN-799A	Application Prototype Board (APB) for M6801/MC6803/MC68701 MCU's
AN-807	Special Considerations in Using the MC6801 Interrupt Capabilities
AN-808	Interfacing M6800 Peripheral Devices to the M68000 Asynchronously
AN-815	Color Graphics for the MC68000 Using the MC6847
AN-820	Hardware Considerations for Direct Memory Access Using the MC6809 Microprocessor Unit and MC68488 DMA Controller
AN-822	The Video Display Generator as a Television Interface
AN-823	CBUG05 Debug Monitor Program for MC146805E2 Microprocessor Unit
AN-825	An Interactive Graphic System Using the MC6809
AN-832	Letting the MC68701 Program Itself
AN-833	Software Refresh Memory Board for an MC6809 System
AN-836	Using Low-cost 1MHz Peripherals in a 2MHz System with the MC68B09 and the MC68B093
AN-851	Motorola MC6845 CRTC Simplifiers Video Display Controllers

APPLICATION NOTES (continued)

AN-852	Monitor for the MC146805G2L1 Microcomputer
AN-853	M146805 CMOS Familiy Emulators
AN-853	Errata
AN-855	Versatile Thermostat Using CMOS MC146805E2 MPU
AN-856A	Low Power Considerations for MC146818 Real Time Clock Applications
AN-857	MC68705P3/R3/U3 8-Bit EPROM Microcomputer Programming Module
AN-858	Bicycle Computer Using the MC146805G2 Microcomputer
AN-859	Memory Management Techniques Using the MC6829
AN-863	Keyless Entry System Using an MC146805F2 8-Bit Microcomputer Unit
AN-865	The MC6809/MC6809E Sync. Instruction
AN-866	Vectoring by Device Using Interrupt Sync. Acknowledge with the MC6809/MC6809E
AN-868	Using High-Speed CMOS Logic for Microprocessor Interfacing
AN-869	Application Summary for the MC6805R2 Single-Chip Microcomputer with A/D Converter