DL133

MOTOROLA INC.

8-BIT MICROPROCESSOR & PERIPHERAL DATA

MOTOROLA MICROPROCESSORS

Prepared by
Technical Information Center

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

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

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

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

Series C

Printed in U.S.A.

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

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

TABLE OF CONTENTS

MASTER INDEX

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

Motorola's Microprocessor Families

1

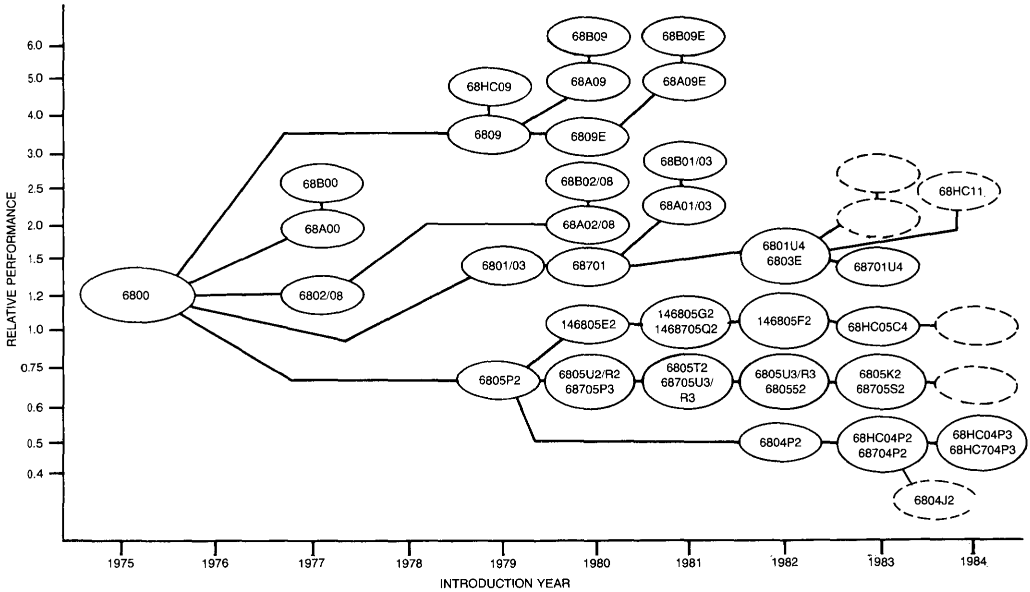
MOTOROLA'S MICROPROCESSOR AND MICROCOMPUTER FAMILIES

1

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

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

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



8-BIT MICROPROCESSORS (MPUs) MC6800 — MC6802 — MC6803 — MC6808 — 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:

The MC6802 MPU has all the attributes of the basic MC6800, but it reduces the component count of a minimum microcomputer system to only two.

The MC6802 adds an on-chip clock oscillator and 128 bytes of RAM to the capability of an MC6800. Data in the first 32 bytes of the built-in RAM can be retained in a lower-power mode by an external power source, allowing memory retention during a power-down situation.

Using this microprocessor, a minimum microcomputer system consists of:

1 — MC6802 Microprocessing Unit
1 — MC6846 ROM-I/O-Timer Unit

Of course, the system is expandable to any requirement with the adapters, expanders, and other peripheral chips that are a part of the M6800 Family.

The MC6803 MPU is the microprocessor version of the MC6801 single-chip microcomputer. The MC6803 accommodates 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 multigeneration 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 MC6803E was designed for uses in which the internal clock needs to be synchronized with system, peripherals, or other MPUs. The MC6803E also supports DMA and dynamic RAM refresh with its halt ($\overline{\text{HALT}}$) and bus available (BA) pins. Other features include:

1

- 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 x 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
- 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

The MC6808 low-cost version of the MC6802 microprocessor has an on-chip clock oscillator and driver, but no on-chip memory. The MC6808 can use up to 64K of external RAM, ROM, or peripherals.

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, and the MC6855 Serial DMA Processor) allow an MC6809 user to realize the full potential of the processor.

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.

1

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 ROMless 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 MC146804E2 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, applicance 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)

The MC146805E3 is an expanded version of the MC146805E2 that includes a 64K memory addressing capability.

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
MC6803NR	HMOS	40	—	13	Serial	82	64K	8	Yes	16-Bit
MC6803E	HMOS	40	128	13	Serial	82	64K	8	No	16-Bit
MC6808	HMOS	40	—	—	—	72	64K	8	Yes	—
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 +
MC146805E3	CMOS	40	112	16	—	61	64K	8	Yes	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.

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

PERIPHERAL AND INTERFACE COMPONENTS SELECTOR GUIDE (CONTINUED)

Reliability

2

RELIABILITY AND QUALITY MONITOR REPORT

OCTOBER 1983

2

Introduction

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

Dynamic Life

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

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

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

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

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

TABLE 1.
SUMMARY OF DYNAMIC LIFE TEST RESULTS

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

*90% Confidence Level

2

TABLE 2.
MICROPROCESSOR FAMILY DYNAMIC LIFE TEST RESULTS

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

*90% Confidence Level

SUMMARY:

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

Plastic Package Environmental Performance

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

Moisture Related Performance

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

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

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

Autoclave

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

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

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

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

Temperature Humidity Bias

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

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

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

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

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

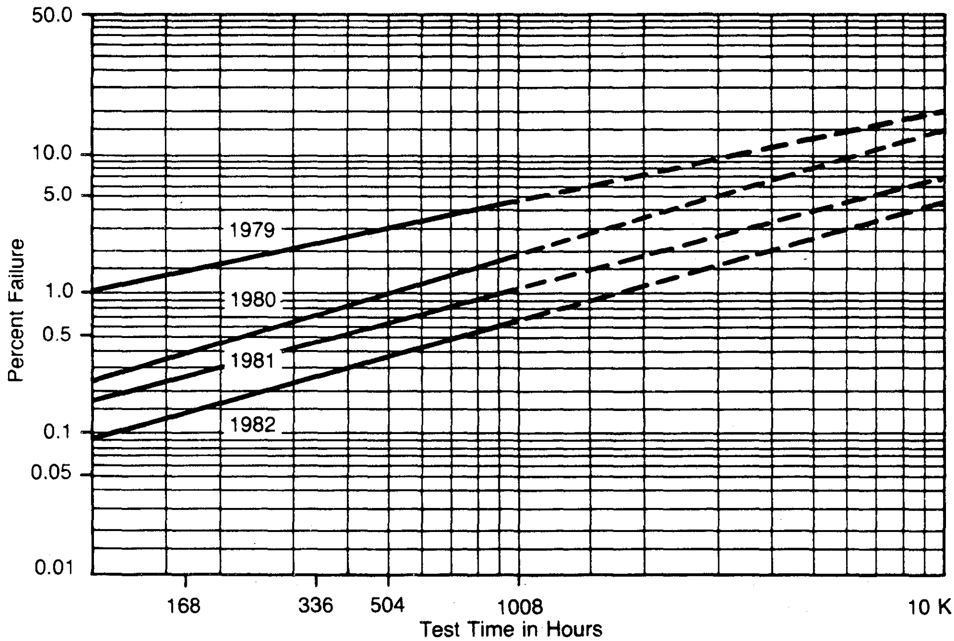


FIGURE 1. WEIBULL PLOT OF TEMPERATURE HUMIDITY BIAS TEST RESULTS

Thermal Cycling Performance

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

Temperature Cycle

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

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

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

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

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

Thermal Shock

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

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

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

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

2

Conclusions

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

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

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

APPENDIX A.
QUALITY AND RELIABILITY SYSTEM

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

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

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

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

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

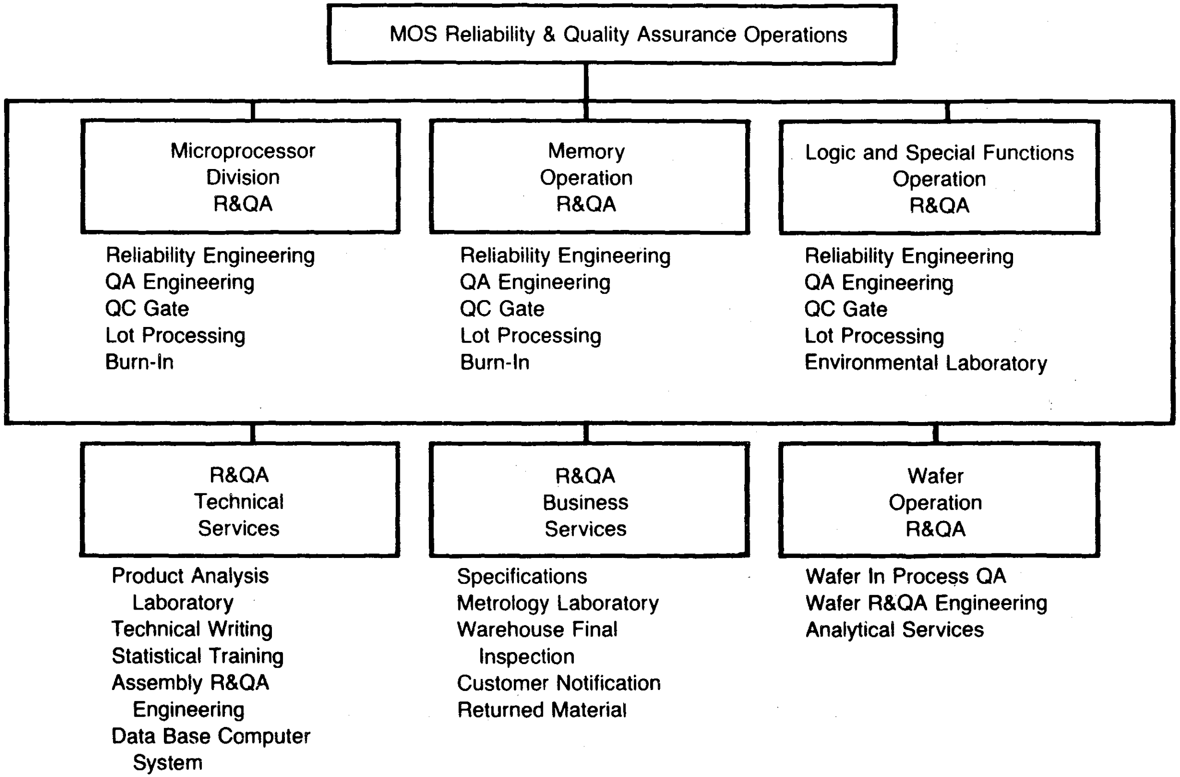


FIGURE A. RELIABILITY AND QUALITY ASSURANCE ORGANIZATION

APPENDIX B.
PACKAGING SYSTEMS

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

Plastic

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

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

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

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

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

Ceramic

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

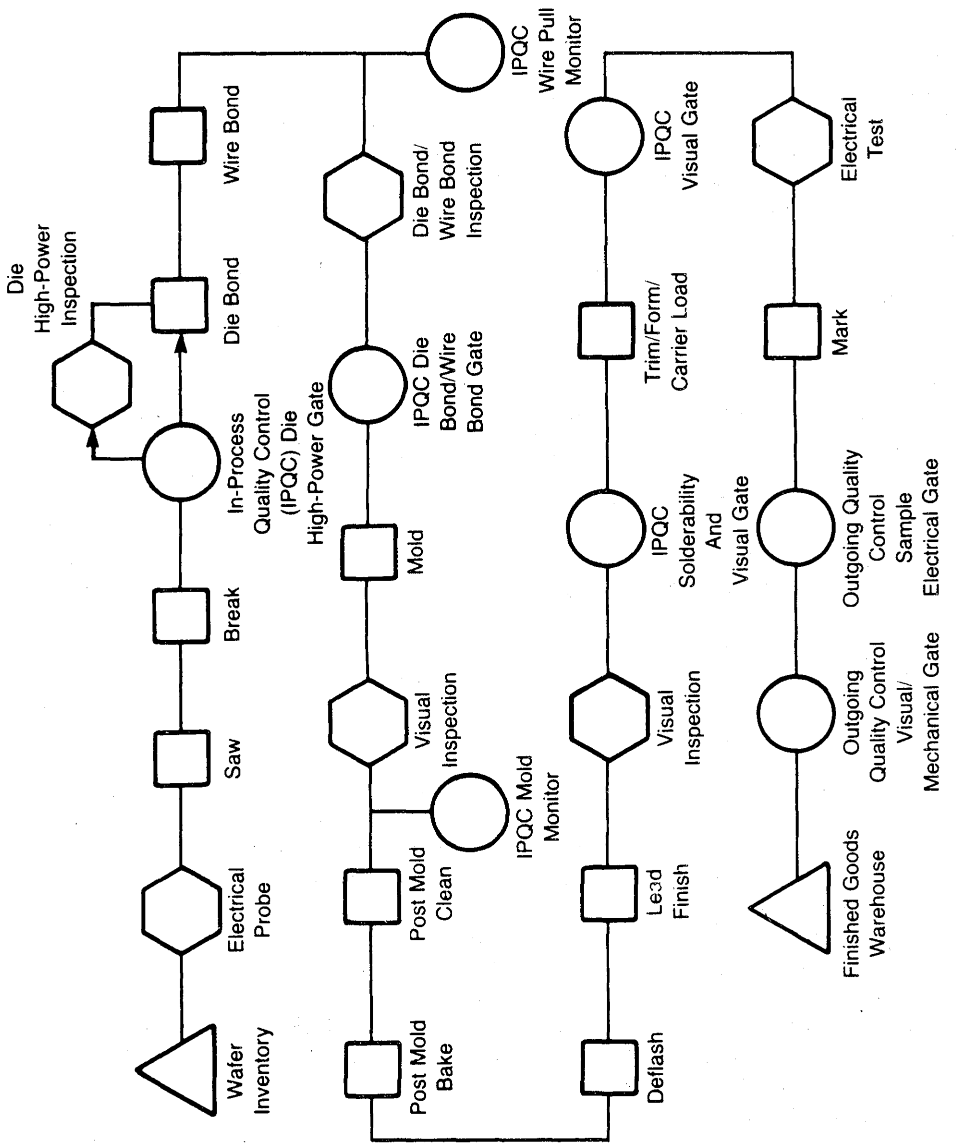


FIGURE B1. HERMETIC PROCESS FLOW

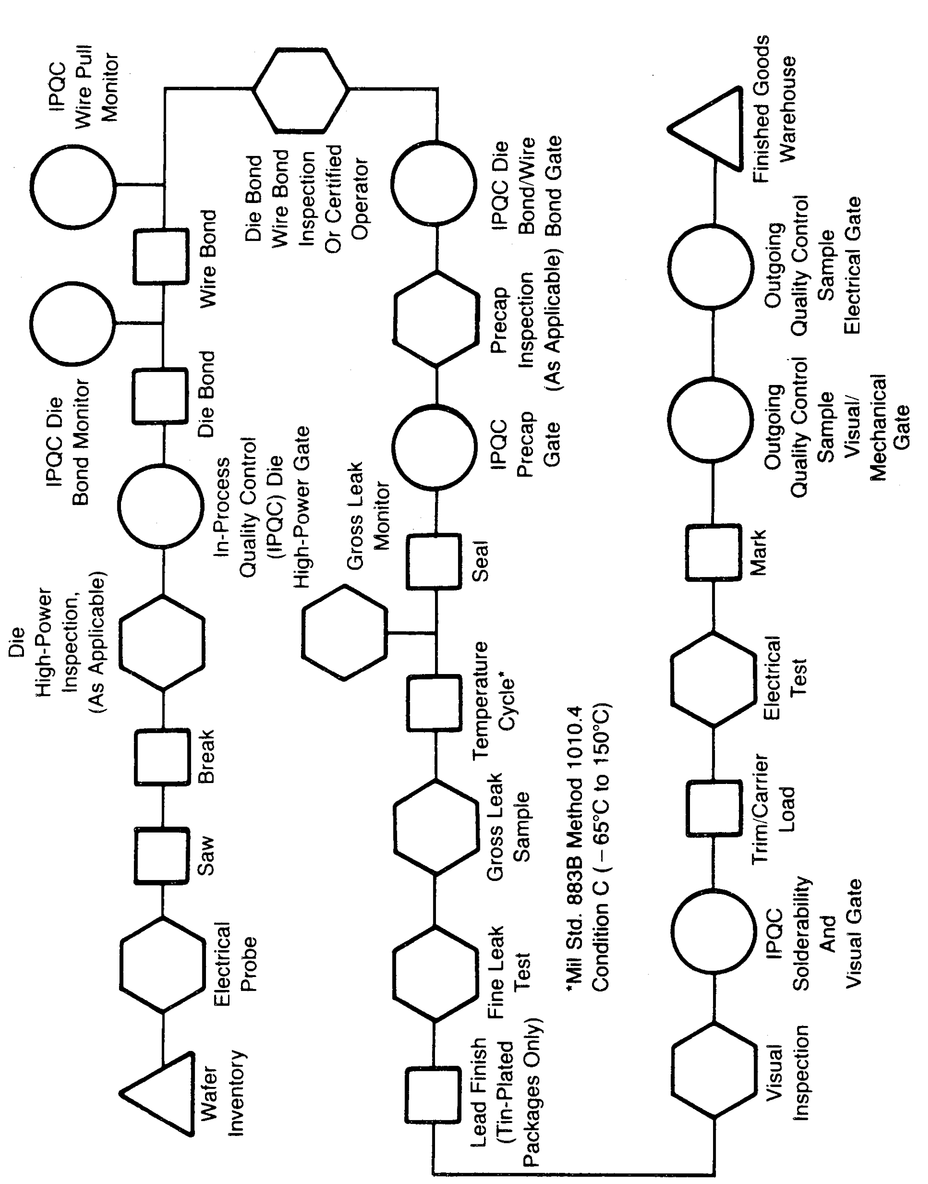


FIGURE B2. PLASTIC PROCESS FLOW

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

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

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

APPENDIX C.
FAILURE RATE CALCULATIONS

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

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

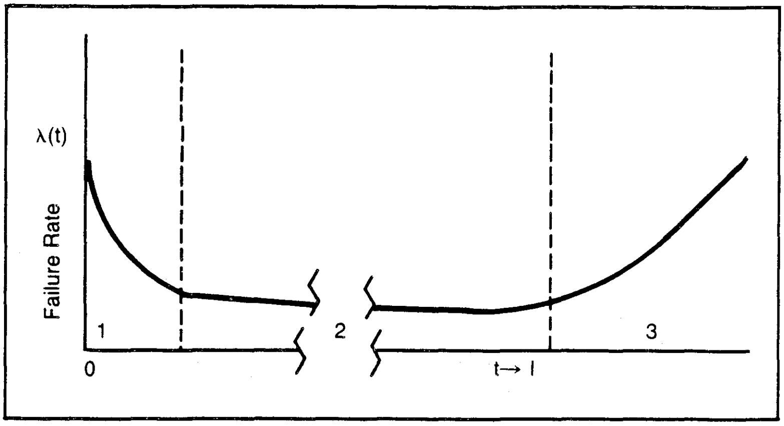


FIGURE C1. DEVICE FAILURE RATE AS A FUNCTION OF TIME.

The following three regions are represented in the curve:

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

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

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

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

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

2

where:

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

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

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

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

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

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

where:

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

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

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

$$T_J = T_A + P_D \cdot \theta_{JA} \tag{3}$$

2

where:

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

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

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

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

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

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. T.H.B. rejects are routinely decapsulated and inspected for corrosion of the metallization.

2

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

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

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

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

TABLE D1.
FAILURE MODE CLASSIFICATION

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

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

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

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

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

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

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

2

APPENDIX E.
MICROPROCESSOR AVERAGE JUNCTION TEMPERATURES AND GATE COUNTS

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

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

APPENDIX F.
RELIABILITY AND QUALITY MONITOR PROGRAM

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

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

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

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

High Temperature Operating Life

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

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

Temperature Humidity Bias

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

Autoclave

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

Temperature Cycle

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

Thermal Shock

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

Data Retention

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

RELIABILITY AND QUALITY MONITOR PROGRAM

SAMPLE GROUPS

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

2

APPENDIX G.
QUALITY PERFORMANCE

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

AVERAGE OUTGOING QUALITY

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

Data Sheets

3

MC1372

COLOR TV VIDEO MODULATOR

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

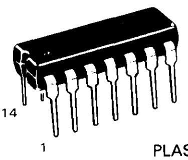
The MC1372 contains a chroma subcarrier oscillator, a lead and lag network, a quasi-quadrature suppressed carrier DSB chroma modulator, an RF oscillator and modulator, and an LSTTL compatible clock driver with adjustable duty cycle.

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

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

COLOR TV VIDEO MODULATOR CIRCUIT

SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 646-05

Pin Connections

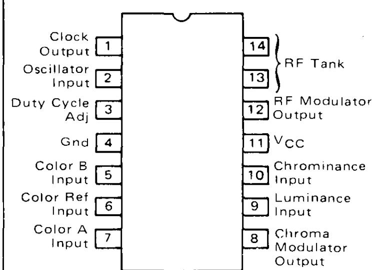
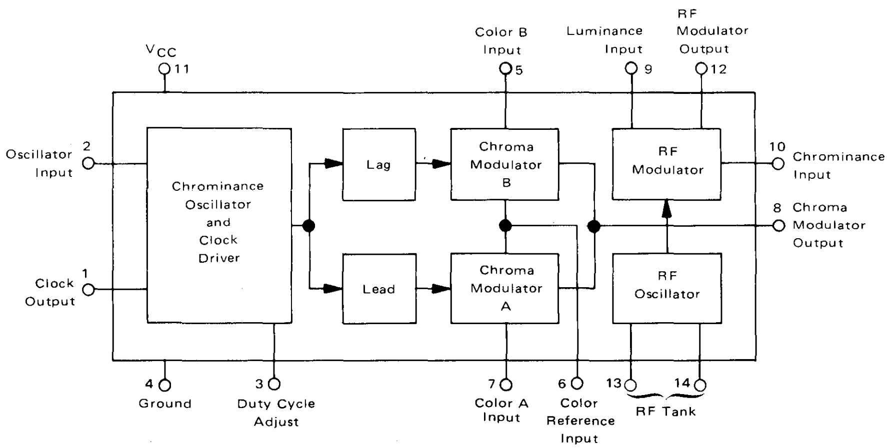


FIGURE 1 — BLOCK DIAGRAM



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

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

RECOMMENDED OPERATING CONDITIONS

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

3

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.

3

FIGURE 2 – TEST CIRCUIT 1

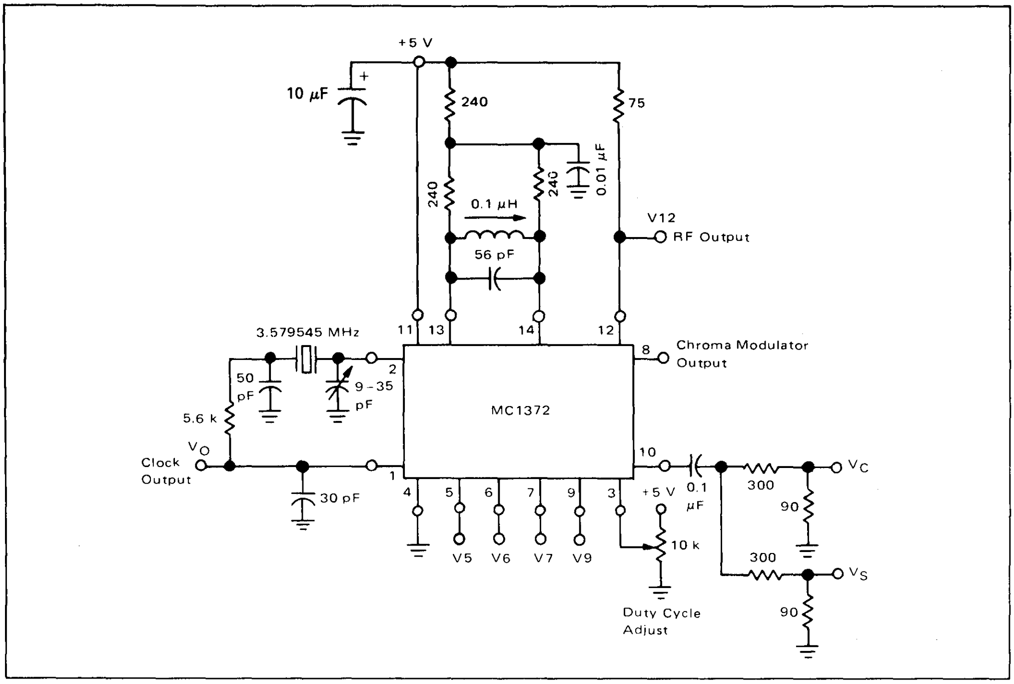


FIGURE 3 – TEST CIRCUIT 2

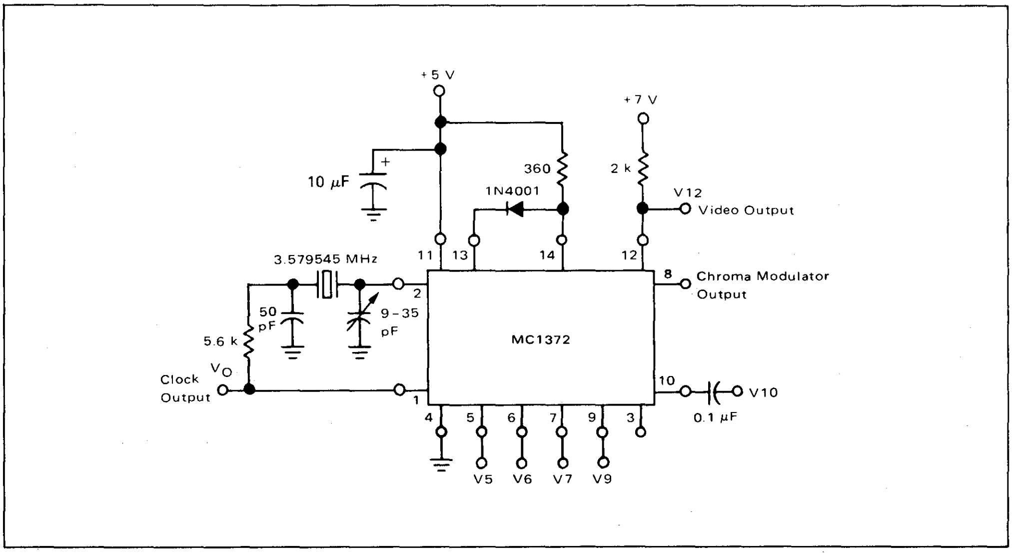
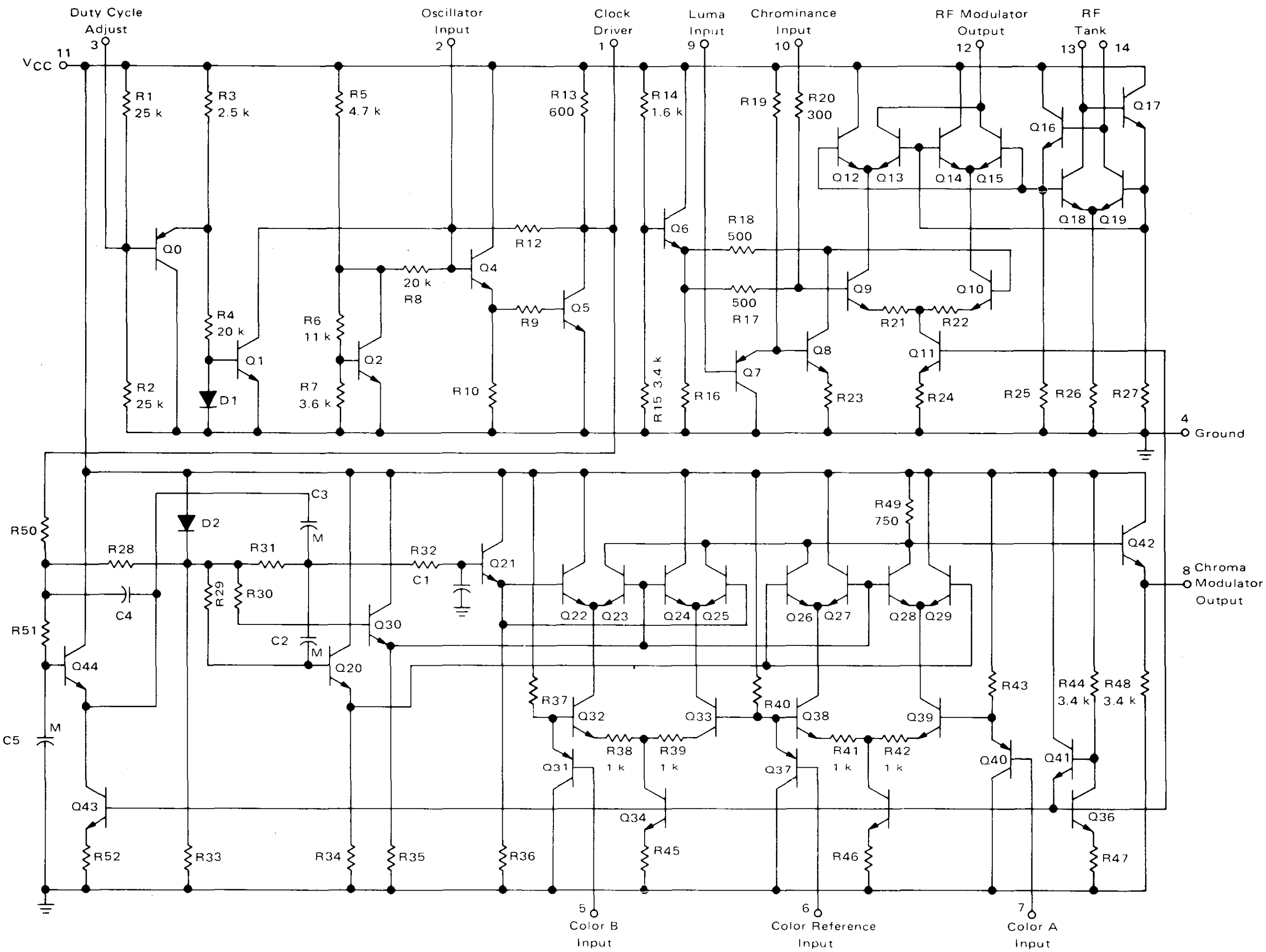


FIGURE 4 — SCHEMATIC DIAGRAM



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

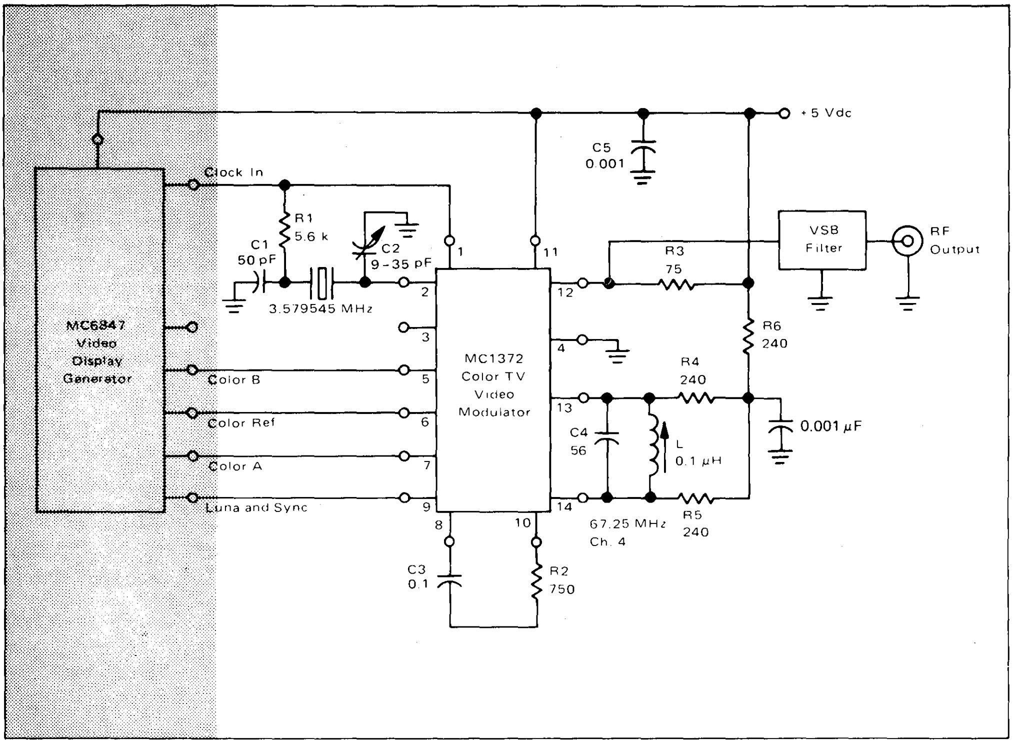
The coil and capacitor connected between pins 13 and 14 should be selected to have a parallel resonance at the carrier frequency of the desired TV channel. The values of 56 pF and 0.1 μH shown in Figure 5 were chosen for a Channel 4 carrier frequency of 67.25 MHz. For Channel 3 operation, the resonant 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.

3

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

3

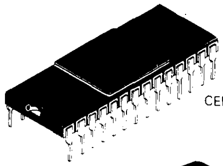
MOTOROLA

MC2670

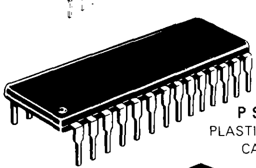
Advance Information

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

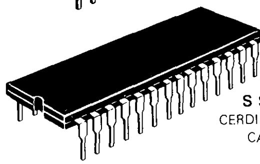
DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG)



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



S SUFFIX
CERDIP PACKAGE
CASE 733

PIN ASSIGNMENT

LA0	1	28	V_{CC}
LSTROBE	2	27	LA1
CA0	3	26	LA2
CA1	4	25	LA3
CA2	5	24	D0
CA3	6	23	D1
CSTROBE	7	22	D2
CA4	8	21	D3
CA5	9	20	D4
CA6	10	19	D5
CA7	11	18	D6
GM	12	17	D7
SCD	13	16	D8
V_{SS}	14	15	D9

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG)

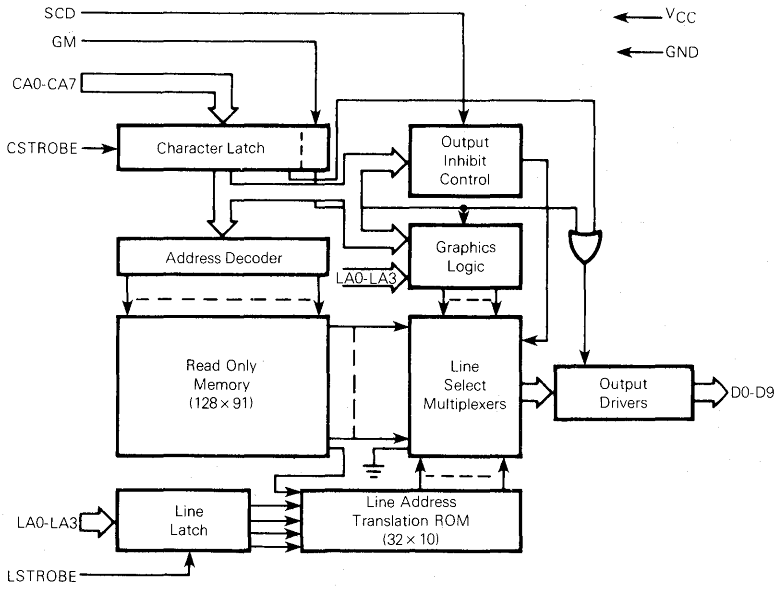
The MC2670 display character and graphics generator (DCGG) is a mask-programmable 11,648-bit line-select character generator. It contains 128 10×9 characters placed in a 10×16 matrix, and has the capability of shifting certain characters such as j, y, g, p, and q that normally extend below the baseline. Character shifting, previously requiring additional external circuitry, is now accomplished internally by the DCGG; effectively, the nine active lines are lowered within the matrix to compensate for the character's position.

Seven bits of an 8-bit address code are used to select one of 128 available characters. The eighth bit functions as a chip-enable signal. Each character is defined by a pattern of logic ones and zeros stored in a 10×9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears at the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10×9 character in one of two pre-programmed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

The MC2670 includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The MC2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user-programmable graphic patterns. Thus, data can be generated for 384 distinct patterns, of which 128 are defined by the mask-programmable ROM. Features include:

- 128 10×9 Matrix Characters
- 256 Graphic Characters
- Optional Thin Graphics for Forms
- Character and Line Address Latches
- Internal Descent Logic
- 200 Nanosecond and 300 Nanosecond Character Select Access Time Versions
- Control Character Output Inhibit Logic
- Static Operation — No Clocks Required
- Single 5-Volt Power Supply
- TTL Compatible Inputs and Outputs

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:

$T_A \equiv$ 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 $P_{PORT} \odot P_{INT}$ 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.

3

MAXIMUM RATINGS

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to 7.0	V
Input Voltage with Respect to Ground	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 electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to the high-impedance circuit.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Plastic	θ_{JA}	115	°C/W
Ceramic		60	
Cerdip		65	

DC ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0 V ±5%, See Notes 1, 2, and 3)

Parameter	Symbol	Min	Typ	Max	Unit
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Output Low Voltage I_{Load} = 1.6 mA	V_{OL}	–	0.3	0.4	V
Output High Voltage I_{Load} = −100 μA	V_{OH}	2.4	3	–	V
Input Leakage Current V_{in} = 0 to 4.25 V	I_{IL}	–	–	10	μA
Hi-Z (Off-State) Leakage Current V_{CC} = 5.25 V, V_{in} = 0.4 to 2.4 V	I_{TSL}	−10	–	10	μA
Internal Power Dissipation V_{CC} = 5.25 V, T_A = 0°C Minimum	P_{INT}	–	200	420	mW
Input Capacitance (All Other Pins Grounded)	C_{in}	–	–	10	pF
Output Capacitance	C_{out}	–	–	15	pF

NOTES:
1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at the 0.8 V to 2.0 V level for inputs and outputs. Input levels are 0 V to 2.4 V.
3. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

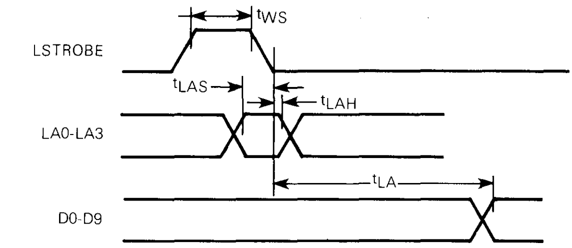
AC ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5 V ±5%, See Notes 1, 2, 3, and 4)

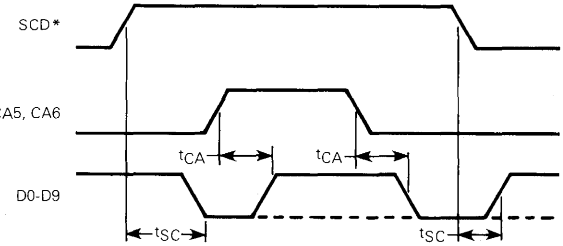
Parameter	Symbol	MC2670*3 Min	MC2670*3 Max	MC2670*2 Min	MC2670*2 Max	Unit
Strobe Pulse Width	t_{WS}	100	–	100	–	ns
Line Address Setup Time	t_{LAS}	50	–	50	–	ns
Line Address Hold Time	t_{LAH}	25	–	25	–	ns
Character Address Setup Time	t_{CAS}	25	–	15	–	ns
Character Address Hold Time	t_{CAH}	25	–	15	–	ns
Character Select Access Time	t_{CA}	–	300	–	200	ns
Line Select Access Time	t_{LA}	–	500	–	350	ns
Chip Select Delay Time	t_{SEL}	–	250	–	150	ns
Chip Deselect Delay Time	t_{DES}	–	200	–	125	ns
Special Character Blank/Unblank Time	t_{SC}	–	300	–	200	ns

* Substitute letter corresponding to standard font for (*) in part number for standard parts. Refer to **ORDERING INFORMATION** for additional information.

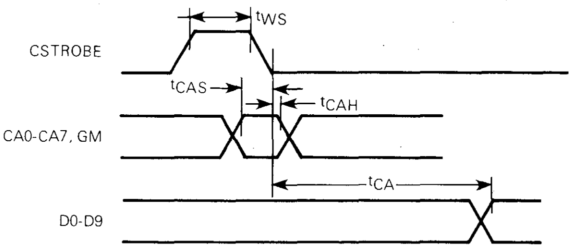
NOTES:
1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at the 0.8 V to 2.0 V level for inputs and outputs. Input levels are 0 V and 2.4 V.
3. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
4. Test conditions: C_L = 100 pF and 1 TTL load.

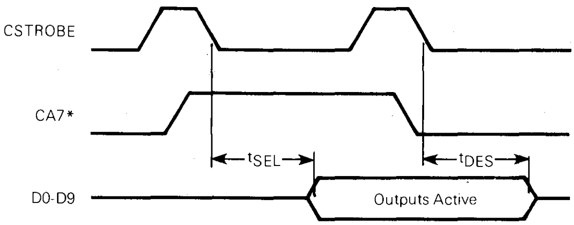
TIMING DIAGRAMS





*When GM = 1 SCD input is inactive





*CA7 operates as output enable only in character mode (GM = 0)

3

SIGNAL DESCRIPTION

The input and output signals for the DCGG are described in the following paragraphs.

V_{CC} AND V_{SS}

Power is supplied to the DCGG using these two pins. V_{CC} is the +5-volt power supply and V_{SS} is the ground connection.

CHARACTER ADDRESS (CA0-CA7)

This 8-bit input code specifies the character or graphic pattern for which matrix data is to be supplied. In character mode (GM = 0), CA0 through CA6 select one of the 128 ROM-defined characters and CA7 is a chip enable. The outputs are active when CA7 = 1 and are in the high-impedance state when CA7 = 0. In graphics mode (GM = 1), the outputs are active and CA0 through CA7 select one of 256 possible graphic patterns to be output.

CHARACTER STROBE (CSTROBE)

This input pin is used to store the character address (CA0 through CA7) and graphics mode (GM) inputs into the character latch. Data is latched on the negative going edge of CSTROBE.

GRAPHICS MODE (GM)

This input pin when low (GM = 0) selects the character mode and when high (GM = 1) selects graphics mode.

LINE ADDRESS (LA0-LA3)

When operating in the character mode, these input pins select one of the 16 lines of matrix data for the selected character to appear at the 10 outputs. LA0 is the least-significant bit and LA3 is the most-significant bit. The input codes which cause each of the nine lines of character data to be output are specified as part of the programming data for both non-shifted and shifted fonts. Cycling through the nine specified counts at the LA0 through LA3 inputs causes successive lines of data to be output on D0 through D9. The seven non-specified codes for both non-shifted and shifted characters cause blanks (logic zeros) to be output. In graphics mode, the line address gates the latched graphics data directly to the outputs.

LINE STROBE (LSTROBE)

This input pin is used to store the line address data (LA0 through LA3) in the line address latch. Data is latched on the negative going edge of LSTROBE.

SELECTED CHARACTER DISABLE (SCD)

In character mode, a high level at this input causes all outputs (regardless of line address) to be blanks (zeros) for characters for which CA6 and CA5 are both zero. A low level input selects normal operation. SCD is inoperative in the graphics mode.

DATA OUTPUTS (D0-D9)

These outputs provide data for the specified character and line.

FUNCTIONAL DESCRIPTION

The DCGG consists of nine major sections which are described in the following paragraphs. Line and character codes are strobed into the line and character latches. The character latch outputs are presented to the three sources of data; the ROM through an address decoder, the graphics logic, and the output inhibit control. The output inhibit control (together with the SCD input) suppresses the ROM data for selected character codes. The outputs from the line latch drive the line address translation ROM which maps the character ROM data onto 9 of 16 line positions. Finally, the line select multiplexers route the ROM or graphics data to the output drivers on D0 through D9.

CHARACTER LATCH

The character latch is a 9-bit edge-triggered latch used to store the character address (CA0 through CA7) and graphics mode (GM) inputs. The data is stored on the falling edge of CSTROBE. Seven latched addresses (CA0 through CA6) are inputs to the ROM character address decoder. In character mode (GM = 0), CA7 operates as a chip enable. The output drivers are enabled when CA7 = 1 and are in the high-impedance state when CA7 = 0. In graphics mode (GM = 1), the output drivers are always enabled and the CA0 through CA7 outputs of the latch are used to generate graphic symbols.

CHARACTER ADDRESS DECODER

This circuit decodes the 7-bit character address from the character latch to select one of the 128 character fonts stored in the ROM section of the DCGG.

READ-ONLY MEMORY

The 11,648-bit ROM stores the fonts for the 128 matrix-defined characters. The data for each character consists of 91 bits. Ninety bits represent the 10×9 matrix and one bit specifies whether the character data is output at the normal (unshifted) lines or at the descended (shifted) lines. The 90 data bit outputs are supplied to the line select multiplexers. The descend control bit is an input to the line address translation ROM.

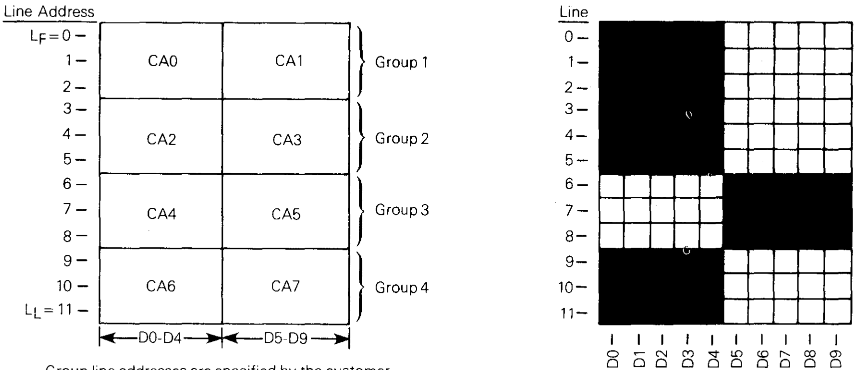
GRAPHICS LOGIC

When the GM input is zero (low), the DCGG operates in the character mode. When it is one (high), it operates in the graphics mode. In graphics mode, output data is generated by the graphics logic instead of the ROM. The graphics logic maps the latched character address (CA0 through CA7) to the outputs (D0 through D9) as a function of line address (LA0 through LA3). For any particular line address value, two of the CA bits are output: CA0, CA2, CA4, or CA6 is output on D0 through D4 and CA1, CA3, CA5, or CA7 is output on D5 through D9. The outputs are paired: when CA0 is output on D0 through D4, CA1 is output on D5 through D9 and likewise for CA2-CA3, CA4-CA5, and CA6-CA7.

A ROM within the graphics logic allows the specific line numbers for which each pair of bits is output to be specified by the customer. Figure 1 illustrates the general format for graphics symbols and an example where (CA7 through CA0) = H'65'. The outputs from the graphics logic go to the line select multiplexers. The multiplexers route the graphic symbol data to the outputs when GM = 1.

As a customer specified option, 16 of the possible graphic codes (H'80' to H'8F') may be used to generate the special graphic characters illustrated in Figure 2. For each of these characters, the vertical component appears on the D4 output. The horizontal components occur on L_H which is specified by the customer. The vertical components specified by CA0 and CA2 are output for line addresses zero through L_H and L_H through 15, respectively.

FIGURE 1 — GENERAL FORMAT GRAPHIC SYMBOLS



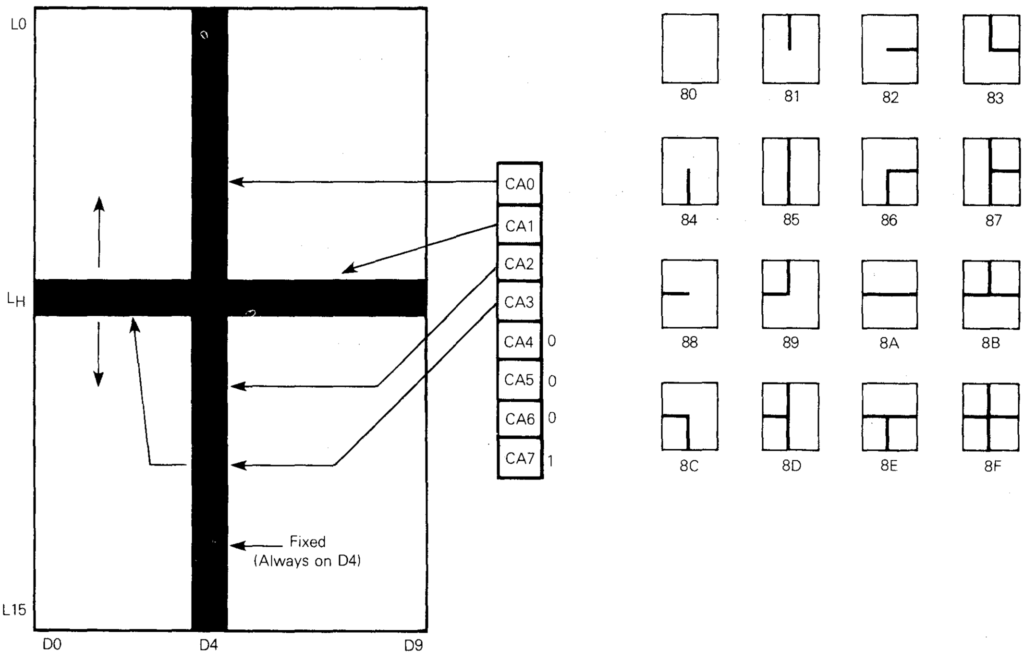
Group line addresses are specified by the customer.

Example: CA7-CA0 = H'65'
Group 1 Specified for Lines 0, 1, 2
Group 2 Specified for Lines 3, 4, 5
Group 3 Specified for Lines 6, 7, 8
Group 4 Specified for Lines 9, 10, 11
Space Specified for Lines 12, 13, 14, 15

FIGURE 2 — SPECIAL GRAPHIC CHARACTERS

General Format
10 × 16 Character Block

Thin Graphics Fonts for CA7-CA0 =
HEX 80 – HEX 8F



LINE SELECT MULTIPLEXERS

The 10 line select multiplexers select ROM data as specified by the line address translation ROM when GM = 0, or graphics data when GM = 1. The inputs to each multiplexer are the nine line outputs from the ROM, an output from the graphics logic, and a logic zero (ground).

OUTPUT DRIVERS

Ten output drivers with three-state capability serve as buffers between the line select multiplexers and external logic. The three-state control input to these drivers is supplied from the CA7 logic when GM = 0. When GM = 1, the outputs are always active.

OUTPUT INHIBIT CONTROL

The output inhibit control logic operates only if GM = 0. It causes the output of the line select multiplexers to be logic zero if the SCD input is high and CA6 and CA5 of the latched character address are D0. If the SCD input is low, normal operation occurs. (This feature is useful in ASCII coded applications to selectively disable character generation for non-displayable characters such as line feed, carriage return, etc.)

LINE ADDRESS LATCH

The line address latch is a 4-bit latch used to store the line address (LA0-LA3). The data is stored on the negative edge of the LSTROBE input.

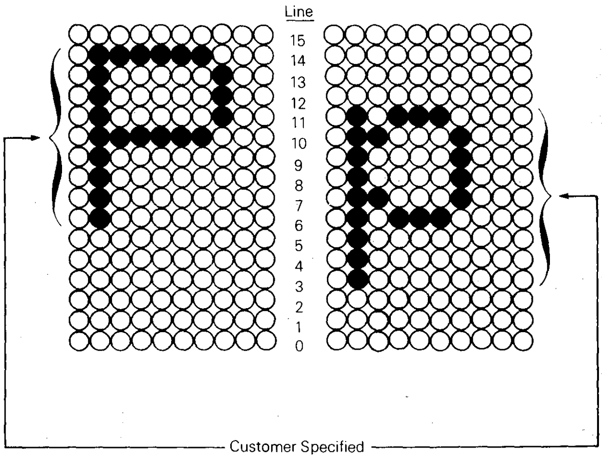
LINE ADDRESS TRANSLATION ROM

This 32 x 10 ROM translates the 5-bit code consisting of the four outputs from the line address latch and the descend control bit from the ROM into a one-of-ten code for the line select multiplexers. Programming information provided by the customer specifies the address which selects each line of ROM data for both shifted and non-shifted characters. Thus, there are nine line addresses which select ROM data for unshifted characters and nine addresses for shifted characters. These combinations are usually specified by the customer in either ascending or descending order. For the remaining 14 codes (seven each for unshifted and shifted characters), the translation ROM forces zeros at the outputs of the line select multiplexers.

This circuitry only operates if GM = 0. When GM = 1, the line select multiplexers are forced to select the outputs from the graphics logic.

Figure 3 shows an example of data outputs where the customer has specified line 14 as the first line for unshifted characters, line 11 as the first line for shifted characters, and line address combinations in descending order.

FIGURE 3 — CUSTOMER SPECIFIED EXAMPLE



ORDERING INFORMATION

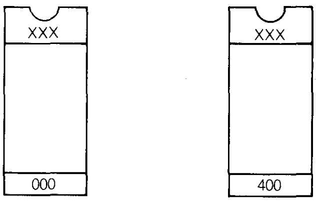
The information required when ordering a custom DCGG is listed below, a sample card deck input is given in Figure 4, and matrix font drawings in Figure 5. The ROM program may be transmitted to Motorola on EPROM(s) or an MDOS disk file.

To initiate a ROM pattern for the DCGG it is necessary to first contact your local Motorola representative or Motorola distributor.

EPROMs

The MCM68764 or MCM2532 type EPROMs, programmed with the customer program (positive logic sense for address and data), may be submitted for pattern generation (see example below). The ROM data format should be D0 through D9 programmed into two 8-bit words with D0 being the lease significant bit. Characters are programmed in sequence starting with CA equal to zero being the first character and CA equal to 7F being the last character. The third bit of the second word of each character is left as a one for unshifted characters and changed to zero for shifted characters. The remaining five bits are programmed to zeros.

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 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 care the custom mask and aid in the verification process.

ROM VERIFICATION UNITS (RVUs)

Ten DCGGs 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 an object file made from a memory dump using the ROLLOUT command are acceptable.

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

EPROM Programming Example:

0 4 2 3 Hex Value of One Character Line

0 0 0 0   0 $\boxed{1}$ 0 0   0 0 1 0   0 0 1 1   Binary Value

Specifies a Shifted Character

D7							D0	
0	0	0	0	0	1	0	0	First Byte of EPROM

D7							D0	
0	0	1	0	0	0	1	1	Second Byte of EPROM

MC2670 DISPLAY CHARACTER AND GRAPHICS GENERATOR ORDERING INFORMATION

Date____________ Customer PO Number____________

Motorola Part Numbers:

Customer Company____________ MC____________

Address____________ SC____________

City____________ State____________ Zip____________

Country____________

Telephone____________ Extension____________

Customer Contact Person____________

Customer Part Number____________

Line numbers in hex which outputs the font word for non-shifted ROM fonts (9 line numbers)____________

Line numbers in hex which outputs the font word for shifted ROM fonts (9 line numbers)____________

Specify left-to-right or right-to-left printing of D0 through D0. Also top-to-bottom or bottom-to-top line addresses.

Thin graphics? ______Yes ______No

Line address in hex for horizontal segments of line graphics fonts____________

Graphics group number 1, 2, 3, 4, or blank corresponds to line address hex 0 through hex F. The group number specified will cause the graphic data generated by that group to be output on the corresponding line address.

Pattern Media (All other media requires prior factory approval)

______EPROMs (MCM68764 or MCM2532)

______Floppy Disk

______Other____________

Character Select Access____________

Temperature Range:____________

Marking Information (12 Characters Maximum)____________

Signature____________

Title____________

FIGURE 4 — SAMPLE CARD DECK INPUT

SAMPLE CARD DECK INPUT

```
2670/CP1000PA  2670 TEST RUN  04/16/79

THIN GRAPHICS=YES  HOR=7       1111222233334444
NONSHIFT=1,2,3,4,5,6,7,8,9   SHIFT=3,4,5,6,7,8,9,A,B        0    0
00 N 022 026 02A n32 0AA 088 088 088 070     40 N 078 084 0B2 nCA 08A 072 002 084 078
01 N 01C 002 00C n10 08E 088 0F8 088 088     41 N 010 028 044 n82 082 0FE 082 082 082
02 N 01C 002 00C n10 08E 050 020 050 088     42 N 03E 044 084 n44 03C 044 084 044 03E
03 N 01E 002 00E 002 09E 050 020 050 088     43 N 078 084 002 n02 002 002 002 084 078
04 N 01E 002 00E n02 01E 0F8 020 020 020     44 N 03E 044 084 n84 084 084 084 044 03E
05 N 01E 002 00E n02 06E 090 090 0D0 0E0     45 N 0FE 002 002 n02 03E 002 002 002 0FE
06 N 00C 012 01E n12 092 050 030 050 090     46 N 0FE 002 002 n02 03E 002 002 002 002
07 N 00E 012 00E n12 00E 010 010 010 0F0     47 N 078 084 002 n02 002 0E2 082 0C4 0B8
08 N 00E 012 00E 012 0EE 010 060 080 070     48 N 082 082 082 n82 0FE 082 082 082 082
09 N 012 012 01E n12 012 0F8 020 02C 020     49 N 07C 010 010 n10 010 010 010 010 07C
0A N 002 002 002 n1E 0F0 010 070 010 010     4A N 0E0 040 040 n40 040 040 042 042 03C
0B N 022 022 022 n14 008 0F8 020 020 020     4B N 082 042 022 n12 00A 016 022 042 082
0C N 01E 002 00E n02 0F2 010 070 010 010     4C N 002 002 002 n02 002 002 002 002 07E
0D N 01C 002 002 n02 07C 090 070 050 090     4D N 082 0C6 0AA n92 092 082 082 082 082
0E N 01C 002 00C n10 06E 090 090 090 060     4E N 082 082 086 n8A 092 0A2 0C2 082 082
0F N 01C 002 00C n10 0EE 040 040 040 0E0     4F N 038 044 082 n82 082 082 082 044 038
10 N 00E 012 012 n12 00E 010 010 010 0F0     50 N 07E 082 082 n82 07E 002 002 002 002
11 N 00E 012 012 n12 04E 060 040 040 0E0     51 N 038 044 082 n82 082 092 0A2 044 0B8
12 N 00E 012 012 n12 06E 090 040 020 0F0     52 N 07E 082 082 n82 07E 012 022 042 082
13 N 00E 012 012 n12 06E 080 060 080 070     53 N 078 084 002 n04 038 040 080 042 03C
14 N 00E 012 012 n12 04E 060 050 0F8 040     54 N 0FE 010 010 n10 010 010 010 010 010
15 N 012 016 01A n12 092 050 030 050 090     55 N 082 082 082 n82 082 082 082 044 038
16 N 01C 002 00C n10 08E 050 020 020 020     56 N 082 082 082 n44 044 028 028 010 010
17 N 01E 002 00E n02 07E 090 070 090 070     57 N 082 082 082 n82 082 092 092 0AA 044
18 N 01C 002 002 n02 01C 090 0F0 0D0 090     58 N 082 082 044 n28 010 028 044 082 082
19 N 01E 002 00E n02 01E 088 0D8 0A8 088     59 N 082 082 044 n28 010 010 010 010 010
1A N 01C 002 00C n10 07E 090 070 090 070     5A N 0FE 080 040 n20 010 008 004 002 0FE
1B N 01E 002 00E n02 01E 0E0 010 010 0E0     5B N 07C 004 004 n04 004 004 004 004 07C
1C N 01E 002 00E n02 0E2 010 060 080 070     5C N 000 002 004 n08 010 020 040 080 000
1D N 01C 002 01A n12 0EC 010 060 080 070     5D N 07C 040 040 n40 040 040 040 040 07C
1E N 00E 012 00E n0A 0F2 010 060 080 070     5E N 010 038 054 n10 010 010 010 010 010
1F N 012 012 012 n12 0EC 010 060 080 070     5F N 000 000 008 n04 0FE 004 008 000 000
20 N 000 000 000 n00 000 000 000 000 000     60 N 018 018 010 020 000 000 000 000 000
21 N 010 010 010 n10 010 000 000 010 010     61 N 000 000 000 n3C 040 07C 042 042 0BC
22 N 028 028 028 n28 000 000 000 000 000     62 N 002 002 002 n3A 046 042 042 046 03A
23 N 028 028 0FE n28 028 028 0FE 028 028     63 N 000 000 000 n3C C42 002 002 042 03C
24 N 028 0FC 02A n2A 07C 0A8 0A8 07E 028     64 N 040 040 040 n5C 062 042 042 062 05C
25 N 004 08A 044 n20 010 008 044 0A2 040     65 N 000 000 000 n3C 042 07E 002 002 03C
26 N 00C 012 012 n0C 00C 012 0A2 042 0BC     66 N 030 048 008 n08 03E 008 008 008 008
27 N 018 018 008 n04 000 000 000 000 000     67 S 000 05C 062 n42 062 05C 040 042 03C
28 N 020 010 008 n08 008 008 CC8 010 020     68 N 002 002 002 n3A 046 042 042 042 042
29 N 008 010 020 n20 020 020 020 010 008     69 N 000 010 000 n18 010 010 010 010 038
2A N 000 010 054 n38 0FE 038 054 010 000     6A S 000 060 040 n40 040 040 040 044 038
2B N 000 010 010 n10 0FE 010 010 010 000     6B N 002 002 002 n22 012 00A 016 022 042
2C S 000 000 000 n00 000 018 018 008 004     6C N 018 010 010 n10 010 010 010 010 038
2D N 000 000 000 n00 0FE 000 000 000 000     6D N 000 000 000 n6A 096 092 092 092 092
2E N 000 000 000 n00 000 000 000 018 018     6E N 000 000 000 n3A 046 042 042 042 042
2F N 000 080 040 n20 010 008 004 002 000     6F N 000 000 000 n3C 042 042 042 042 03C
30 N 038 044 0C2 nA2 092 08A 086 044 038     70 S 000 03A 046 n42 046 03A 002 002 002
31 N 010 018 014 n10 010 010 010 010 07C     71 S 000 05C 062 n42 062 05C 040 040 040
32 N 07C 082 080 n40 038 004 002 002 0FE     72 N 000 000 000 n3A 046 002 002 002 002
33 N 07C 082 080 n80 070 080 080 082 07C     73 N 000 000 000 n3C 042 00C 030 042 03C
34 N 040 060 050 n48 044 0FE 040 040 040     74 N 000 008 008 n1C 008 008 008 048 030
35 N 0FE 002 002 n02 07E 080 080 082 07C     75 N 000 000 000 n42 042 042 042 062 05C
36 N 078 084 002 n02 07A 086 082 082 07C     76 N 000 000 000 n44 044 044 044 028 010
37 N 0FE 080 080 n40 020 010 008 004 002     77 N 000 000 000 082 082 092 092 092 06C
38 N 07C 082 082 n44 038 044 082 082 07C     78 N 000 000 000 n42 024 018 018 024 042
39 N 07C 082 082 nC2 0BC 080 080 042 03C     79 S 000 042 042 n42 062 05C 040 042 03C
3A N 000 000 000 n18 018 000 000 018 018     7A N 000 000 000 n7E 020 010 008 004 07E
3B S 000 018 018 n00 000 018 018 008 004     7B N 030 008 008 n08 004 008 008 008 030
3C N 020 010 008 n04 002 004 008 010 020     7C N 010 010 010 000 000 000 010 010 010
3D N 000 000 000 nFE 000 000 0FE 000 000     7D N 018 020 020 n20 040 020 020 020 018
3E N 008 010 020 n40 080 040 020 010 008     7E N 000 000 000 n0C 092 060 000 000 000
3F N 07C 082 082 n80 060 010 010 000 010     7F N 0AA 054 0AA n54 0AA 054 0AA 054 0AA
```

FIGURE 5 — 10 × 16 MATRIX FONT DRAWING (Sheet 1 of 4)

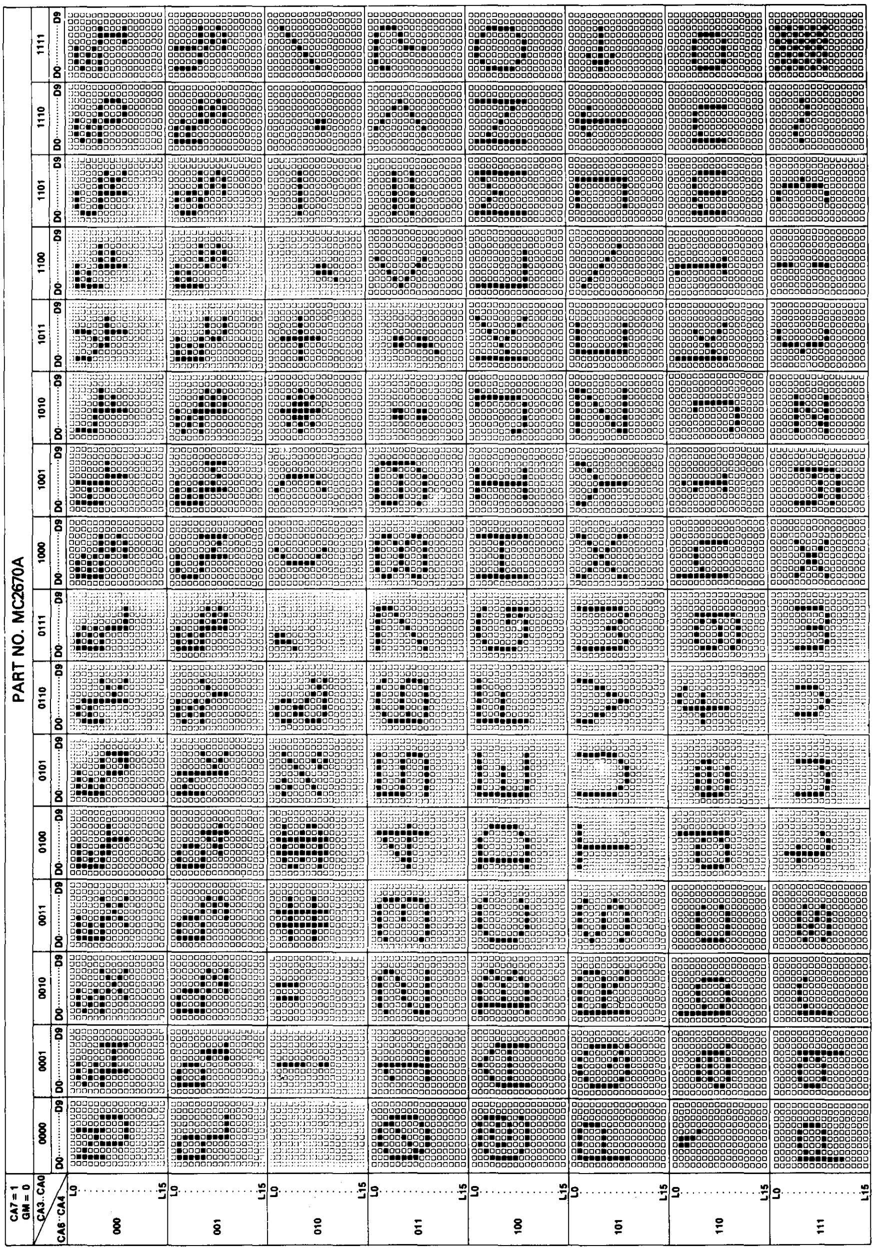


FIGURE 5 — 10 × 16 MATRIX FONT DRAWING (Sheet 2 of 4)

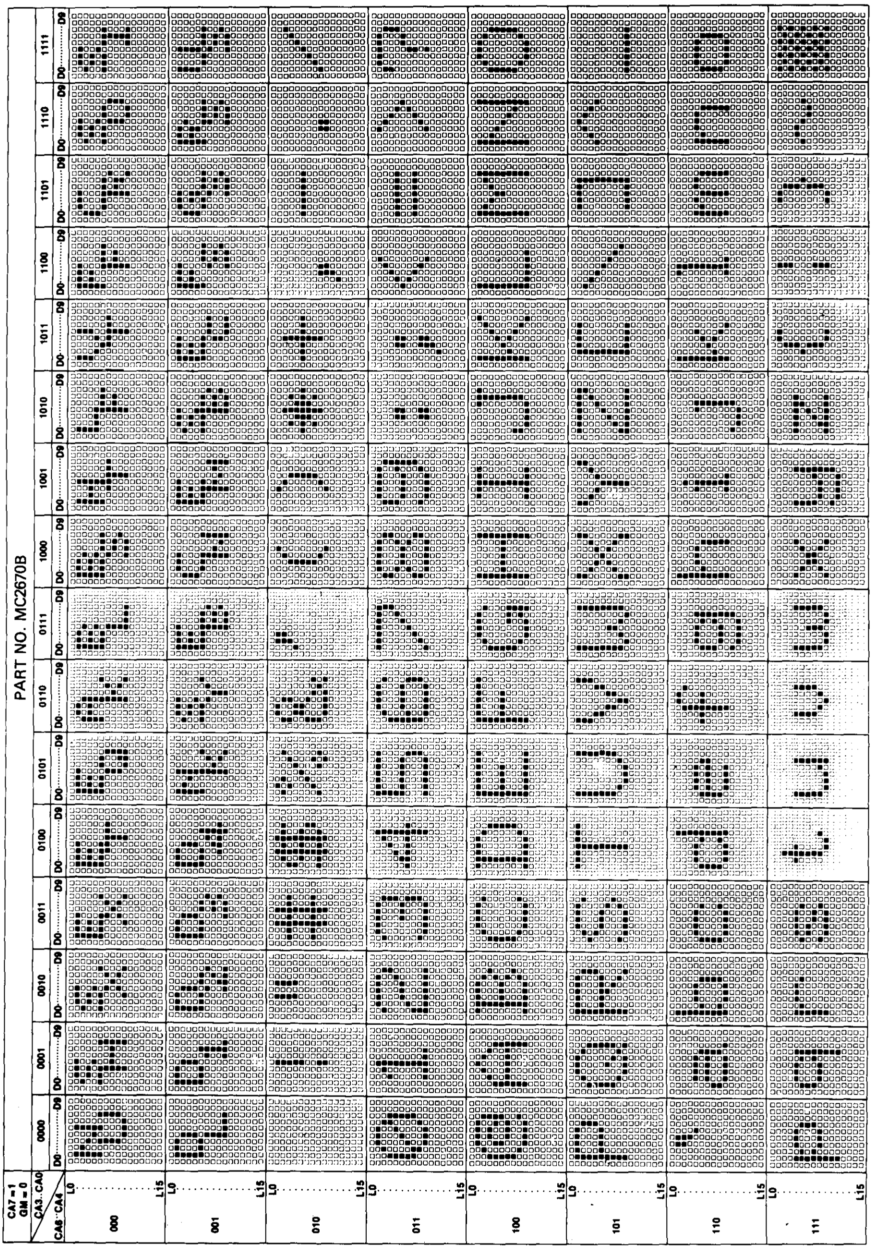


FIGURE 5 — 10 × 16 MATRIX FONT DRAWING (Sheet 3 of 4)

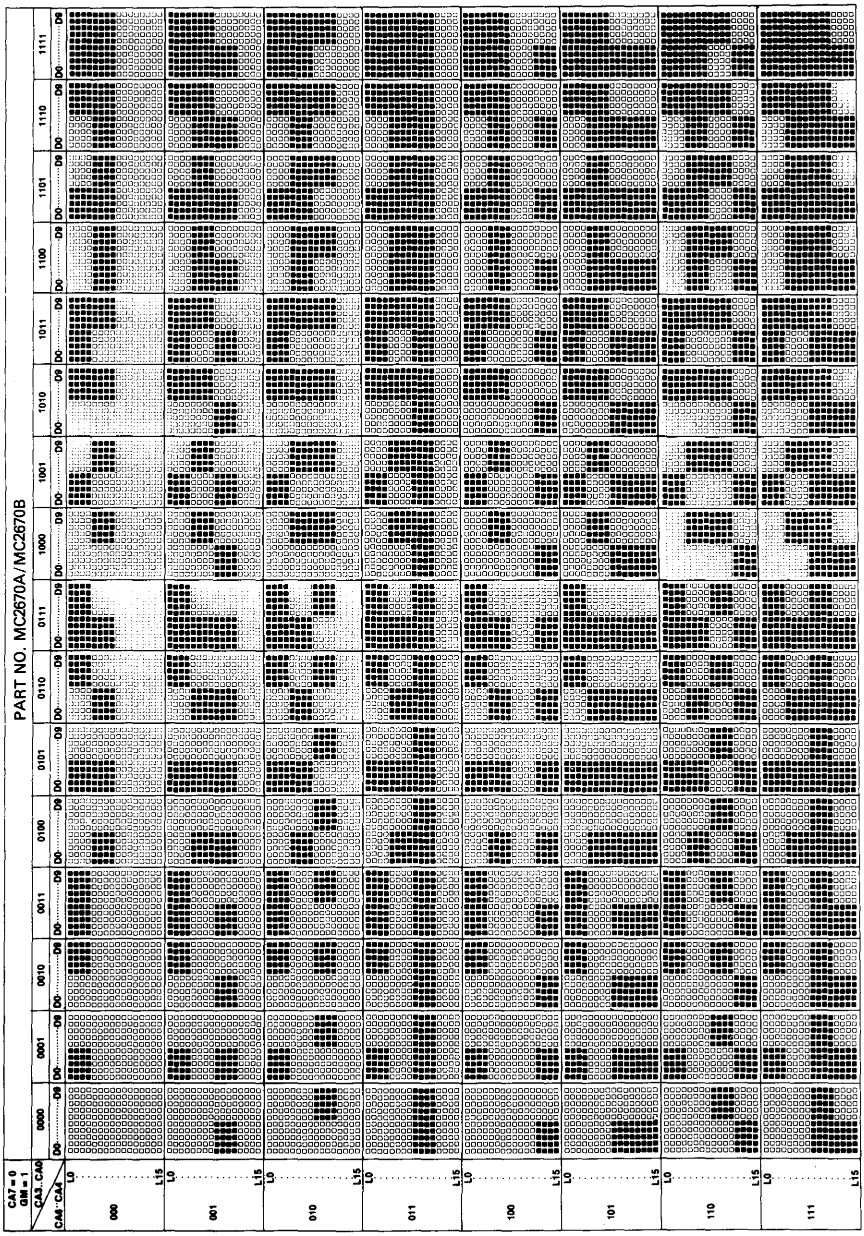
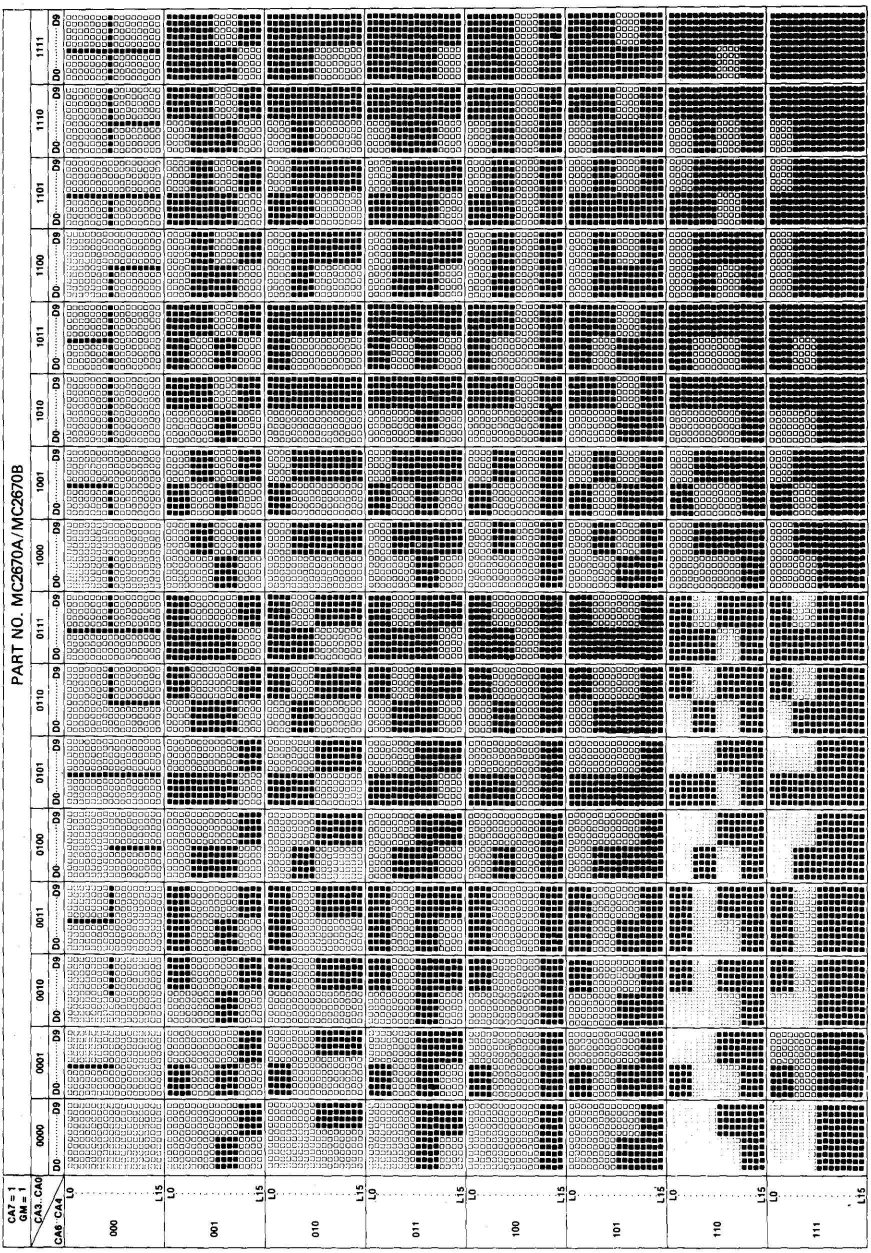


FIGURE 5 — 10 × 16 MATRIX FONT DRAWING (Sheet 4 of 4)



MOTOROLA

MC2671

Advance Information

PROGRAMMABLE KEYBOARD AND COMMUNICATIONS CONTROLLER (PKCC)

The MC2671 programmable keyboard and communications controller (PKCC) is an MOS/LSI device which provides a versatile keyboard encoder and an independent full-duplex asynchronous communications controller. It is intended for use in microprocessor-based systems and provides an 8-bit data bus interface.

Applications for the MC2671 include: CRT terminals, hard-copy terminals, word-processing systems, data-entry terminals, and small business computers.

- Keyboard Interface
- Contact or Capacitive Keyboard
- Up to 128 Keys on an 8 × 16 Matrix
- Encoded or Unencoded Operation
- Four Code Levels Per Key
- Latched Key Option — Separate Depress and Release Codes
- Programmable Scan Rate and Debounce Time
- Programmable Rollover Modes
- Programmable Auto-Repeat for Selected Keys
- Tone Output — Two Frequencies
- Asynchronous Communication Interface
- Internal Baud-Rate Generator — 16 Rates
- Full-Duplex Operation
- Detection of Start and End of Break
- Programmable Break Generation
- Programmable Character Parameters
- Auto-Echo and Maintenance Loopback Modes
- Polled or Interrupt Operation
- Interrupt Priority Controller and Vector Generator
- Operates Directly from Crystal or External Clocks
- TTL Compatible
- Single +5 Volt Power Supply

ORDERING INFORMATION

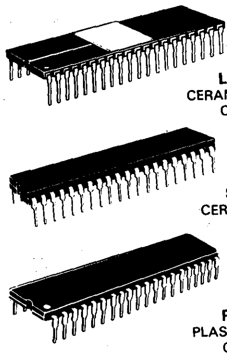
Package Type	Frequency	Temperature	Order Number
Ceramic L Suffix	1.0 MHz	0°C to 70°C	MC2671AL
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC2671AS
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC2671AP

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

HMOS

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

PROGRAMMABLE KEYBOARD AND COMMUNICATIONS CONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 715

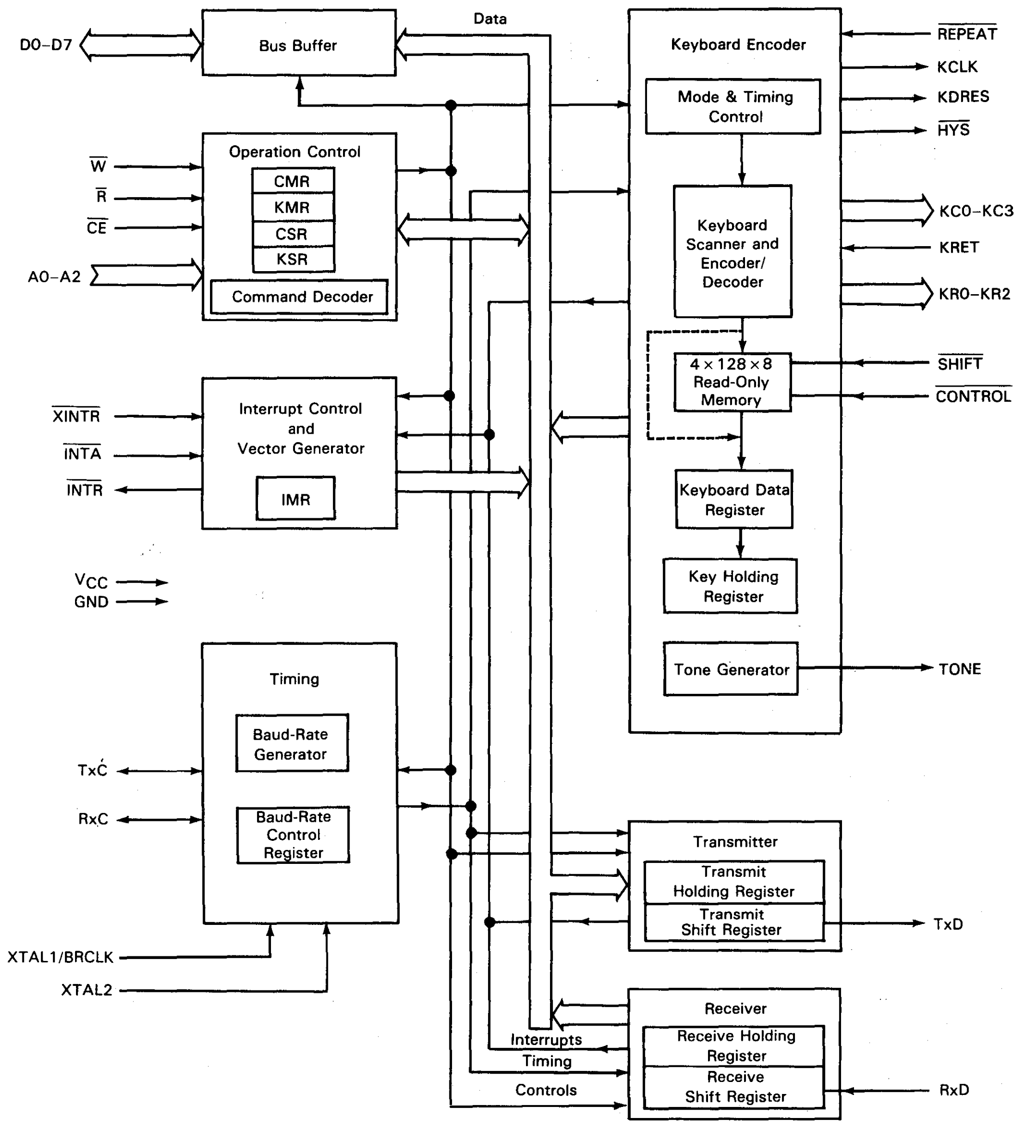
S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
$\overline{\text{HYS}}$	1	40	V_{CC}
KDRES	2	39	RxD
KCLK	3	38	TxD
KC3	4	37	XTAL2/BRCLK
KC2	5	36	XTAL1
KC1	6	35	RxC
KC0	7	34	TxC
KR2	8	33	A2
KR1	9	32	A1
KR0	10	31	A0
$\overline{\text{REPEAT}}$	11	30	$\overline{\text{CE}}$
$\overline{\text{SHIFT}}$	12	29	$\overline{\text{R}}$
$\overline{\text{CONTROL}}$	13	28	$\overline{\text{W}}$
TONE	14	27	$\overline{\text{INTA}}$
KRET	15	26	D7
D0	16	25	D6
D1	17	24	D5
D2	18	23	D4
D3	19	22	$\overline{\text{INTR}}$
V_{SS}	20	21	$\overline{\text{XINTR}}$

BLOCK DIAGRAM



3

MAXIMUM RATINGS

Characteristics	Symbol	Rating	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	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

Characteristics	Symbol	Rating	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic		100	
Cerdip		60	
Ceramic		50	

3

POWER CONSIDERATIONS

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

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

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

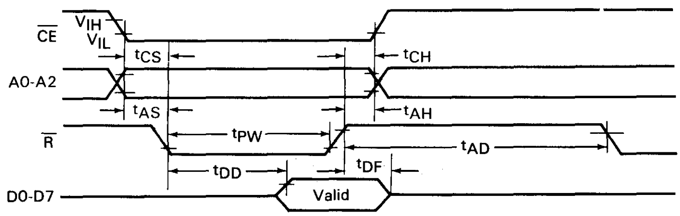
Parameter		Symbol	Min	Typ	Max	Unit
Input Low Voltage		V_{IL}	–	–	0.8	V
Input High Voltage	XTAL1, XTAL2	V_{IH}	4.0	–	–	V
	All Other Inputs		2,0	–	–	
Output Low Voltage (I_{OL} = 1.6 mA)		V_{OL}	–	–	0.4	V
Output High Voltage (Except $\overline{INTR}$) (I_{OH} = −100 μA)		V_{OH}	2.4	–	–	V
Input Leakage Current (V_{in} = 0 to V_{CC})	XTAL2/BRCLK	I_{IL}	–	−100	–	μA
	All Other Inputs		−10	–	10	
Data Bus Hi-Z Leakage Current (V_O = 0 to V_{CC})		I_{LL}	−10	–	10	μA
Power Supply Current		I_{CC}	–	–	150	mA

AC ELECTRICAL SPECIFICATIONS — READ CYCLE (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figure 1)

Parameter	Symbol	Min	Typ	Max	Unit
Address Setup Time to $\overline{R}$	t_{AS}	50	—	—	ns
$\overline{CE}$ Setup Time to $\overline{R}$	t_{CS}	50	—	—	ns
Read Cycle Pulse Width	t_{PW}	250	—	—	ns
Address Hold Time from $\overline{R}$	t_{AH}	20	—	—	ns
$\overline{CE}$ Hold Time from $\overline{R}$	t_{CH}	0	—	—	ns
Data Delay Time for Read Cycle (C_L = 150 pF)	t_{DD}	—	—	200	ns
Data Bus Floating Time for Read Cycle (C_L = 150 pF)	t_{DF}	10	—	100	ns
Access Delay Time from any Read to Next Read or Write	t_{AD}	250	—	—	ns

3

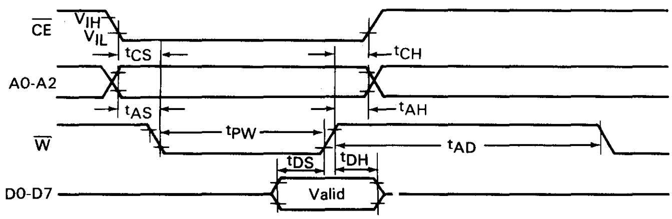
FIGURE 1 — READ CYCLE TIMING DIAGRAM



AC ELECTRICAL SPECIFICATIONS — WRITE CYCLE (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figure 2)

Parameter	Symbol	Min	Typ	Max	Unit
Address Setup Time to $\overline{W}$	t_{AS}	50	—	—	ns
$\overline{CE}$ Setup Time to $\overline{W}$	t_{CS}	50	—	—	ns
Write Cycle Pulse Width	t_{PW}	250	—	—	ns
Address Hold Time from $\overline{W}$	t_{AH}	20	—	—	ns
$\overline{CE}$ Hold Time from $\overline{W}$	t_{CH}	0	—	—	ns
Data Setup Time	t_{DS}	100	—	—	ns
Data Hold time	t_{DH}	10	—	—	ns
Access Delay Time from any Write to Next Read or Write	t_{AD}	250	—	—	ns
Access Delay Time from Reset Command to Next Read or Write	t_{AD}	1.0	—	—	μs

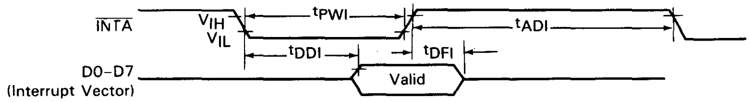
FIGURE 2 — WRITE CYCLE TIMING DIAGRAM



AC ELECTRICAL SPECIFICATIONS — INTERRUPT KNOWLEDGE (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figure 3)

Parameter	Symbol	Min	Typ	Max	Unit
$\overline{INTA}$ Pulse Width	t_{PWI}	300	–	–	ns
Data Delay Time for Interrupt Vector (C_L = 150 pF)	t_{DDI}	–	–	250	ns
Data Bus Floating Time after $\overline{INTA}$ (C_L = 150 pF)	t_{DFI}	10	–	100	ns
$\overline{INTA}$ to $\overline{INTA}$ Access Delay Time	t_{ADI}	300	–	–	ns

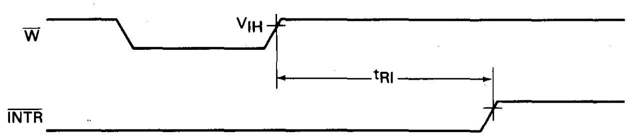
FIGURE 3 — INTERRUPT KNOWLEDGE TIMING



AC ELECTRICAL SPECIFICATIONS — INTERRUPT RESET (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figure 4)

Parameter	Symbol	Min	Typ	Max	Unit
$\overline{INTR}$ Delay Time from:	t_{RI}				ns
Read RxHR (RxRDY)		–	–	400	
Read KHR (KRDY)		–	–	400	
Reset Commands (KOVR, KERR, BREAK)		–	–	450	
Load TxHR (TxEMT, TxRDY)		–	–	400	
Mask Bit Reset		–	–	300	

FIGURE 4 — INTERRUPT RESET TIMING DIAGRAM

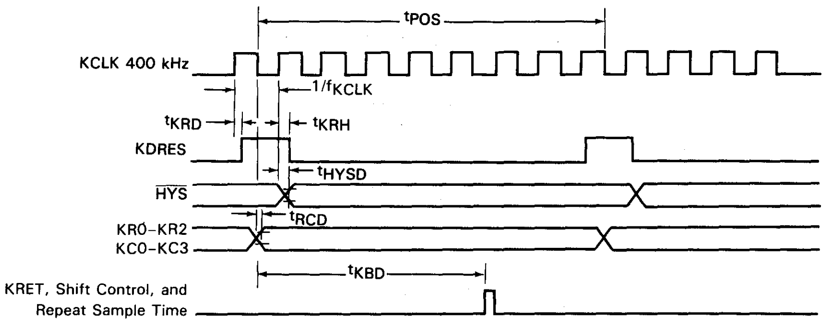


AC ELECTRICAL SPECIFICATIONS — KEYBOARD (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figures 5 and 6)

Parameter	Symbol	Min	Typ	Max	Unit
KCLK Frequency	f_{KCLK}	—	409	—	kHz
KRi, KCi, to KRET Sample Delay Time: Fast Scan Slow Scan	t_{KBD}	 12.0 55.0	 — —	 — —	μs
Scan Time per Matrix Position: Fast Scan Slow Scan	t_{POS}	 — —	 20 80	 — —	μs
KDRES Delay Time from KCLK (C_L = 150 pF)	t_{KRD}	—	—	400	ns
KDRES Hold Time from KCLK (C_L = 150 pF)	t_{KRH}	—	—	400	ns
$\overline{HYS}$ Delay Time from KCLK (C_L = 150 pF)	t_{HYSD}	—	—	600	ns
KRi, KCi Delay Time from KCLK (C_L = 150 pF)	t_{RCD}	—	—	400	ns

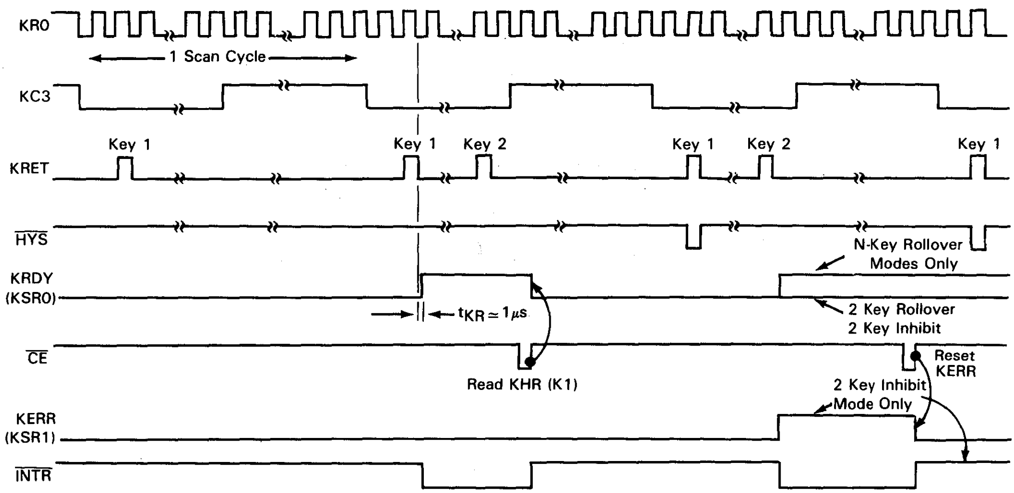
3

FIGURE 5 — KEYBOARD SCAN TIMING DIAGRAM



NOTE: Scan timing shown is for fast scan (KMR1 = 1). For slow scan (KMR1 = 0). All signals except KLCK run at ¼ the shown rates.

FIGURE 6 — KEYBOARD TIMING



AC ELECTRICAL SPECIFICATIONS — UART (T_A = 0°C to 70°C, V_{CC} = ±5%) (See Figures 7, 8, and 9)

Parameter	Symbol	Min	Typ	Max	Unit
RxD Setup Time	t_{RxS}	200	—	—	ns
RxD Hold Time	t_{RxH}	200	—	—	ns
TxD Delay Time from Falling Edge of TxC (C_L = 150 pF)	t_{TxD}	—	—	300	ns
Skew Between TxD Transition and Falling Edge of TxC Output (C_L = 150 pF)	t_{TCS}	—	0	—	ns
XTAL1 Clock High (see Figures 10 and 11)	t_{BRH}	70	—	—	ns
XTAL1 Clock Low (see Figures 10 and 11).	t_{BRL}	70	—	—	ns
BRG Input Frequency	f_{BRG}	1.0	4.915	5.075	MHz
TxC or RxC Input Frequency	$f_{R/T}$				MHz
Clock Rate Factor = 16X, 32X, 64X		—	—	1.3	
Clock Rate Factor = 1X		—	—	1.0	
TxC or RxC Clock High	$t_{R/TH}$	350	—	—	ns
TxC or RxC Clock Low	$t_{R/TL}$	350	—	—	ns

FIGURE 7 — CLOCK, TRANSMIT, AND RECEIVE TIMING DIAGRAMS

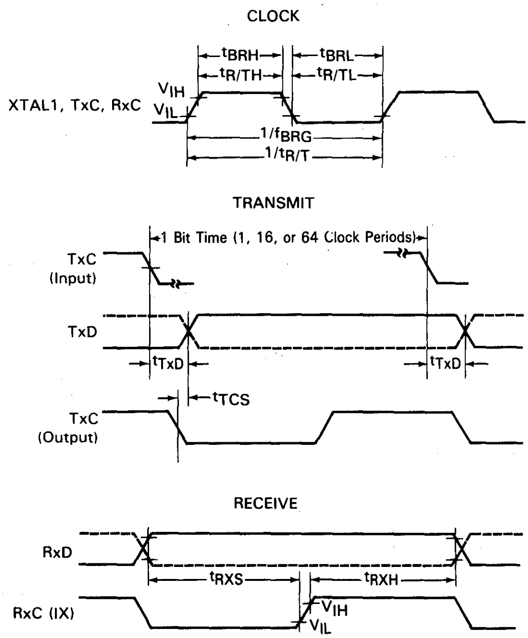
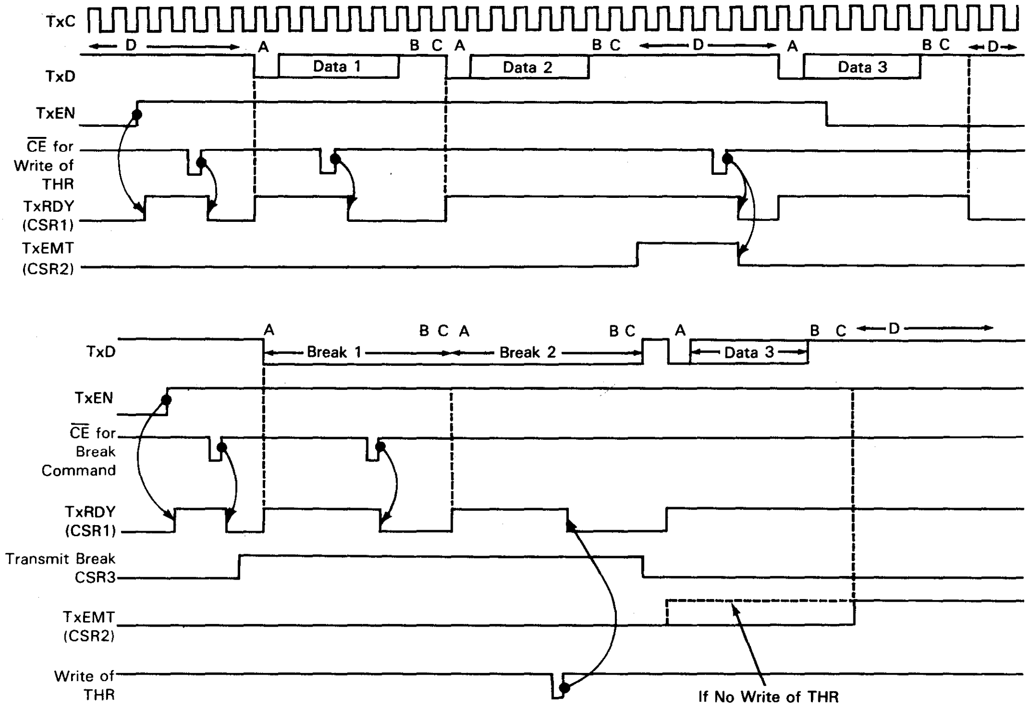


FIGURE 8 — TRANSMITTER TIMING DIAGRAM
(5-Bit Characters, No Parity, 2 Stop Bits)



A = Start Bit
B = First Stop Bit
C = Second Stop Bit
D = Mark

FIGURE 9 — RECEIVER TIMING DIAGRAM
(5-Bit Characters, No Parity, 2 Stop Bits)

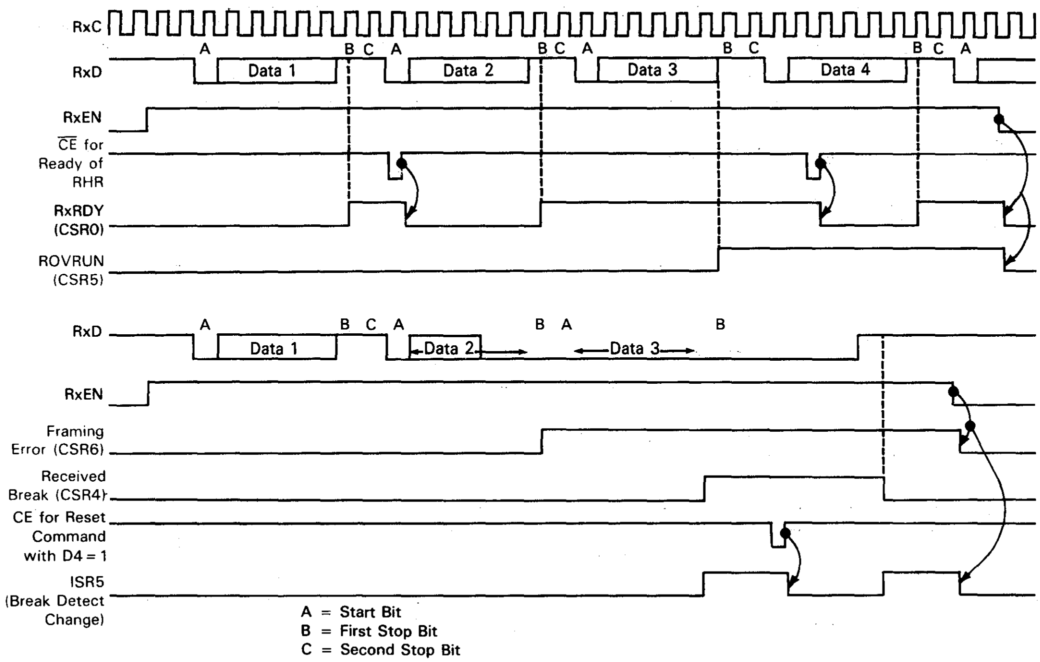


FIGURE 10 — CRYSTAL CONNECTIONS BRG CLOCK

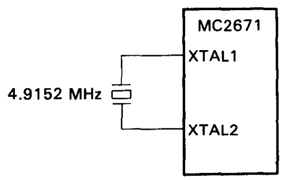
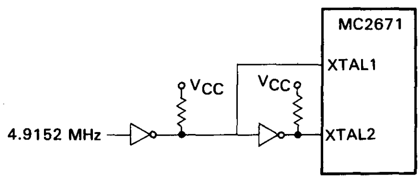


FIGURE 11 — CONNECTION FOR EXTERNAL BRG CLOCK SOURCE



3

SIGNAL DESCRIPTION

The input and output signals for the PKCC are described in the following paragraphs.

V_{CC} AND V_{SS}

Power is supplied to the PKCC using these two pins. V_{CC} is the +5 volt power supply and V_{SS} is the ground connection.

DATA BUS (D0-D7)

This 8-bit three-state bidirectional data bus makes all data, command, and status transfers. D0 is the least significant bit and D7 is the most signficant bit.

ADDRESS BUS (A0-A2)

These input lines are used to select internal PKCC registers or commands.

READ STROBE ($\overline{R}$)

This input, when low, gates the selected PKCC register onto the data bus if chip enable is also low.

WRITE STROBE ($\overline{W}$)

This input, when low, gates the contents of the data bus into the selected PKCC register if chip enable is also low.

CHIP ENABLE ($\overline{CE}$)

This input, when high, places the data bus output drivers in a high-impedance condition. If chip enable is low, data transfers are enabled in conjunction with the read and write inputs.

INTERRUPT REQUEST ($\overline{INTR}$)

Using this active low open-drain output, several conditions may be programmed to request an interrupt to the CPU. This pin will be inactive after power-on reset or a master reset command.

INTERRUPT ACKNOWLEDGE ($\overline{INTA}$)

This input is used to indicate that an interrupt request has been accepted by the CPU. When $\overline{INTA}$ goes low, the PKCC outputs an 8-bit address vector on D0–D7 corresponding to the highest priority interrupt currently active.

EXTERNAL INTERRUPT ($\overline{XINTR}$)

This is an active low external interrupt input to the PKCC interrupt priority receiver.

TRANSMITTER CLOCK (TxC)

The function of this input/output pin depends on bit 7 of the baud-rate control register (BRR7). If the external transmitter clock is selected (BRR7 = 0), this pin is an input for the transmitter clock. If the internal transmitter clock is selected (BRR7 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR5. The data is transmitted on the falling edge of TxC. This pin is an input after power on and after master reset or communications reset commands.

RECEIVER CLOCK (RxC)

The function of this input/output pin depends on BRR6. If the external receiver clock is selected (BRR6 = 0), this pin is an input for the receiver clock. If internal receiver clock is selected (BRR6 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR4. The received data is sampled on the rising edge of RxC. This pin is an input after power on and after master reset or communications reset commands.

TRANSMITTER DATA (TxD)

This output is the transmitted serial data; the least significant bit is transmitted first. This pin is high after power-on reset or a reset command that affects the transmitter.

RECEIVER DATA (RxD)

This input is the serial data input to the receiver. The least significant bit is received first.

CONNECTIONS FOR CRYSTAL (XTAL1, XTAL2/BRCLK)

The crystal connections provide an on-chip clock generator for the internal baud-rate generator and the keyboard interface logic. If an external clock is provided, use XTAL2 as the clock input. See Figures 10 and 11.

All timing parameters such as keyboard scan times, tone frequency, and baud rate assume a clock input at the specified BRG input frequency. If this frequency is different, the timing parameters will vary proportionately.

KEYBOARD ROW SCAN (KR0-KR2)

This output is decoded externally and selects one of eight rows.

KEYBOARD COLUMN SCAN (KC0-KC3)

This output is decoded externally and selects one of 16 columns.

KEY RETURN (KRET)

This input, when active high, indicates that the key being scanned is closed.

SHIFT KEY ($\overline{SHIFT}$)

This is the active low input from the shift key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs selects one of four possible codes from the internal key encoding ROM.

CONTROL KEY (CONTROL)

This is the active low input from the $\overline{CONTROL}$ key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs selects one of four possible codes from the internal key encoding ROM.

REPEAT KEY ($\overline{REPEAT}$)

This is the active low input from the $\overline{REPEAT}$ key which causes the key depression currently active to be repeated at a rate of approximately 15 times per second.

KEYBOARD CLOCK (KCLK)

This high frequency (approximately 400 kHz) output is used to scan capacitive keyboards.

3

KEY DETECT RESET (KDRES)

This output resets the analog detector before scanning a key. It is used for capacitive keyboards.

HYSTERESIS OUTPUT ($\overline{HYS}$)

This output is sent to the analog detector for capacitive keyboard applications. A low indicates the key currently scanned has been recognized on previous scan cycles.

SQUARE WAVE OUTPUT (TONE)

This output is used for tone generation.

FUNCTIONAL DESCRIPTION

The programmable keyboard and communications controller (PKCC) consists of six major sections. These are the transmitter, receiver, timing, operation control, keyboard encoder, and a priority encoded interrupt control unit. These sections communicate with each other via an internal data bus and an internal control bus. The internal bus interfaces to the microprocessor data bus via a bidirectional 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 KMR and CMR, the command decoder, and status registers KSR and CSR. Details of operating modes and status information are presented in **OPERATION.** The register addressing is specified in Table 1.

TIMING

The PKCC 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 baud rates, any of which can be selected for full-duplex operation. The external clock to the baud-rate generator can be applied directly to the XTAL2 input (see Figure 11) or can be generated internally by connecting a crystal across the XTAL1, XTAL2 input pins. The clock input is also utilized by the keyboard en .der section. Thus, a clock must be provided even if external transmitter and receiver clocks are used.

RECEIVER

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for break conditions, framing and parity errors, and loads an "assembled" character in the receive holding register for access by the CPU.

TRANSMITTER

The transmitter accepts parallel data loaded by the CPU into the transmit holding register and converts it to a serial bit stream framed by the start bit, calculated parity bit (if specified), and stop bit(s). The composite serial stream of data is transmitted on the TxD output pin.

KEYBOARD ENCODER

The keyboard encoder provides encoded scanning signals for a matrix keyboard. Key depressions are detected on the KRET input. The debounced and verified key codes (or matrix addresses) are loaded into the key holding register for access by the CPU. Figures 12 and 13 illustrate the PKCC interface to contact and capacitive keyboards, respectively.

TABLE 1 — REGISTER ADDRESSING

$\overline{CE}$	A2	A1	A0	R, $\overline{W}$	Function
1	X	X	X	X	Three-State Data Bus
0	0	0	0	$\overline{W}$	Reset Command
0	0	0	0	R	Read Interrupt Status Register (ISR)
0	0	0	1	R, $\overline{W}$	Read/Write Communications Mode Register (CMR)
0	0	1	0	$\overline{W}$	Write Transmit Holding Register (TxHR)
0	0	1	0	R	Read Receiver Holding Register (RxHR)
0	0	1	1	$\overline{W}$	Write Baud-Rate Mode Register (BRR)
0	0	1	1	R	Read Communications Status Register (CSR)
0	1	0	0	R, $\overline{W}$	Read/Write Interrupt Mask Register (IMR)
0	1	0	1	R, $\overline{W}$	Read/Write Keyboard Mode Register (KMR)
0	1	1	0	R	Read Keyboard Holding Register (KHR)
0	1	1	1	R	Read Keyboard Status Register (KSR)
0	1	1	1	$\overline{W}$	Miscellaneous Commands

X = Don't Care

FIGURE 12 — CONTACT KEYBOARD INTERFACE

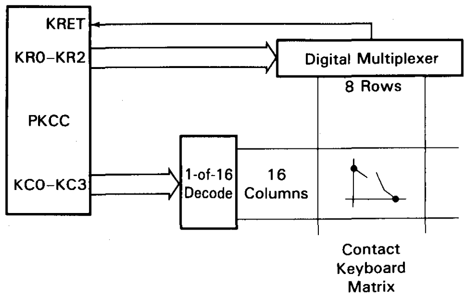
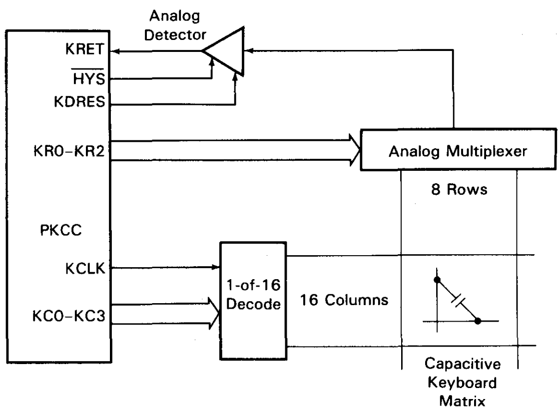


FIGURE 13 — CAPACITIVE KEYBOARD INTERFACE



INTERRUPT CONTROL

The interrupt controller unit contains a software programmable interrupt mask register which selectively enables status conditions from the keyboard encoder and communication controller to generate interrupts. The interrupts are priority encoded and individually generate an 8-bit vector which is output on the data bus in response to a CPU interrupt acknowledge on the $\overline{\text{INTA}}$ input pin.

OPERATION

KEYBOARD ENCODER

The keyboard is continuously scanned by KC0-KC3 and KR0-KR2 which are decoded externally to handle 128 possible keys (see Figures 12 and 13). KC0-KC3 select one of 16 columns and KR0-KR2 multiplex the eight row return lines into the KRET pin. Debouncing is accomplished by remembering a one state at the KRET pin when a key is being addressed and verifying it one scan later. Once the key is verified a key code is loaded into the keyboard data register (KDR). If the keyboard holding register is empty, the contents of the KDR will be transferred to the KHR immediately; if the KHR is full (i.e., the CPU has not read the previous key code), the transfer will be held off until the KHR is read. The data transfer to the KHR causes keyboard data (KRDY) to be set in the keyboard status register.

For capacitive keyboards, the high frequency output KCLK can be used to gate the column scan to the keyboard (see Figure 13). The key detector reset (KDRES) output resets the analog detector prior to scanning each key location. The output from the analog multiplexer is sensed and then latched in the analog detector. The $\overline{\text{HYS}}$ output controls the sense level. A zero will lower the sense level causing hysteresis, and a one will raise the sense level with no hysteresis.

The $\overline{\text{REPEAT}}$ input enables the keyboard logic to recognize any key repeatedly, 15 times per second. Additionally, certain keys can be programmed to repeat automatically if depressed for more than one-half second.

A square wave is output on the TONE pin when the CPU issues a ring tone command to the PKCC.

KEYBOARD MODE REGISTER

Operating modes are selected by programming the keyboard mode register (KMR), whose format is illustrated in Figure 14.

Bit KMR7 is used for testing the device. For normal operation, this bit should always be written to a zero.

Bits KMR6–KMR5 select the rollover modes for keyboard processing:

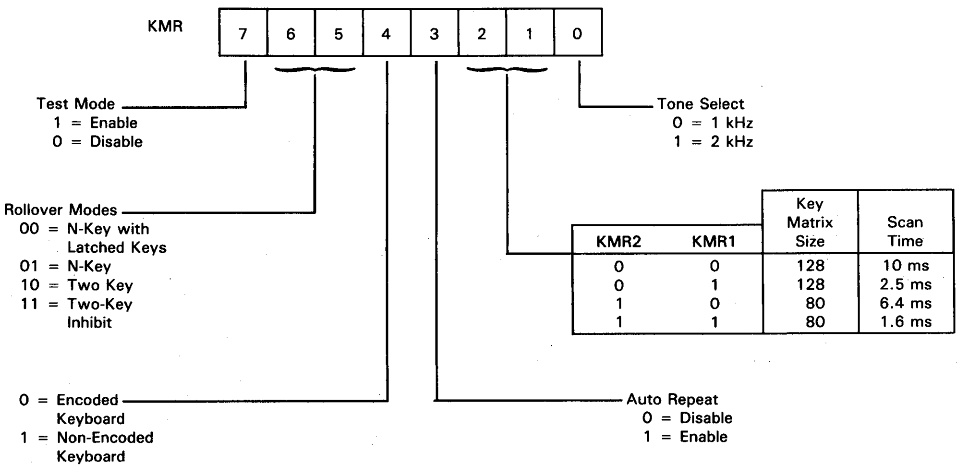
N-Key Rollover. In this mode, the code corresponding to each key depression is loaded into the KDR as soon as that key is debounced, independent of the release of other keys. Two or more closures occurring within one scan cycle are considered to be simultaneous which will set keyboard error in the keyboard status register (KSR1). As soon as the keyboard holding register is empty the code in the KDR is transferred to the KHR and the KRDY status bit is set (KSR0).

N-Key Rollover with Latched Keys. This mode is the same as regular N-key rollover, except that the keys which are assigned to row 0 of the keyboard matrix (KR2-KR0 = 000) produce a code both when depressed and when released. The codes are independent of the states of the inputs as $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$. If one or more of the latched keys are depressed when the keyboard is enabled (after a keyboard reset), the corresponding codes will be sent out as the keys are scanned and debounced. Note that simultaneous latched keys will not set KERR (KSR1) and that latched keys will not be auto-repeat and will not be affected by the $\overline{\text{REPEAT}}$ input.

Two-Key Rollover. The first key code is loaded into the KDR immediately and the second code is loaded only after the first key is released. Simultaneous keys will set KERR (KSR1). If three or more keys remain closed at any given time, the KERR bit will also be set. All keys must then be released before the next KRET will be processed.

Two-Key Inhibit. All keys must be released between keystrokes; otherwise, KERR (KSR1) will be set.

FIGURE 14 — KEYBOARD MODE REGISTER FORMAT



KMR2	KMR1	Key Matrix Size	Scan Time
0	0	128	10 ms
0	1	128	2.5 ms
1	0	80	6.4 ms
1	1	80	1.6 ms

Bit KMR4 specifies the key encoding mode. Each key is assigned four 8-bit codes, corresponding to the states of the $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs. If the encoded mode is programmed, the row/column address of the detected key is used to load one of the four key codes into the KDR. See Table 2 for key code assignments. If the non-encoded mode is programmed, the row/column address is loaded directly into the KDR with the following format:

KDR

7	6	5	4	3	2	1	0

Bits 2–0: KR2, KR1, KR0

Bits 6–3: KC3, KC2, KC1, KC0

Bit 7: "0" for momentary keys
"1" for latched keys release
"0" for latched keys depress

TABLE 2 — STANDARD KEY CODES (HEX)

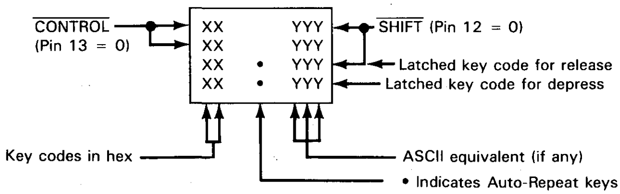
Column (KC3–KC0)	Row (KR3–KR0) 0	1	2	3	4	5	6	7
0	E0	C0	1B ESC	09 HT	1F US	1A SUB	30 0	2B +
	F0	D0	1B ESC	09 HT	1F US	1A SUB	30 0	3B ;
	E0	C0	1B ESC	09 • HT	1F US	5A Z	30 0	2B +
	F0	D0	1B ESC	09 • HT	1F US	7A z	30 0	3B ;
1	E1	C1	21 !	11 DC1	01 SOH	18 CAN	3D =	2A •
	F1	D1	31 1	11 DC1	01 SOH	18 CAN	2D –	3A :
	E1	C1	21 !	51 Q	41 A	58 X	3D • =	2A •
	F1	D1	31 1	71 q	61 a	78 x	2D • –	3A :
2	E2	C2	22 "	17 ETB	13 DC3	03 ETX	1E RS	1F US
	F2	D2	32 2	17 ETB	13 DC3	03 ETX	1E RS	1F US
	E2	C2	22 "	57 W	53 S	43 C	7E ~	7F • DEL
	F2	D2	32 2	77 w	73 s	63 c	5E ↑	5F • _
3	E3	C3	23 #	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	F3	D3	33 3	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	E3	C3	23 #	45 E	44 D	56 V	7C ¦	7B {
	F3	D3	33 3	65 e	64 d	76 v	5C \	5B [
4	E4	C4	24 $	12 DC2	06 ACK	02 STX	08 BS	1D GS
	F4	D4	34 4	12 DC2	06 ACK	02 STX	08 BS	1D GS
	E4	C4	24 $	52 R	46 F	42 B	08 • BS	7D }
	F4	D4	34 4	72 r	66 f	62 b	08 • BS	5D]
5	E5	C5	25 %	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	F5	D5	35 5	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	E5	C5	25 %	54 T	47 G	4E N	50 P	08 BS
	F5	D5	35 5	74 t	67 g	6E n	70 p	08 • BS
6	E6	C6	26 &	19 EM	08 BS	0D CR	00 NUL	09 HT
	F6	D6	36 6	19 EM	08 BS	0D CR	00 NUL	09 HT
	E6	C6	26 &	59 Y	48 H	4D M	60 `	09 • HT
	F6	D6	36 6	79 y	68 h	6D m	40 @	09 • HT

Continued

TABLE 2 — STANDARD KEY CODES (HEX) (Continued)

Column (KC3–KC0)	Row (KR3–KR0) 0	1	2	3	4	5	6	7
7	E7	C7	27 '	15 NAK	0A LF	3C <	7F DEL	20 SP
	F7	D7	37 7	15 NAK	0A LF	2C ,	7F DEL	20 SP
	E7	C7	27 '	55 U	4A J	3C <	7F DEL	20 • SP
	F7	D7	37 7	75 u	6A j	2C ,	7F DEL	20 • SP
8	E8	C8	28 (	09 HT	0B VT	3E >	0A LF	0B VT
	F8	D8	38 8	09 HT	0B VT	2E .	0A LF	0B VT
	E8	C8	28 (	49 I	4B K	3E >	0A LF	0B • VT
	F8	D8	38 8	69 i	6B k	2E .	0A LF	0B • VT
9	E9	C9	29)	0F SI	0C FF	3F ?	0D CR	0A LF
	F9	D9	39 9	0F SI	0C FF	2F /	0D CR	0A LF
	E9	C9	29)	4F O	4C L	3F ?	0D CR	0A • LF
	F9	D9	39 9	6F o	6C l	2F /	0D CR	0A • LF
A	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
B	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
C	EC	CC	39 9	36 6	33 3	BF	A2	A8
	FC	DC	39 9	36 6	33 3	AF	B2	B8
	EC	CC	39 9	36 6	33 3	9F	A2	A8
	FC	DC	39 9	36 6	33 3	8F	B2	B8
D	ED	CD	90	93	82	95	A3	A9
	FD	DD	90	93	82	95	B3	B9
	ED	CD	90	93	82 •	95	A3 •	A9 •
	FD	DD	90	93	82 •	95	B3 •	B9 •
E	EE	CE	91	80	84	81	A4	AA
	FE	DE	91	80	84	81	B4	BA
	EE	CE	91	80 •	84	81 •	A4 •	AA •
	FE	DE	91	80 •	84	81 •	B4 •	BA •
F	EF	CF	92	94	83	96	A5	AB
	FF	DF	92	94	83	96	B5	BB
	EF	CF	92	94	83 •	96	A5 •	AB •
	FF	DF	92	94	83 •	96	B5 •	BB •

This row contains the latched keys when that mode is selected (KMR6, KMR5 = 00).



Bit KMR3 enables the auto-repeat mode. In this mode, if a key that is programmed for auto-repeat is depressed for longer than one-half second, the key code will be loaded into the KDR approximately 15 times per second until that key is released. Only the non-control codes will auto-repeat, i.e., $\overline{\text{CONTROL}} = 1$. Table 2 specifies the auto-repeat keys.

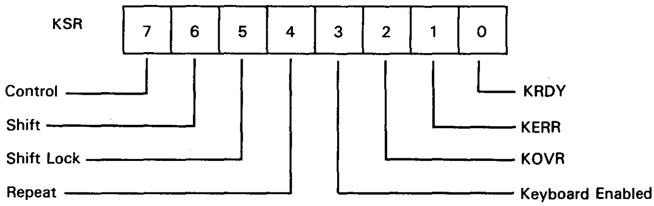
KMR2 and KMR1 select the key matrix size and debounce time (scan rate). The keyboard row outputs (KR2, KR1, KR0) always scan from 0 to 7. The column outputs (KC3, KC2, KC1, KC0) scan from 0 to 15 for a 128-key matrix and from 0 to 9 for an 80-key matrix.

KMR0 selects between a 1 kHz and 2 kHz frequency to be output on the TONE pin in response to a ring tone command.

KEYBOARD STATUS REGISTER

The keyboard status register (KSR) provides operational feedback to the CPU. Its format is illustrated in Figure 15.

FIGURE 15 — KEYBOARD STATUS REGISTER FORMAT



KSR7, KSR6, and KSR4 reflect the state of the inputs at the corresponding pins. $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are latched at the time the key is accepted. As the verified codes are loaded into the KDR, the corresponding states of $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are loaded into the KSR. $\overline{\text{REPEAT}}$ is updated on every matrix example. The status bits are the complements of the input levels.

KSR5 reflects the state of the internal shift lock flag which is controlled by the set/reset shift lock commands.

KSR3 indicates that the keyboard controller is enabled. It is controlled by the set/clear keyboard enable command.

Keyboard overrun (KSR2) is set when both the KHR and KDR are full and a third key is validated. The original content of the KHR is preserved and the content of the KDR is overwritten with the new key code. This bit can be specified (by IMR1) to generate an interrupt and is cleared by the reset command with D2 = 1.

Keyboard error (KR1) is set when the operator depresses more keys than are allowed in the selected rollover mode, or when keys are depressed simultaneously (within one scan cycle). This bit can be specified (by IMR3) to generate an interrupt and is cleared by the reset command with D1 = 1.

Keyboard data ready (KSRO) is set when the key code or address is transferred from the KDR to the KHR. This bit can be specified (by IMR2) to generate an interrupt. It is cleared when the CPU reads the KHR.

COMMUNICATIONS CONTROLLER

The communications controller section of the PKCC comprises a full duplex asynchronous receiver/transmitter (UART) with a baud-rate generator. Registers associated with these elements are the communications mode register (CMR), the baud-rate control register (BRR), and the communications status register (CSR).

RECEIVER

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high-to-low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again after a delay of one half of the bit time. If RxD is then high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RxHR) and the RxRDY bit in the CSR is set to a one. If the character length is less than eight bits, the most significant unused bits in the RxHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e., framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the space is interpreted as a start bit.

The parity error, framing error, and overrun error (if any) are strobed into the CSR at the received character boundary. If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be transferred to the RxHR and the received break bit in the CSR is set to one (RxRDY is not set when a break is received). The RxD input must return to a high condition for one bit time before a search for the next start bit begins.

TRANSMITTER

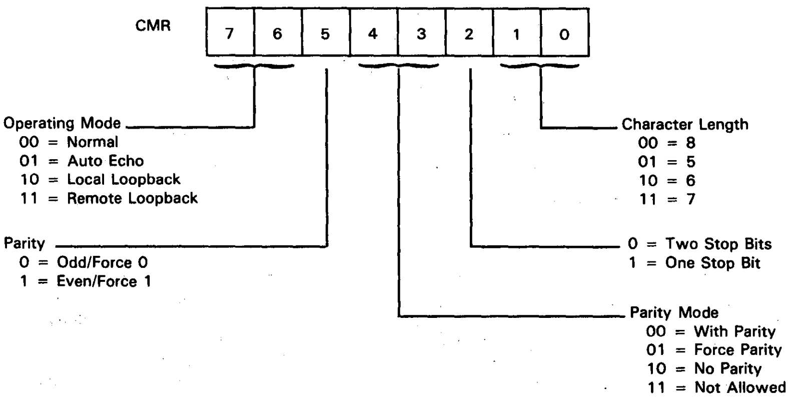
The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the transmit holding register (TxHR), the TxD output remains high and the TxEMT bit in the CSR will be set to one. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxHR. The transmitter can be forced to send a continuous low condition by a transmit break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out.

COMMUNICATION MODE REGISTER

Figure 16 illustrates the bit format of the CMR, which controls the operational mode of the communications controller and the character parameters.

FIGURE 16 — COMMUNICATIONS MODE REGISTER FORMAT



Bits CMR1-CMR0 select a character length of five, six, seven, or eight bits. The character length does not include the parity, start, or stop bits.

CMR2 selects the transmitted character framing as one or two stop bits. The receiver always checks for one stop bit.

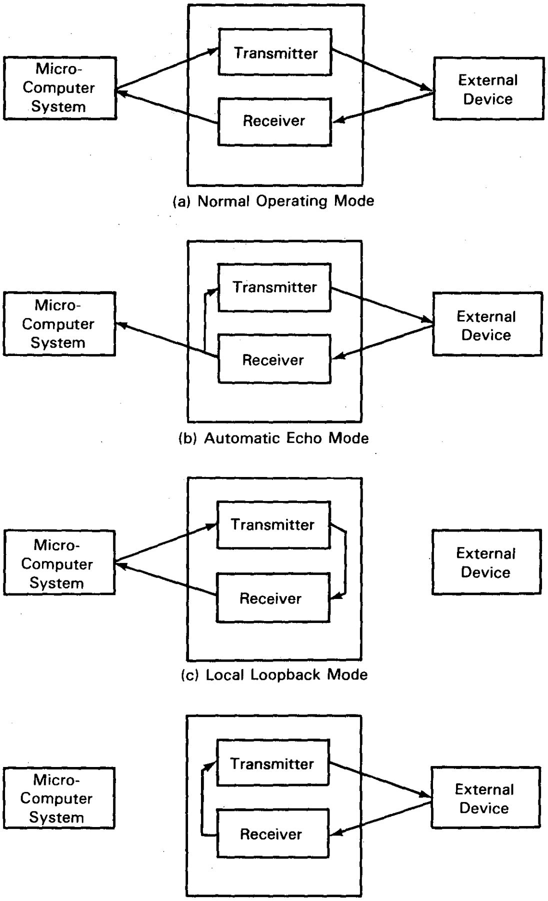
The parity format is selected by bits CMR4 and CMR3. If parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. CMR5 selects odd or even parity and determines the polarity of the parity bit in the force parity mode.

The bits in the mode register affecting character assembly and disassembly (CMR5-CMR0) can be changed dynamically and affect the characters currently being assembled in RxSR and transmitted by TxSR. To affect assembly of a received character, the CMR must be updated within n-1 bit times of the receipt of that character's start bit. To affect a transmitted character, the CMR must be updated within n-1 bit times of transmitting that character's start bit (n = the smaller of the new and old character lengths).

The UART can operate in one of four modes, as illustrated in Figure 17. The operating modes are selected by bits CMR7 and CMR6, which should only be changed when both the transmitter and receiver are operating independently. CMR7 – CMR6 = 01 places the UART in the automatic-echo mode, which automatically retransmits the received data. The following conditions are true while in automatic-echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. Status bit TxRDY is not set. TxEMT operates normally.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Only the first character of a break condition is echoed; the TxD output will go high until the next received character is assembled.
7. CPU-to-receiver communication continues normally, but the CPU-to-transmitter link is disabled.

FIGURE 17 — OPERATING MODES OF THE MC2671 UART



(a) Normal Operating Mode

(b) Automatic Echo Mode

(c) Local Loopback Mode

(d) Remote Loopback Mode

Two diagnostic modes can also be configured. In local loopback mode (CMR7 – CMR6 = 10):

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU-to-transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode (CMR7 – CMR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. No data is sent to the local CPU, but the error status conditions (parity and framing) are set if required.
4. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.

BAUD-RATE CONTROL REGISTER

The baud-rate control register (BRR) controls the frequency generated by the baud-rate generator (BRG) and the clock source used by the receiver and transmitter. Its format is illustrated in Figure 18.

BRR3-BRR0 select one of sixteen frequencies to be generated by the BRG. See Table 3.

BRR7 and BRR6 select the source of the transmit and receive clocks. If external clocks are chosen (BRR7 = 0 or BRR6 = 0), then the clock rate factor is determined by BRR5 and BRR4. The external clock input(s) should be the desired baud rate multiplied by the clock rate factor.

FIGURE 18 — BAUD-RATE CONTROL REGISTER FORMAT

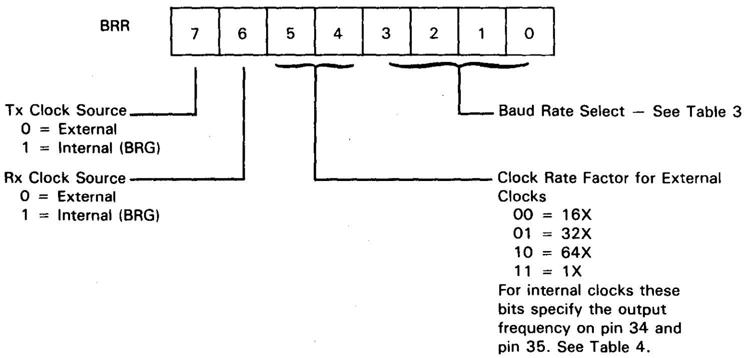


TABLE 3 — BAUD RATE GENERATOR CHARACTERISTICS
(BRCLK = 4.9152 MHz)

BRR3-0	Baud Rate	Actual Frequency 16X Clock	Percent Error	Divisor
0000	50	0.8 kHz	–	6144
0001	110	1.7598	-0.01	2793
0010	134.5	2.152	–	2284
0011	150	2.4	–	2048
0100	200	3.2	–	1536
0101	300	4.8	–	1024
0110	600	9.6	–	512
0111	1050	16.8329	+0.20	292
1000	1200	19.2	–	256
1001	1800	28.7438	-0.20	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

If internal clock(s) are specified, (BRR7 = 1 or BRR6 = 1), the clock is supplied by the internal baud-rate generator at the selected baud rate. The clock rate factor for internally generated clocks is always 16. Pins 35 and 34 become outputs for transmit or receive clocks, respectively. See Table 4 for the description and selection of these outputs.

COMMUNICATIONS STATUS REGISTER

Figure 19 illustrates the bit format of the communications status register (CSR), which provides UART status to the CPU.

Receiver ready (CSR0) indicates that a received character is assembled and transferred to the RxHR and is ready to be read by the CPU. This bit can be specified (by IMR0) to generate an interrupt and is reset by reading the RxHR.

Transmitter ready (CSR1) indicates that the TxHR is empty and ready to be loaded with character. This bit will be cleared when the TxHR is loaded and has not yet transferred the character to the transmit shift register (TxSR). TxRDY is reset when the transmitter is disabled. It will be set when the transmitter is enabled, provided that no data was loaded into the TxHR during the time the transmitter was disabled. This bit can be specified (by IMR7) to generate an interrupt.

Transmitter empty (CSR2) indicates that the transmitter has underrun, i.e., both the TxHR and TxSR are empty. This bit can only be set after transmission of at least one character, and is cleared when the TxHR is loaded by the CPU. TxEMT is reset when the transmitter is disabled. This bit can be specified (by IMR6) to generate an interrupt.

CSR3 will be set when the PKCC receives a command to transmit a break. This bit will be cleared after the break is completed.

Received break (CSR4) indicates that an all zero character of the programmed length has been received without a stop bit. Breaks originating in the middle of a received character can be detected. This bit is cleared when RxD returns to a high state for at least one bit time.

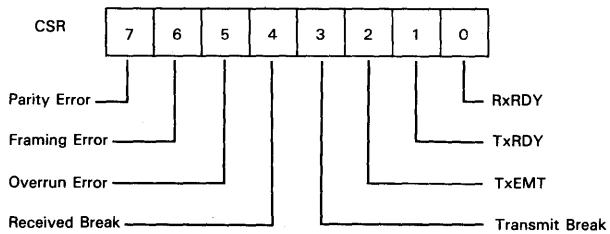
Receiver overrun (CSR5) indicates that the previous character in the RxHR has not been read by the CPU and that a new character has been loaded into the RxHR. This bit is cleared by a reset command with D3 = 1.

TABLE 4 — BAUD-RATE CONTROL REGISTER

BRR7–BRR4	Clock Source TxC	Clock Source RxC	Pin Functions Pin 34	Pin Functions Pin 35	BRR3–BRR0 Baud Rate Selection
00**	E	E	TxC	RxC	The baud rates are listed in Table 3.
01**	E	I	TxC	1X	
10**	I	E	16X	RxC	
1100	I	I	1X	1X	
1101	I	I	1X	16X	
1110	I	I	16X	1X	
1111	I	I	16X	16X	

NOTES
1. ** = Clock rate factor for external clocks: 00 = 16X, 01 = 32X, 10 = 64X, 11 = 1X
2. E = External clock.
3. I = Internal clock (BRG).
4. 1X and 16X are clock outputs at 1 or 16 times the actual baud rate. For receive, the 1X output is the actual data sample clock.
5. BRR7–BRR6 = 01 or 10 not permitted in automatic echo or remote loopback modes unless BRR5–BRR4 = 00.

FIGURE 19 — COMMUNICATIONS STATUS REGISTER FORMAT



Framing error (CSR6) indicates that the stop bit has not been detected. The stop bit check is made in the middle of the first stop bit position. This bit is cleared by a reset command with D3 = 1.

Parity error (CSR7) indicates that a character was received with incorrect parity when 'with parity' is enabled. This bit is cleared by a reset command with D3 = 1.

INTERRUPT CONTROLLER

The MC2671 contains a maskable interrupt status register (ISR) which can be enabled to generate an active low interrupt request on the INTR output. The eight interrupt conditions in the ISR are individually enabled by writing a one into the corresponding bit of the interrupt mask register (IMR).

Each of the interrupt conditions is assigned a priority and a vector. When an enabled ISR bit is set, the MC2671 asserts the INTR output. If the CPU activates the INTA input, the MC2671 responds by placing the corresponding 8-bit on the data bus (D7-D0). If multiple interrupts are pending, the vector corresponds to the condition with the highest priority. The interrupt will persist until all pending interrupt conditions are cleared.

The ISR can also be polled by reading at address A2 – A0 = 000. All pending interrupt conditions which are enabled by the IMR will be read independent of priority.

The bit assignments of the ISR and IMR and corresponding vectors and priorities are listed in Table 5.

COMMANDS

In addition to the control exercised by programming of the PKCC control registers, several functions can be performed by executing command operations. There are two classes of commands which are initiated by writing to the MC2671 at address A2 – A0 = 000 (reset command) and address A2 – A0 = 111 (miscellaneous commands). Individual commands are specified by the bit pattern of the data bus (D7 – D0).

RESET COMMANDS

The reset command bit format is illustrated in Figure 20 and the detailed command descriptions are given in Table 6.

A reset command with D7 – D0 = 111XXXX1 is a master reset for the MC2671. This command must be given following a power-on condition to release the internal power-on reset latch which deactivates the MC2671 on power up.

MISCELLANEOUS COMMANDS

The miscellaneous command format is illustrated in Figure 21.

TABLE 5 — INTERRUPT MASK REGISTER (IMR) AND INTERRUPT STATUS REGISTER (ISR)

Bit in IMR/ISR	Interrupt Condition	Priority	Vector on D7-D0 Binary	Vector on D7-D0 Hex	Condition Reset by:
IMR0/ISR0	RxRDY	1	11001111	CF	Read RxHR
IMR1/ISR1	KOVR	2	11010111	D7	Reset CMD (D2 = 1)
IMR2/ISR2	KRDY	3	11011111	DF	Read KHR
IMR3/ISR3	KERR	4	11100111	E7	Reset CMD (D1 = 1)
IMR4/ISR4	XINT[1]	5	11101111	EF	External
IMR5/ISR5	ΔBREAK[2]	6	11110111	F7	Reset CMD (D4 = 1)
IMR6/ISR6	TxEMT	7	11000111	C7	Load TxHR
IMR7/ISR7	TxRDY	8	11000111	C7	Load TxHR

NOTES:
1. XINT is an input from an external interrupt source, active low (pin 21).
2. ΔBREAK refers to the change of a received break condition.

FIGURE 20 — RESET COMMAND FORMAT

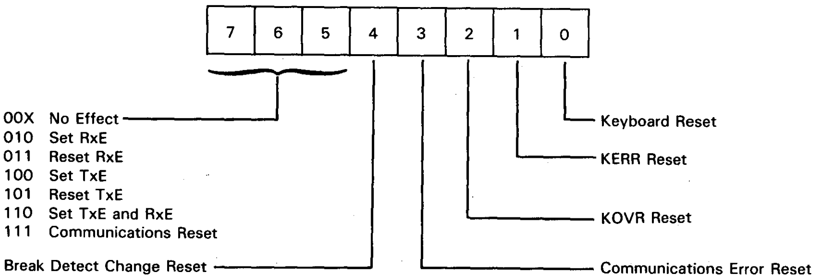
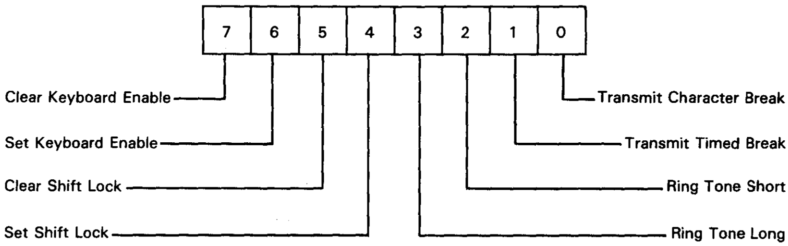


TABLE 6 — RESET COMMAND DESCRIPTION

Command	Resets	Comments
Keyboard Reset	KMR7-KMR0 KSR5, KSR2-KSR0 IMR3-IMR1	The keyboard controller is reset, ignoring the input at KRET.
KERR Reset	KSR1	Keyboard error status bit reset.
KOVR Reset	KSR2	Keyboard overrun status bit reset.
Communications Error Reset	CSR7-CSR5	Resets the receiver overrun, parity, and framing error status bits.
Break Detect Change Reset	ISR5	Resets the break detect change bit in the interrupt status register.
Set RxE	See note	Enables receiver operation.
Reset RxE	CSR7-CSR4, CSR0 See note	Disables the receiver.
Set TxE	See note	Enables transmitter operation
Reset TxE	CSR3-CSR1 See note	Disables the transmitter. Sets the TxD output to a one after transmitting the character in TxSR.
Communications Reset	CMR, CSR, BRR, TxE, RxE, IMR7-IMR5, IMR0	Resets the communication controller. The RxD input is ignored and the TxD output is set to a one.
Master Reset	CMR, CSR, BRR, TxE, RxE, KMR, KSR5, KSR3-KSR0, IMR7-IMR0. Releases the internally latched power-on reset.	Resets the keyboard and communication controllers. Inputs at KRET and RxD are ignored and the TxD output is set to a one.

NOTE: Command does not affect the CMR or the BRR.

FIGURE 21 — MISCELLANEOUS COMMANDS FORMAT



The transmit break commands force a break (steady low output) on the TxD pin immediately or after the character in the TxSR (if any) is transmitted. A timed break lasts for approximately 200 milliseconds, and a character break lasts for one character time including parity and stop bit time. In either case, TxRDY (CSR1) will be set at the beginning of the break which can be extended indefinitely (by 200 milliseconds or one character time increments) by reasserting the command in response to TxRDY. Note that these commands reset TxRDY. When a transmit break command is asserted, CSR3 will be set. This bit will be cleared after the break is completed.

The ring tone commands cause the tone generator to output a square wave on the TONE output. The tone durations are specified by the commands.

Ring tone short = 25 milliseconds
Ring tone long = 100 milliseconds

The tone frequency is either 1 kHz or 2 kHz, as specified by KMRO.

The set/clear shift lock commands control the state of the internal shift lock flip flop. When shift lock is set the keyboard controller encodes all key depressions as if the $\overline{\text{SHIFT}}$ input was asserted. The state of the shift lock flip flop is reflected in KSR5.

The set keyboard enable command enables the keyboard controller and sets KSR3 in the keyboard status register. The clear keyboard enable command resets KSR3 and disables key processing at the KRET input. The keyboard controller is not reset by this command, and the current state of the keyboard (key depressions and latched key states) is preserved internally. When the keyboard is subsequently enabled, key processing resumes, old and new keys are debounced, and latched keys are encoded if there has been a change in their state.

MASK PROGRAMMABLE OPTIONS

Characteristics of certain portions of the PKCC are internally programmed by means of a ready-only memory.

The items which can be programmed are:

- Key codes
- Auto-repeat keys
- Scan times, tone frequency, and tone duration
- Baud rates
- Interrupt vectors

Consult your local Motorola representative for costs, minimum quantities, and data submission requirements for customized versions of the PKCC.

MOTOROLA

MC2672

Advance Information

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC)

The MC2672 programmable video timing controller (PVTC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC. Applications include CRT terminals, word-processing systems, small business computers, and home computers.

- 4 MHz Character Rate
- Up to 256 Characters Per Row
- 1 to 16 Raster Lines Per Character Row
- Up to 128 Character Rows Per Frame
- Programmable Horizontal and Vertical Sync Generators
- Interlaced or Non-Interlaced Operation
- Up to 16K RAM Addressing for Multiple Page Operation
- Automatic Wraparound of RAM
- Addressable, Incrementable, and Readable Cursor
- Programmable Cursor Size, Position, and Blink
- Split Screen and Horizontal Scroll Capability
- Light Pen Register
- Selectable Buffer Interface Modes
- Dynamic RAM Refresh
- Completely TTL Compatible
- Single +5-Volt Power Supply
- Power-On Reset Circuit

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

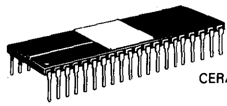
Package Type	Frequency	Order Number
Plastic P Suffix	2.7 MHz 4.0 MHz	MC2672A3P MC2672A4P
Ceramic L Suffix	2.7 MHz 4.0 MHz	MC2672A3L MC2672A4L
Cerdip S Suffix	2.7 MHz 4.0 MHz	MC2672A3S MC2672A4S

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

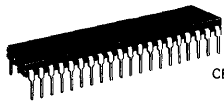
HMOS

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

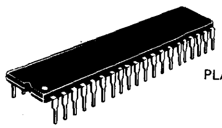
PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC)



L SUFFIX
CERAMIC PACKAGE
CASE 715



S SUFFIX
CERDIP PACKAGE
CASE 734

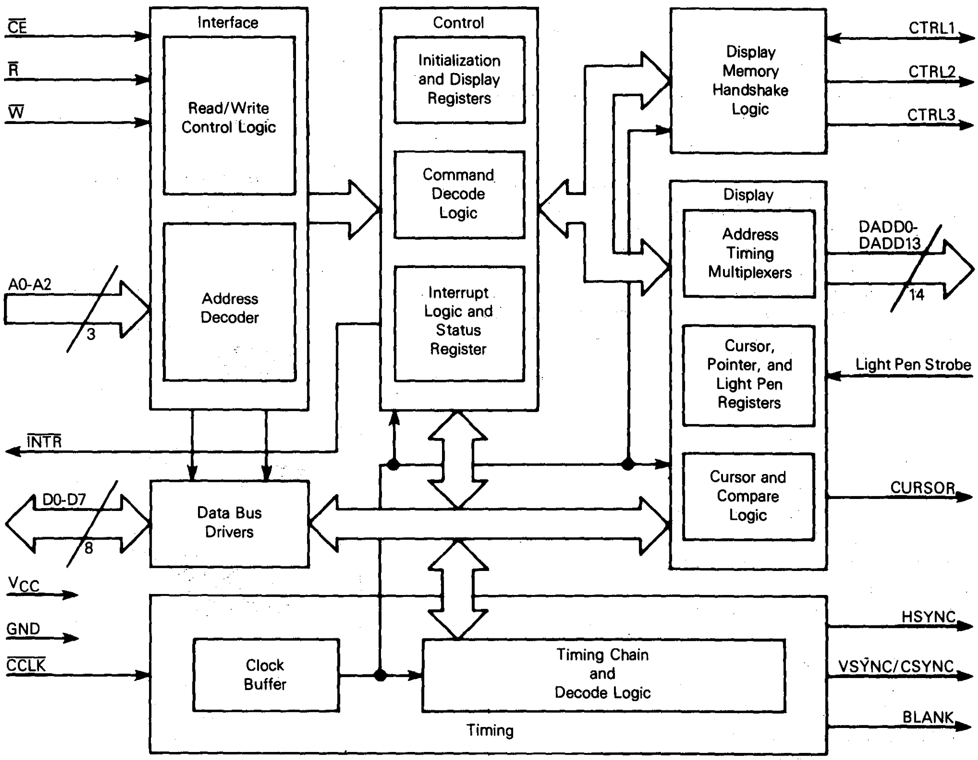


P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
$\overline{R}$	1	40	V_{CC}
$\overline{CE}$	2	39	A2
$\overline{W}$	3	38	A1
CTRL1	4	37	A0
CTRL2	5	36	LPS
CTRL3	6	35	$\overline{INTR}$
CURSOR	7	34	DADD0
D0	8	33	DADD1
D1	9	32	DADD2
D2	10	31	DADD3/LI
D3	11	30	DADD4/LA0
D4	12	29	DADD5/LA1
D5	13	28	DADD6/LA2
D6	14	27	DADD7/LA3
D7	15	26	DADD8/LNZ
$\overline{CCLK}$	16	25	DADD9/LPL
BLANK	17	24	DADD10/UL
VSYNC/CSYNC	18	23	DADD11/BLINK
HSYNC	19	22	DADD12/ODD
GND	20	21	DADD13/LL

BLOCK DIAGRAM



ABSOLUTE 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 Package	θ_{JA}	100	°C/W
Ceramic Package	θ_{JA}	50	°C/W
Cerdip Package	θ_{JA}	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. 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:

$T_J = T_A + (P_D \bullet \theta_{JA})$ (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 (T_A = 0°C to 70°C, V_{CC} = 5.0 V ±5%)

Parameter	Symbol	Min	Max	Unit
Input Low Voltage	V_{IL}	−0.3	0.8	V
Input High Voltage	V_{IH}	2.0	V_{CC}	V
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	—	0.4	V
Output High Voltage (Except $\overline{INTR}$ Output) I_{Load} = −100 μA	V_{OH}	2.4	—	V
Input Leakage Current V_{in} = 0 to V_{CC}	I_{in}	−10	10	μA
Hi-Z (Offstate) Input Current V_{in} = 0.4 to 2.4 V	I_{TSI}	−10	10	μA
$\overline{INTR}$ Open-Drain Output Leakage Current V_{OH} = 2.4 V_{CC}	I_{LOH}	—	10	μA
Internal Power Dissipation	P_{INT}	—	800	mW

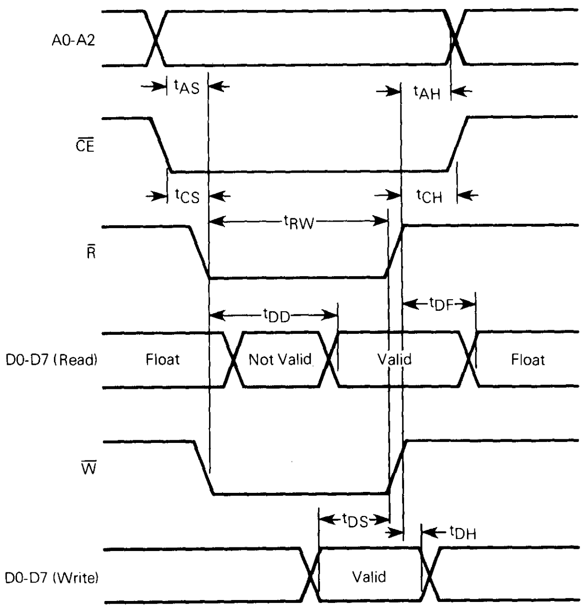
AC ELECTRICAL CHARACTERISTICS — BUS TIMING ($T_A = 0°$ to $70°C$, $V_{CC} = 5.0\ V \pm 5\%$, See Note 1)

Parameter		Symbol	MC2672A3 Min	MC2672A3 Max	MC2672A4 Min	MC2672A4 Max	Unit
A0-A2 Setup Time to $\overline{W}$, $\overline{R}$ Low		t_{AS}	30	–	30	–	ns
A0-A2 Hold Time from $\overline{W}$, $\overline{R}$ High		t_{AH}	0	–	0	–	ns
$\overline{CE}$ Setup Time to $\overline{W}$, $\overline{R}$ Low		t_{CS}	0	–	0	–	ns
$\overline{CE}$ Hold Time from $\overline{W}$, $\overline{R}$ High		t_{CH}	0	–	0	–	ns
$\overline{W}$, $\overline{R}$ Pulse Width		t_{RW}	250	–	250	–	ns
Data Valid after $\overline{R}$ Low		t_{DD}	–	200	–	200	ns
Data Bus Floating after $\overline{R}$ High		t_{DF}	–	100	–	100	ns
Data Setup Time to $\overline{W}$ High		t_{DS}	150	–	150	–	ns
Data Hold Time from $\overline{W}$ High		t_{DH}	10	–	5	–	ns
High Time from $\overline{CE}$ to $\overline{CE}$ (see Note 2)	Consecutive Commands	t_{CC}	600	–	600	–	ns
	Other Commands		300	–	300	–	ns

NOTES:

1. Timing is illustrated and specified referenced to $\overline{W}$ and $\overline{R}$ inputs. Device may also be operated with $\overline{CE}$ as the "strobing" input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.
2. This specification requires that the $\overline{CE}$ input be negated (high) between read and/or write cycles.

BUS TIMING DIAGRAM

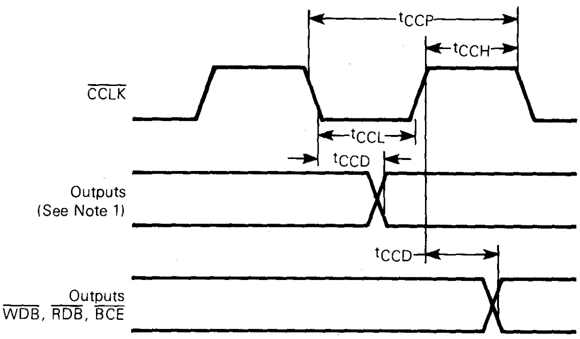


AC ELECTRICAL CHARACTERISTICS — CHARACTER CLOCK TIMING (T_A = 0°C to 70°C, V_{CC} = 5.0 V ±5%, See Note 1)

Parameter	Symbol	MC2672A3 Min	MC2672A3 Max	MC2672A4 Min	MC2672A4 Max	Unit
$\overline{CCLK}$ Period	t_{CCP}	370	–	250	–	ns
$\overline{CCLK}$ High Time	t_{CCH}	125	–	100	–	ns
$\overline{CCLK}$ Low Time	t_{CCL}	125	–	100	–	ns
Output Delay Time from $\overline{CCLK}$ Edge	t_{CCD}					
DADD0-DADD13, $\overline{BCE}$, $\overline{WDB}$, $\overline{RDB}$, MBC		40	175	40	150	ns
BLANK, HSYNC, VSYNC/CSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BACK}$*		40	225	40	200	

*$\overline{BCE}$, $\overline{WDB}$, and $\overline{RDB}$ delays track each other within 10 nanoseconds. Also, these output delays will tend to follow the direction (minimum/maximum) of DADD0-DADD13 delays.

CHARACTER CLOCK TIMING DIAGRAM



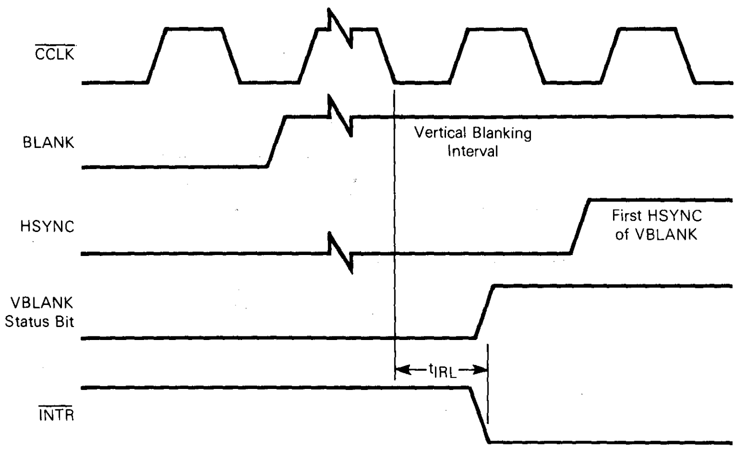
NOTES:
1. DADD0-DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, BACK.
2. $\overline{BCE}$ changes state on both $\overline{CCLK}$ edges.

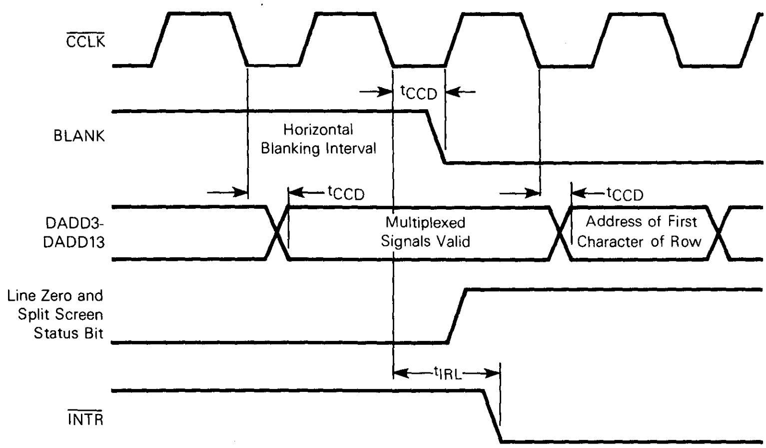
AC ELECTRICAL CHARACTERISTICS — OTHER TIMINGS (T_A=0°C to 70°C, V_{CC}=5.0 V ±5%)

Parameter	Symbol	MC2672A3 Min	MC2672A3 Max	MC2672A4 Min	MC2672A4 Max	Unit
READY/RDFLG Low from $\overline{\text{W}}$ HIGH*	t_{RDL}	–	t_{CCP}+30	–	t_{CCP}+30	ns
$\overline{\text{BACK}}$ High from $\overline{\text{PGREQ}}$ Low	t_{BAK}	–	225	–	200	ns
$\overline{\text{BEXT}}$ High from $\overline{\text{PBREQ}}$ High	t_{BXT}	–	225	–	200	ns
Light Pen Strobe Setup Time to $\overline{\text{CCLK}}$ Low	t_{LPS}	120	–	120	–	ns
Light Pen Strobe Hold Time from $\overline{\text{CCLK}}$ Low	t_{LPH}	–10	–	–10	–	ns
$\overline{\text{INTR}}$ Low from $\overline{\text{CCLK}}$ Low	t_{IRL}	–	225	–	200	ns
$\overline{\text{INTR}}$ High from $\overline{\text{W}}$, $\overline{\text{R}}$ High*	t_{IRH}	–	600	–	600	ns

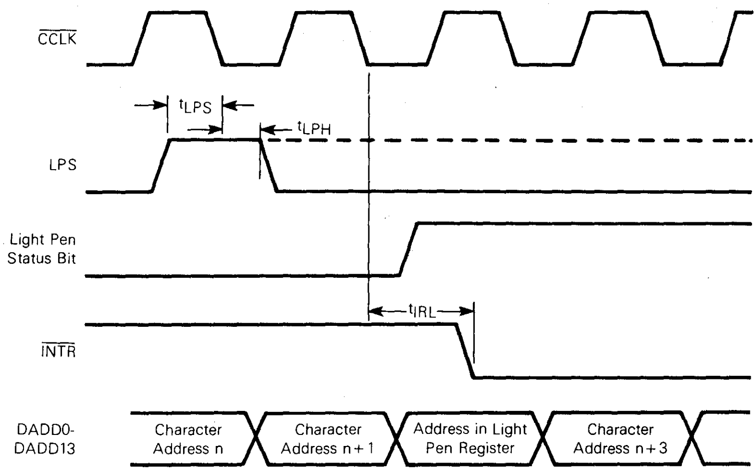
* Timing is illustrated and specified referenced to $\overline{\text{W}}$ and $\overline{\text{R}}$ inputs. Device may also be operated with $\overline{\text{CE}}$ as the "strobing" input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{\text{CE}}$.

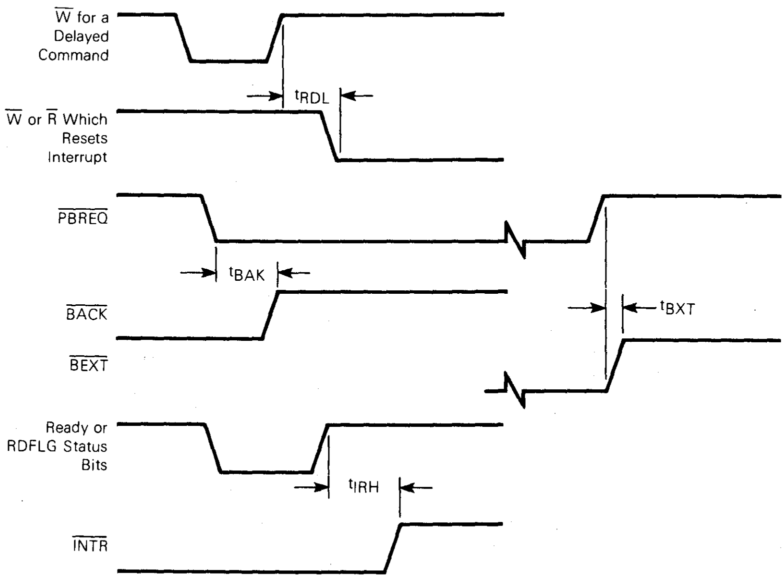
OTHER TIMING DIAGRAMS



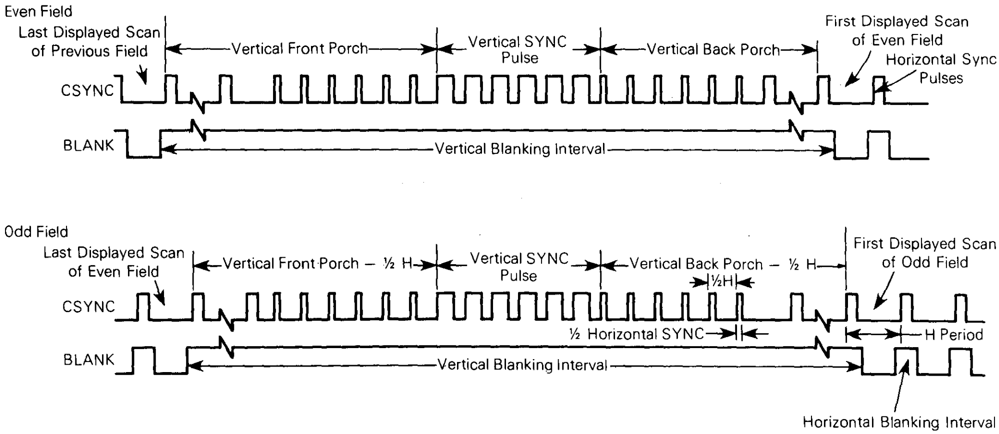


OTHER TIMING DIAGRAMS (Continued)





COMPOSITE SYNC TIMING DIAGRAM



NOTES:
1. In non-interlaced operation the even field is repeated continuously, and the odd field is not.
2. In interlaced operation the even field alternates with the odd field.

SIGNAL DESCRIPTION

The input and output signals for the PVTC are described in the following paragraphs.

V_{CC} AND GND

Power is supplied to the PVTC using these two pins. V_{CC} is the +5 volts ±5% power input and GND is the ground connection.

ADDRESS LINES (A0-A2)

These lines are used to select PVTC internal registers for read/write operations and for commands.

DATA BUS (D0-D7)

These lines comprise the 8-bit bidirectional three-state data bus. Bit 0 is the least significant bit and bit 7 is the most significant bit. All data, command, and status transfers between the CPU and the PVTC take place over this bus. The direction of the transfer is controlled by the read and write inputs when the chip enable input is low. When the chip enable input is high the data bus is in the high-impedance state.

READ STROBE ($\overline{R}$)

This pin is an active low input. A low on this pin while chip enable is low causes the contents of the register selected by A0-A2 to be placed on the data bus. The read cycle begins on the falling edge of $\overline{R}$.

WRITE STROBE ($\overline{W}$)

This pin is an active low input. A low on this pin while chip enable is also low causes the contents of the data bus to be transferred to the register selected by A0-A2. The transfer occurs on the rising edge of $\overline{W}$.

CHIP ENABLE ($\overline{CE}$)

This pin is an active low input. When low, data transfers between the CPU and the PVTC are enabled on D0-D7 as controlled by the $\overline{W}$, $\overline{R}$, and A0-A2 inputs. When $\overline{CE}$ is high, the PVTC is effectively isolated from the data bus and D0 through D7 are placed in the high-impedance state.

CHARACTER CLOCK ($\overline{CCLK}$)

This pin is the timing signal derived from the video dot clock which is used to synchronize the PVTC's timing functions.

HORIZONTAL SYNC (HSYNC)

This pin is an active high output which provides video horizontal sync pulses. The timing parameters are programmable.

VERTICAL SYNC/COMPOSITE SYNC (VSYNC/CSYNC)

A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.

BLANK (BLANK)

This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD3 through DADD13 are valid on the trailing edge of BLANK.

CURSOR GATE (CURSOR)

This active high output becomes active for a specified number of scan lines when the address contained in the cursor registers matches the address output on the display address (DADD0 through DADD13). The first and last lines of the cursor and a blink option are programmable.

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

This pin is an open-drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after power-on reset or a master reset command.

LIGHT PEN STROBE (LPS)

This positive edge triggered input indicates a light pen 'hit' causing the current value of the display address to be strobed into the light pen register.

HANDSHAKE CONTROL 1 (CTRL1)

In independent mode, this pin provides an active low write data buffer ($\overline{\text{WDB}}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{\text{PBREQ}}$) input which indicates that the CPU desires to access the display memory. This pin must be tied high when operating in row-buffer mode.

HANDSHAKE CONTROL 2 (CTRL2)

In independent mode, this pin provides an active low read data buffer ($\overline{\text{RDB}}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, CTRL2 is an active low bus external enable ($\overline{\text{BEXT}}$) output which indicates that the PVTC has relinquished control of the display memory (DADD0-DADD13 are in the high-impedance state) in response to a CPU bus request. $\overline{\text{BEXT}}$ also goes low in response to a "display off and float DADD" command. In row-buffer mode, CTRL2 is an active low bus request ($\overline{\text{BREQ}}$) output which halts the CPU during a line DMA.

HANDSHAKE CONTROL 3 (CTRL3)

In independent mode, this pin provides the active low buffer chip enable ($\overline{\text{BCE}}$) signal to the display memory. In transparent and shared modes, CTRL3 provides an active low bus acknowledge ($\overline{\text{BACK}}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, CTRL3 is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.

DISPLAY ADDRESS (DADD0-DADD13)

The display address is used by the PVTC to address up to 16K of display memory. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD3 through DADD13 and are valid at the trailing edge of BLANK. The following paragraphs describe these control signals.

LINE INTERLACE (DADD3/LI) — Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications. A low indicates an even row of an even field or an odd row of an odd field.

LINE ADDRESS (DADD4-DADD7/LA0-LA3) — Provides the number of the current scan line within each character row.

LINE ZERO (DADD8/LNZ) — Asserted before the first scan line in each character row.

LIGHT PEN LINE (DADD9/LPL) — Asserted before the scan line which matches the programmed light pen line position (line three, five, seven, or nine).

UNDERLINE (DADD10/UL) — Asserted before the scan line which matches the programmed underline position (line 0 through 15).

BLINK FREQUENCY (DADD11/BLINK) — Provides an output divided down from the vertical sync rate.

ODD FIELD (DADD12/ODD) — Active high signal which is asserted before each scan line of the odd field when interlace is specified.

LAST LINE (DADD13/LL) — Asserted before the last scan line of character row.

FUNCTIONAL DESCRIPTION

The following paragraphs describe the major blocks (data-bus buffer, interface logic, operation control, timing, display control, and buffer control) which comprise the PVTC.

DATA-BUS BUFFER

The data-bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the PVTC.

INTERFACE LOGIC

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data-bus buffer. The functions performed by the CPU read and write operations are as shown in Table 1.

TABLE 1 — PVTC ADDRESSING

A2	A1	A0	Read ($\overline{R}$=0)	Write ($\overline{W}$=0)
0	0	0	Interrupt Register	Initialization Registers*
0	0	1	Status Register	Command Register
0	1	0	Screen Start Address Lower Register	Screen Start Address Lower Register
0	1	1	Screen Start Address Upper Register	Screen Start Address Upper Register
1	0	0	Cursor Address Lower Register	Cursor Address Lower Register
1	0	1	Cursor Address Upper Register	Cursor Address Upper Register
1	1	0	Light Pen Address Lower Register	Display Pointer Address Lower Register
1	1	1	Light Pen Address Upper Register	Display Pointer Address Upper Register

*There are 11 initialization registers which are accessed sequentially via a simple address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split-screen register) is accessed. The pointer then continues to point to the split-screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the "load IR address pointer" command.

OPERATION CONTROL

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating modes, the interrupt logic, and the status register which provides operational feedback to the CPU.

TIMING

The timing section contains the cursors and decoding logic necessary to generate and monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

DISPLAY CONTROL

The display control section generates linear addressing of up to 16K bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning, storage of light pen "hit" locations, and address comparisons required for generation of timing signals and the split-screen interrupt.

BUFFER CONTROL

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different handshaking schemes are supported. These are described in **SYSTEM CONFIGURATIONS.**

SYSTEM CONFIGURATIONS

A typical display terminal using the MC2670, MC2671, MC2672, and MC2673 CRT terminal devices is shown in Figure 1. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The PVTC supports four common system configurations of display buffer memory, designated the independent, transparent, shared, and row-buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row-buffer mode makes use of a single row buffer (which can be shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user program bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1-CNTRL3 outputs perform different functions for each mode and are named accordingly in the description of each mode given in the following paragraphs.

INDEPENDENT MODE

The CPU-to-RAM interface configuration for this mode is illustrated in Figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not address the memory directly. The read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

The CPU manages the data transfers by supply commands to the PVTC. The commands used are:

1. Read/write at pointer address.
2. Read/write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues "write at cursor with/without increment" or "write at pointer" command.
5. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.

6. PVTC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1. and 3. as above
2. CPU issues "read at cursor with/without increment" or "read at pointer" command.
3. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and PVTC sets RDFLG status to indicate that the read is complete.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the "write from cursor-to-pointer" command:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues "write from cursor-to-pointer" command.
5. PVTC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. PVTC sets RDFLG status to indicate that the block write is completed.

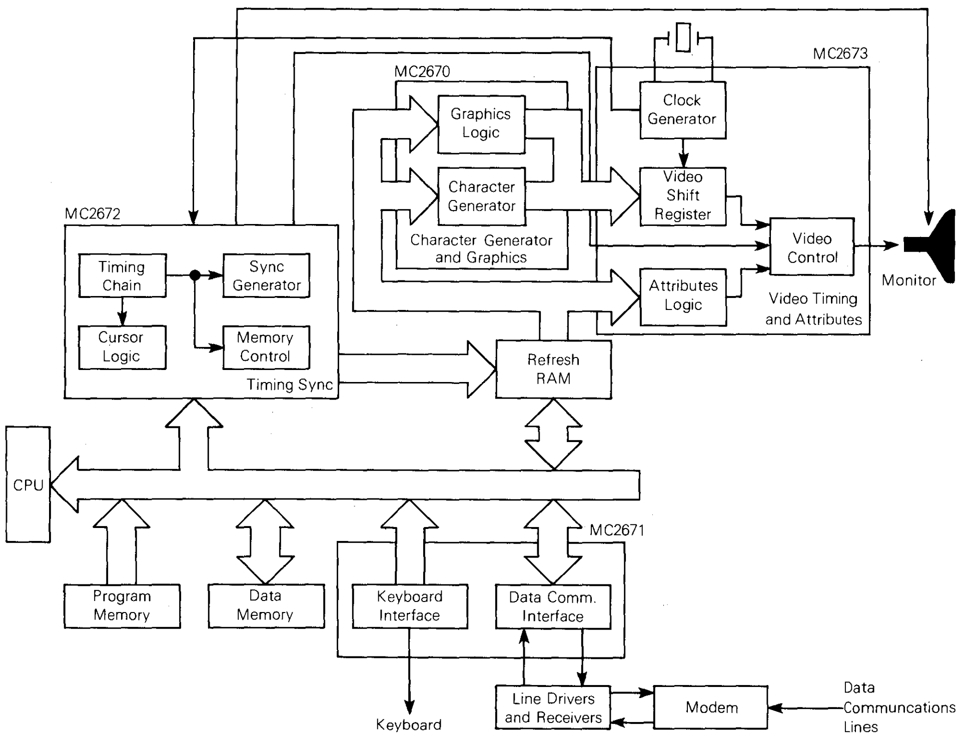
Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously requested command has been completed.

Two timing sequences are possible for the "read/write at cursor/pointer" commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in Figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately one microsecond plus six character clocks (see Figure 4).

Timing for the "write from cursor-to-pointer" operation is shown in Figure 5. The BLANK output is asserted automatically and remains asserted until the vertical retrace interval following completion of the command. The memory is filed at a rate of one location per two character times, plus a small amount of overhead.

3

FIGURE 1 — CRT TERMINAL BLOCK DIAGRAM



3

FIGURE 2 — INDEPENDENT BUFFER-MODE CONFIGURATION

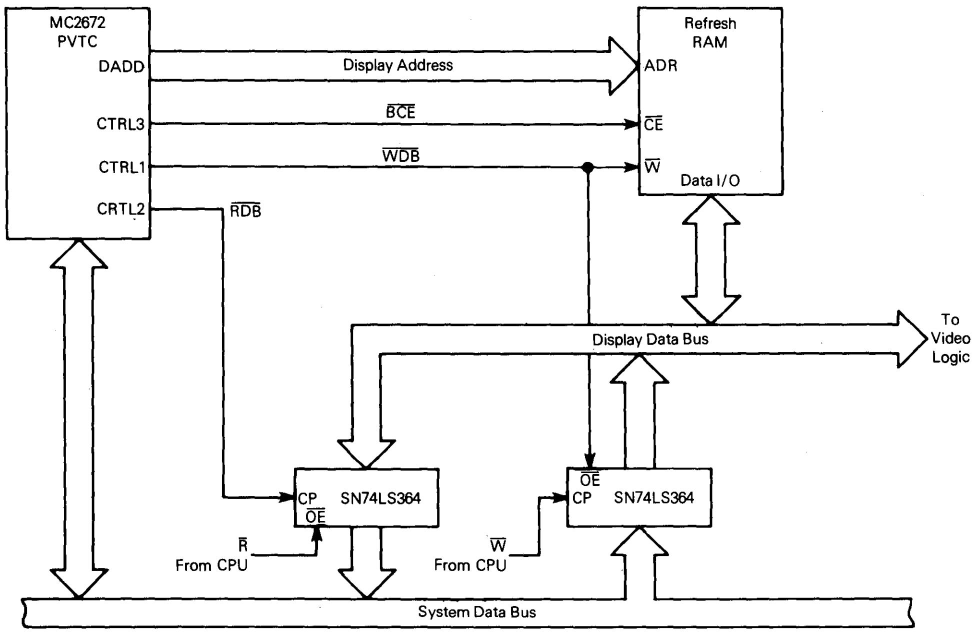
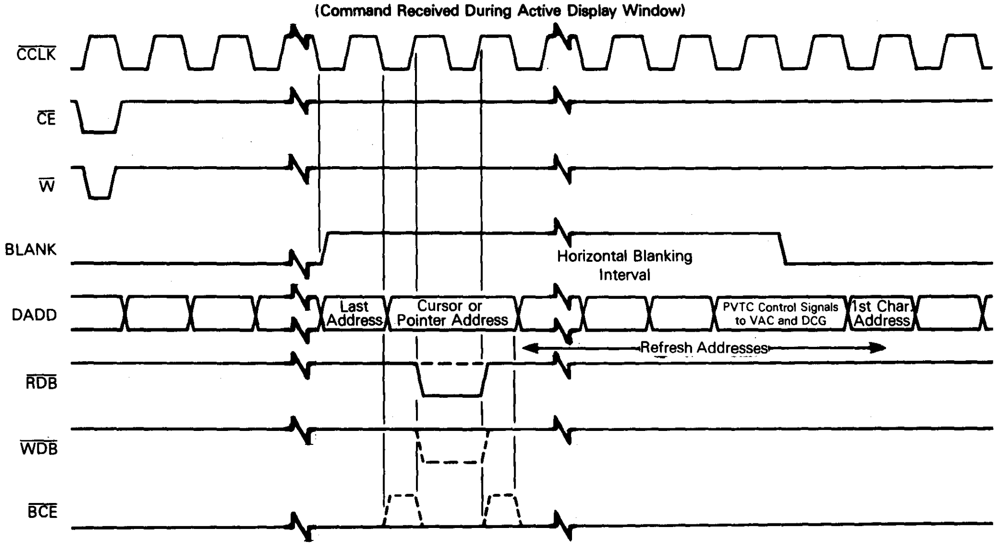


FIGURE 3 — READ/WRITE AT CURSOR/POINTER COMMAND TIMING DIAGRAM
(Command Received During Active Display Window)



NOTE: Write waveforms shown in dotted lines.

FIGURE 4 — READ/WRITE AT CURSOR/POINTER COMMAND TIMING DIAGRAM
(Command Received While Display is Blanked)

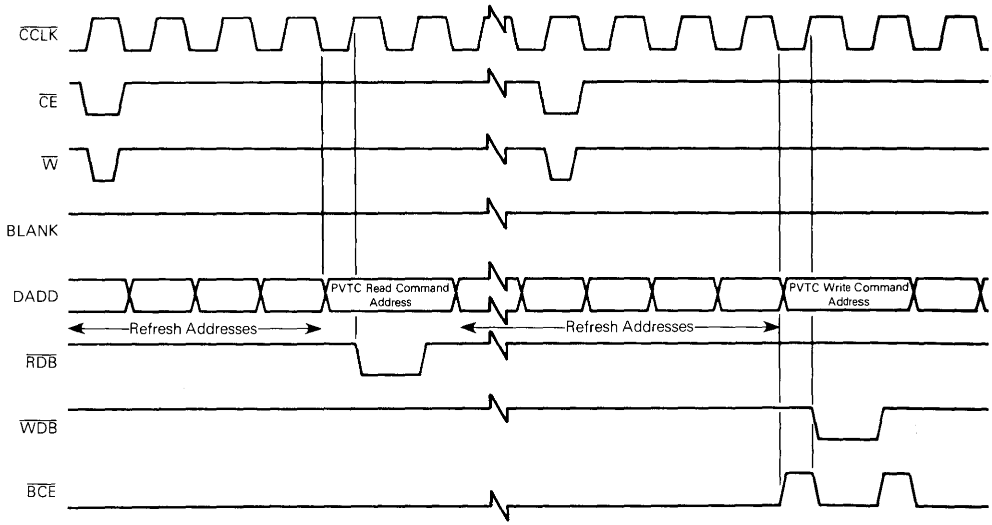
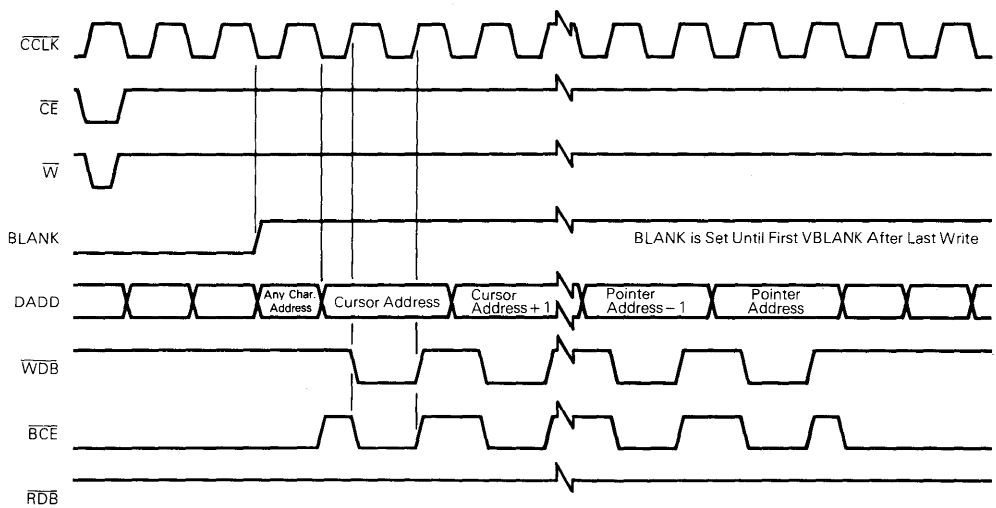


FIGURE 5 — WRITE FROM CURSOR-TO-POINTER COMMAND TIMING



3

SHARED AND TRANSPARENT BUFFER MODES

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see Figure 6). The processor bus request ($\overline{PBREQ}$) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge ($\overline{BACK}$) until its bus external ($\overline{BEXT}$) output has freed the display address and data buses for CPU accesses. $\overline{BACK}$, which can be used as a "hold" input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in Figures 7, 8, and 9.

FIGURE 6 — PVTC SHARED OR TRANSPARENT BUFFER MODES

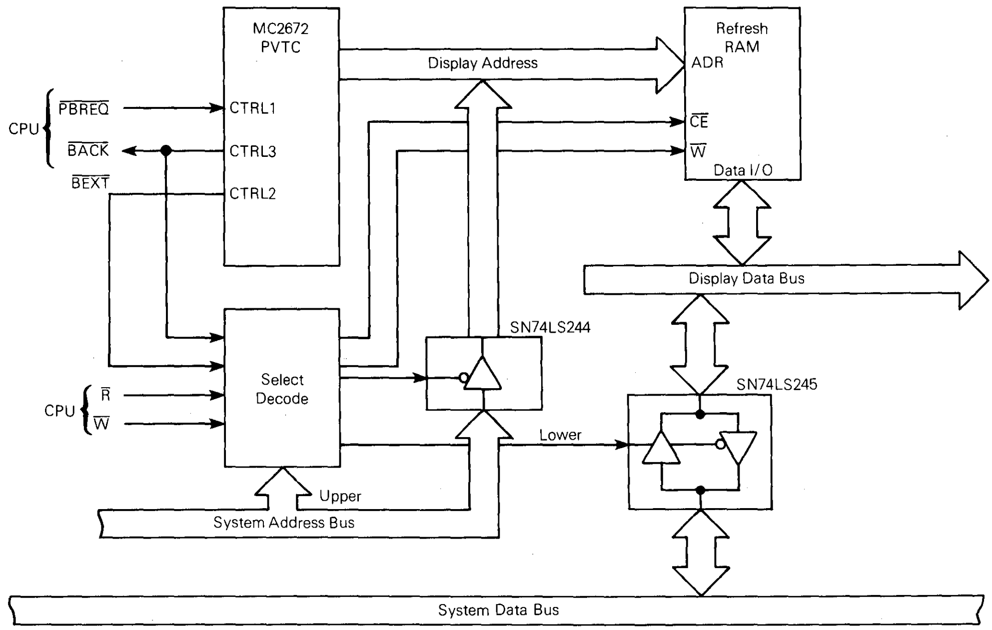
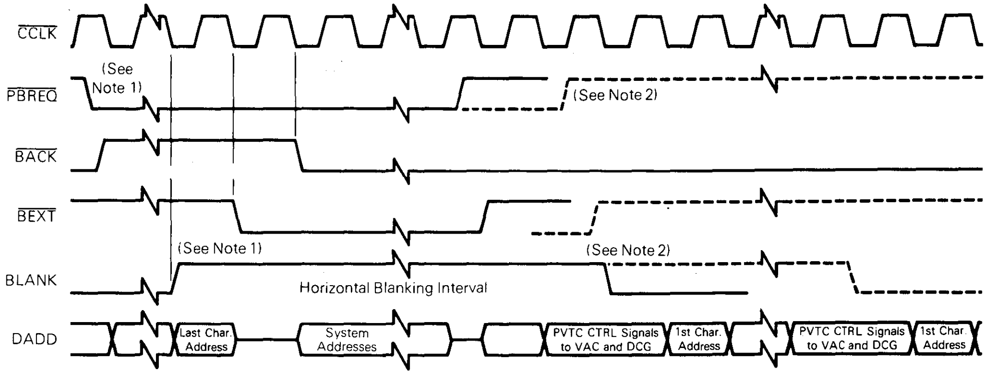
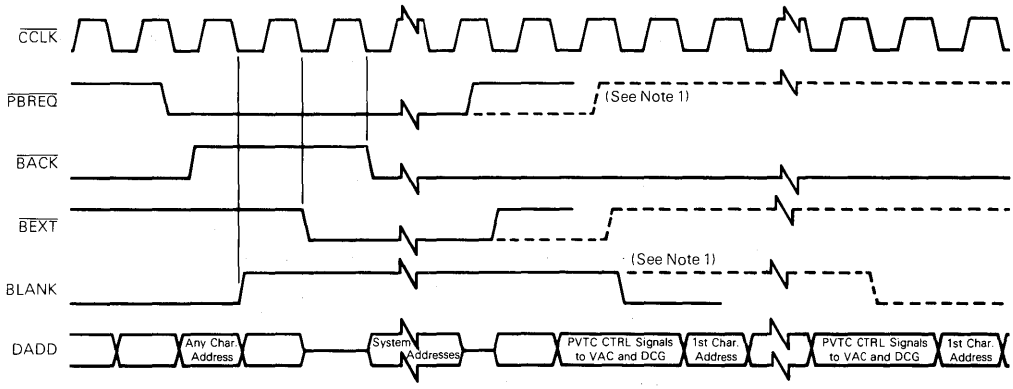


FIGURE 7 — TRANSPARENT-BUFFER MODE TIMING



NOTES:
1. $\overline{PBREQ}$ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If $\overline{PBREQ}$ is negated after the next to last $\overline{CCLK}$ of the horizontal blanking interval, the next scan line will also be blanked.

FIGURE 8 — SHARED-BUFFER MODE TIMING

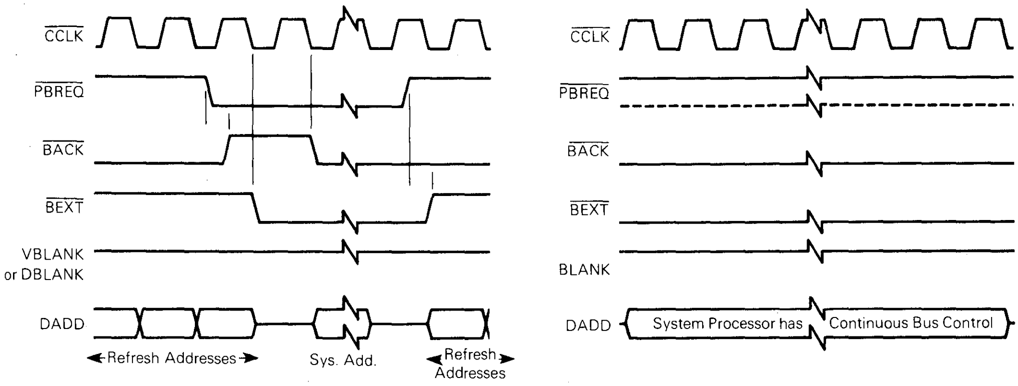


NOTE:
1. If $\overline{PBREQ}$ is negated after the next to last $\overline{CCLK}$ of the horizontal blanking interval, the next scan line will also be blanked.

FIGURE 9 — SHARED AND TRANSPARENT MODE TIMING

(a) During Vertical Blank or after 'display off' command

(b) After 'display off and three-state' command



ROW-BUFFER MODE

Figures 10 and 11 show the timing and a typical hardware implementation for the row-buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMA's the next row of character data from the system memory to row-buffer memory. The PVTC then releases the CPU and displays the row-buffer data for the programmed number of scan lines. The bus-request control ($\overline{\text{BREQ}}$) signal informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data buses before this time to prevent bus contention. After the row of character data is transferred to the CPU, $\overline{\text{BREQ}}$ returns high to grant memory control back to the CPU.

FIGURE 10 — ROW-BUFFER MODE CONFIGURATION

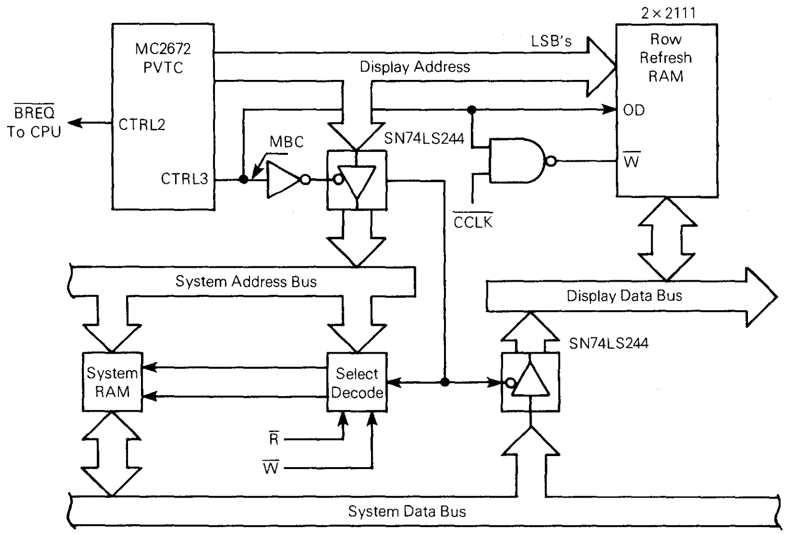
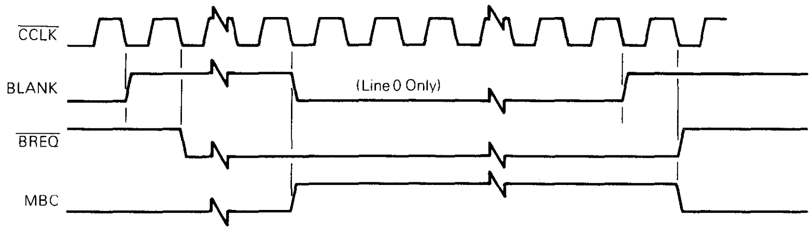


FIGURE 11 — ROW-BUFFER MODE TIMING



OPERATION

After power is applied, the PVTC will be in an inactive state. Two consecutive "master reset" commands are necessary to release this circuitry and ready the PVTC for operation. Two register groups exist within the PVTC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the pointer address for independent memory access mode. These usually require modification during operation.

After initial loading of the two register groups, the PVTC is ready to control the monitor screen. Prior to executing the PVTC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the PVTC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the PVTC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the PVTC supply the "handshaking" information necessary for the CPU to effect the display changes in the proper time frame.

INITIALIZATION REGISTERS

There are 11 initialization registers (IR0-IR10) which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split-screen register) is accessed. The pointer then continues to point to the split-screen register. Upon power-up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the "load IR address pointer" command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in Figure 12 and described in the following paragraphs.

FIGURE 12 — INITIALIZATION REGISTER FORMATS (Page 1 of 3)

	7	6 5 4 3		2	1 0
IR0		Scan Lines Per Character Row			
	Not Used	Non-Interlaced	Interlaced	Sync Select	Buffer-Mode Select
		0000 = 1 Line	0000 = Undefined	0 = VSYNC	00 = Independent
		0001 = 2 Lines	0001 = 5 Lines	1 = CSYNC	01 = Transparent
		0010 = 3 Lines	0010 = 7 Lines		10 = Shared
		•	•		11 = Row
		•	•		
		1110 = 15 Lines	1110 = 31 Lines		
		1111 = 16 Lines	1111 = Undefined		

FIGURE 12 — INITIALIZATION REGISTER FORMATS (Page 2 of 3)

IR1

7	6 5 4 3 2 1 0
Interlace Enable	Equalizing Constant
0 = Non-Interlace 1 = Interlace	0000000 = 1 $\overline{CCLK}$ 0000001 = 2 $\overline{CCLK}$ • • • 1111110 = 127 $\overline{CCLK}$ 1111111 = 128 $\overline{CCLK}$ Calculated from: $EC = 0.5\ (H_{ACT} + H_{FP} + HSYNC + H_{BP}) - 2(HSYNC)$

IR2

7	6 5 4 3	2 1 0
Not Used	Horizontal Sync Width	Horizontal Back Porch
	0000 = 2 $\overline{CCLK}$ 0001 = 4 $\overline{CCLK}$ • • 1110 = 30 $\overline{CCLK}$ 1111 = 32 $\overline{CCLK}$	000 = 1 $\overline{CCLK}$ 001 = 5 $\overline{CCLK}$ • • 110 = 25 $\overline{CCLK}$ 111 = 29 $\overline{CCLK}$

IR3

7 6 5	4 3 2 1 0
Vertical Front Porch	Vertical Back Porch
000 = 4 Scan Lines 001 = 8 Scan Lines • • 110 = 28 Scan Lines 111 = 32 Scan Lines	00000 = 4 Scan Lines 00001 = 6 Scan Lines • • 11110 = 64 Scan Lines 11111 = 66 Scan Lines

IR4

7	6 5 4 3 2 1 0
Character Blink Rate	Active Character Rows Per Screen*
0 = 1/16 VSYNC 1 = 1/32 VSYNC	0000000 = 1 Row 0000001 = 2 Rows • • • 1111110 = 127 Rows 1111111 = 128 Rows

*In interlace mode with odd total character rows per screen the last character row will be the programmed scan lines per character row minus one.

IR5

7 6 5 4 3 2 1 0
Active Characters Per Row
00000010 = 2 Characters 00000011 = 4 Characters • • 11111110 = 255 Characters 11111111 = 256 Characters

FIGURE 12 — INITIALIZATION REGISTER FORMATS (Page 3 of 3)

IR6

7 6 5 4	3 2 1 0
First Line of Cursor	Last Line of Cursor
0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15	0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15

IR7

7 6	5	4	3 2 1 0
Light Pen Line	Cursor Blink	Double Height Char.	Underline Position
00 = Scan Line 3 01 = Scan Line 5 10† Scan Line 7 11† Scan Line 9	0 = No 1 = Yes	0 = No 1 = Yes	0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15

IR8

7 6 5 4 3 2 1 0
Display Buffer First Address LSBs
H"000" = 0 H"001" = 1 • • H"FFE" = 4,094 H"FFF" = 4,095 NOTE: MSBs are in IR9[3:0]

IR9

7 6 5 4	3 2 1 0
Display Buffer Last Address	Display Buffer First Address MSBs
0000 = 1,023 0001 = 2,047 • • 1110 = 15,359 1111 = 16,383	See IR8

IR10

7	6 5 4 3 2 1 0
Cursor Blink Rate	Split-Screen Interrupt Row
0 = 1/16 VSYNC 1 = 1/32 VSYNC	0000000 = Row 0 0000001 = Row 1 • • 1111110 = Row 126 1111111 = Row 127

SCAN LINES PER CHARACTER ROW (IR0[6:3]) — Both interlaced and non-interlaced scanning are supported by the PVTC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the PVTC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0-LA3 and LI pins.

VS/CS ENABLE (IR0[2]) — This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

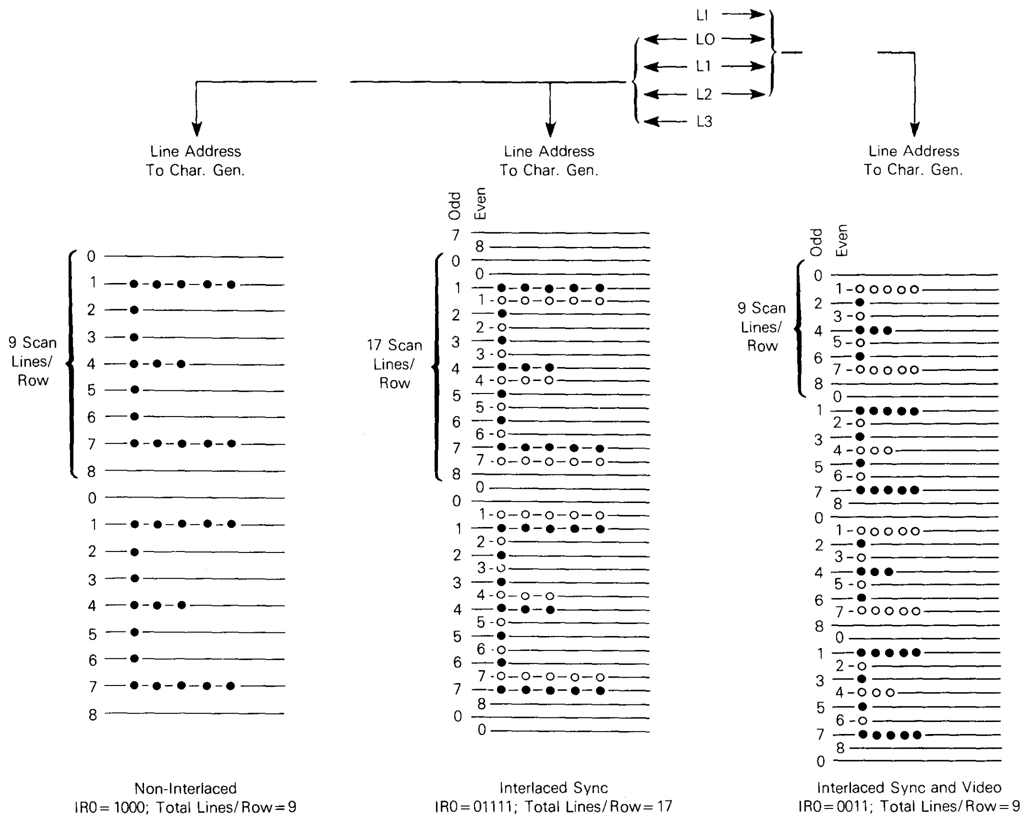
BUFFER MODE SELECT (IR0[1:0]) — Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See **SYSTEM CONFIGURATION.**

INTERLACE ENABLE (IR1[7]) — Specifies interlaced or non-interlaced timing operation. Two modes of interlaced operation are available, depending on whether L0-L3 or LI, L0-L2 are used as the line address for the character generator. The resulting displays are shown in Figure 13.

For "interlaced sync" operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The PVTC outputs successive line numbers in ascending order on the LA0-LA3 lines, one per scan line for each field.

The "interlaced sync and video" format doubles the character density on the screen. The PVTC outputs successive line numbers in ascending order on the LI, LA0-LA2 lines, one per scan line for each field, but alternates beginning the count with even and odd line numbers. This displays the odd field with even scan lines in even character rows and odd scan lines in odd character rows, and the even field with odd scan lines in even character rows and even scan lines on odd character rows. This provides balanced beam currents in the odd and even fields, thus minimizing character variations due to different loading of the CRT anode supply between fields.

FIGURE 13 — INTERLACED DISPLAY MODES



EQUALIZING CONSTANT (IR1[6:0]) — This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{CCLK}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(HSYNC)$$

The definition of the individual parameters is illustrated in Figure 14. The minimum value of H_{FP} is two character clocks.

Note that when using the MC2673 video attributes controller (VAC), the blank pulse is delayed three $\overline{CCLK}$s relative to the HSYNC pulse.

HORIZONTAL SYNC PULSE WIDTH (IR2[6:3]) — This field specifies the width of the HSYNC pulse in $\overline{CCLK}$ periods.

HORIZONTAL BACK PORCH (IR2[2:0]) — This field defines the number of $\overline{CCLK}$s between the trailing edge of HSYNC and the trailing edge of BLANK.

VERTICAL FRONT PORCH (IR3[7:5]) — Programs the number of scan line periods between the rising edges of BLANK and VSYNC during a vertical retrace interval. The width of the VSYNC pulse is fixed at three scan lines.

VERTICAL BACK PORCH (IR3[4:0]) — This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

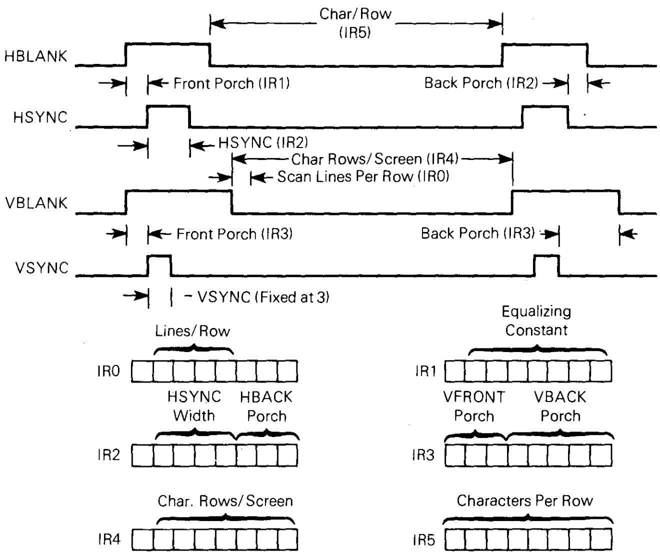
CHARACTER BLINK RATE (IR4[7]) — Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/16 or 1/32 of the vertical field rate. The timing signal has a duty cycle of 75% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

CHARACTER ROWS PER SCREEN (IR4[6:0]) — This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front and back porch values, and the vertical sync pulse width (three scan lines) is the vertical scan period in scan lines.

ACTIVE CHARACTERS PER ROW (IR5[7:0]) — This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period is $\overline{CCLK}$s.

FIRST AND LAST SCAN LINE OF CURSOR (IR6[7:4] AND IR6[3:0]) — These two fields specify the height and position of the cursor on the character block. The "first" line is the topmost line when scanning from the top to the bottom of the screen.

FIGURE 14 — HORIZONTAL AND VERTICAL TIMING



LIGHT PEN LINE POSITION (IR7[7:6]) — This field defines which of four scan lines of the character row will be used for the light pen strike — through attribute by the MC2673 VAC. The timing signal is multiplexed onto the DADD9/LPL output during the falling edge of BLANK.

CURSOR BLINK ENABLE (IR7[5]) — This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR10[7]). The blink duty cycle for the cursor is 50%.

DOUBLE HEIGHT CHARACTER ROW ENABLE (IR7[4]) — If enabled, the number of each scan line will be repeated twice in succession, causing the height of the character row to double. This bit can be changed at any time but will only become effective at the beginning of the character row following the time it is changed. This allows selected character rows to be of double height. The split-screen interrupt can be used to notify the CPU when the effectuate changes to this bit. For each double height row which replaces a normal row, one row count should be subtracted from the "character rows per screen" field (IR4) to maintain the same total number of scan lines per field.

UNDERLINE POSITION (IR7(3:0)) — This field defines which scan line of the character row will be used for the underline attribute by the MC2673 VAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

DISPLAY BUFFER FIRST ADDRESS (IR9[3:0] AND IR8[7:0]) AND DISPLAY BUFFER LAST ADDRESS (IR9[7:4]) — These two fields define the area within the buffer memory where the display data will reside. When the data at the "display buffer last address" is displayed, the PVTC will wrap-around and obtain the data to be displayed at the next screen position from the "display buffer first address". If "last address" is the end of a character row and a new screen start address has been loaded into the screen start register, or if "last address" is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the split-screen interrupt feature of the PVTC.

CURSOR BLINK RATE (IR10[7]) — The cursor blink rate can be specified at 1/16 or 1/32 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

SPLIT-SCREEN INTERRUPT (IR10[6:0]) — The split-screen interrupt can be used to provide special screen effects such as a row of double height characters or to change the normal addressing sequence of the display memory. The contents of this field is compared, in real time, to the current character row number. Upon a match, the PVTC sets the split-screen status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split-screen character row.

TIMING CONSIDERATIONS

Normally, the contents of the initialization registers are not changed during operation. However, this may be necessary to implement special display features such as multiple cursors, smooth scrolling, horizontal scrolling, and double height character rows. Table 2 describes the timing details for these registers which should be considered when implementing these features.

TABLE 2 — TIMING CONSIDERATIONS

Parameter	Timing Considerations
Field Line of Cursor Last Line of Cursor Light Pen Line Underline	These parameters must be established at a minimum of two characters times prior to their occurrence.
Double Height Characters	Set/reset during the character row prior to the row which is to be/not to be double height.
Cursor Blink Cursor Blink Rate Character Blink Rate	New values become effective within one field after values are changed.
Split-Screen Interrupt Row	Change anytime prior to line zero of desired row.
Character Rows Per Screen	Change only during vertical blanking period.
Vertical Front Porch	Change prior to first line of VFP.
Vertical Back Porch	Change prior to fourth line after VSYNC.
Screen-Start Register	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used.

DISPLAY CONTROL REGISTERS

There are nine registers in this group, each with an individual address. Their formats are illustrated in Figure 15. The command register is used to invoke one of 16 possible PVTC commands as described in COMMANDS. The remaining registers in the group store address values which specify the cursor and buffer pointer locations, the location of the first character to be displayed on the screen, and the location of a light pen "hit". With the exception of the light pen register, the user initializes these registers after powering on the system and changes their values to control the data which is displayed.

FIGURE 15 — DISPLAY CONTROL REGISTER FORMATS

(a) Command Register (Write Only)

7	6	5	4	3	2	1	0
Command Code							
Refer to COMMANDS for Command Codes							

(b) Screen Start Registers (Read and Write),
Cursor Address Registers (Read and Write),
Pointer Address Register (Write Only), and
Light Pen Address Register (Read Only)

7	6	5	4	3	2	1	0
		Upper Register					
Not Used		MSBs					

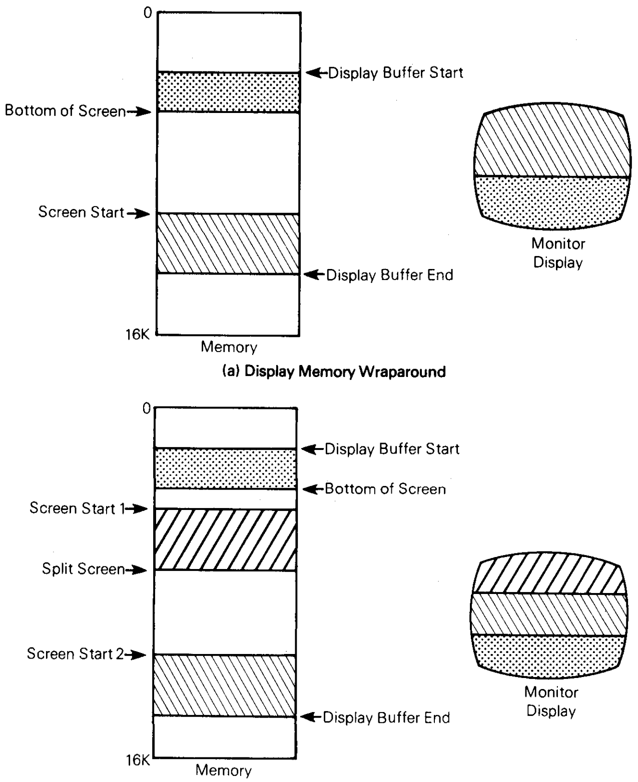
7	6	5	4	3	2	1	0
Lower Register (LSBs)							
H"0000" = 0 H"0001" = 1 • • H"3FFE" = 16,382 H"3FFF" = 16,383 NOTE: MSBs are in Upper Register [5:0]							

SCREEN-START REGISTERS

The screen-start registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row-start register (RSR) and into the memory-address counter (MAC). The counter is then advanced sequentially at the character rate the number of times programmed into the active characters-per-row register (IR5) thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC is loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the entire process repeats again.

The sequential operation described above will be modified upon the occurrence of either of two events. First, if during the incrementing of the memory address counter the "display buffer last address" (IR9[7:4]) is reached, the MAC will be loaded from the "display buffer first address" register (IR9[3:0]), (IR8[7:0]) at the next character clock. Sequential operation will then resume starting form this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see Figure 16a).

FIGURE 16 — DISPLAY ADDRESSING OPERATION



(a) Display Memory Wraparound

(b) Display Memory Split Screen With Wraparound

The sequential row-to-row addressing can also be modified under CPU control. If the contents of the screen-start register (upper, lower, or both) are changed during any character row (say row "n"), the starting address of the next character row (row "n+1") will be the next value of the screen-start register and addressing will continue sequentially from there. This allows features such as split-screen operation, partial scroll, or status line display to be implemented. The split-screen interrupt feature of the PVTC is useful in controlling this type of operation. Note that in order to obtain the correct screen display, the screen-start register must be reloaded with the original value prior to the end of the vertical retrace. See Figure 16b.

During vertical blanking the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered refreshing continues from the display buffer first address.

CURSOR ADDRESS REGISTERS

The contents of these registers define the buffer memory address of the cursor. If enabled, the cursor output will be asserted when the memory address counter matches the value of the cursor address registers. The cursor address registers may be read or written by the CPU or incremented via the "increment cursor address" command. In independent buffer mode, these registers define a buffer memory address for PVTC controlled access in response to "read/write at cursor with/without increment" commands, or the first address to be used in executing the "write for cursor to pointer" command.

DISPLAY POINTER ADDRESS REGISTERS

These registers define a buffer memory address for PVTC controlled accesses in response to "read/write at pointer" commands. They also define the last buffer memory address to be written for the "write from cursor to pointer" command.

LIGHT PEN ADDRESS REGISTERS

If the light pen input is enabled, these registers are used to store the current character address upon receipt of a light pen strobe input. Several sources of delay between the display of a character upon the screen and the receipt of a light pen hit can be expected to exist in a system environment. These delays include address pipelining in the character generation circuits, delays in the video generation circuits, and delays in the light detection circuitry itself. These delays cause the value stored in the light pen register to differ from the actual address of the character at which the light pen hit actually was detected. Software must be used to correct this condition.

INTERRUPT/STATUS REGISTERS

The interrupt and status registers provide information to the CPU to allow it to interface with the PVTC to effect desired changes to implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in Figure 17. These conditions may be selectively enabled or disabled (masked) from causing interrupts by certain PVTC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{\text{INTR}}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set upon occurrence of interrupt condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{\text{INTR}}$ output or the interrupt register.

The status register provides six bits of status information; the five possible interrupting conditions plus the NOT BUSY bit. For this register, however, the contents are not effected by the state of the mask bits.

Descriptions of each interrupt/status register bit follows. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a "reset interrupt/status bits" command. The bits are also reset by a "master reset" command and upon power-up.

RDFLG (SR[5]) — This bit is present in the status register only. A zero indicates that the PVTC is currently executing the previously issued command. A one indicates that the PVTC is ready to accept a new command.

VBLANK (I/SR[4]) — Indicates the beginning of a vertical blanking interval, is set to a one at the beginning of the first scan line of the vertical front porch.

LINE ZERO (I/SR[3]) — Is set to a one at the beginning of the first scan line (line zero) of each active character row.

SPLIT SCREEN (I/SR[2]) — This bit is set when a match occurs between the current character row number and the value contained in the split-screen interrupt register, IR10[6:0]. The equality condition is only checked at the beginning of line zero of each character row. This bit is reset when either of the screen-start registers is loaded by the CPU.

READY (I/SR[1]) — Certain PVTC commands affect the display and may require the PVTC to wait for a blanking interval before enacting the command. This bit is set to one when execution of the command has been completed. No command should be invoked until the prior command is completed.

LIGHT PEN (I/SR[0]) — A one indicates that a light pen hit has occurred and that the contents of the light pen register have been updated. This bit will be reset when either of the light pen registers is read.

COMMANDS

The PVTC commands are divided into two classes: the instantaneous commands, which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in Table 3. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

FIGURE 17 — INTERRUPT AND STATUS REGISTER FORMAT

7–6	5	4	3	2	1	0
Not Used	RDFLG	VBLANK	Line Zero	Split Screen	Ready	Light Pen
Always Read as Zero	0 = Busy 1 = Ready	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Busy 1 = Ready	0 = No 1 = Yes

TABLE 3 — PVTC COMMAND FORMATS

D7	D6	D5	D4	D3	D2	D1	D0	Hex	Command
Instantaneous Commands									
0	0	0	0	0	0	0	0		Master Reset
0	0	0	1	V	V	V	V		Load IR Pointer with Value V (V=0 to 10)
0	0	1	d	d	d	1	0*		Disable Light Pen
0	0	1	d	d	d	1	1*		Enable Light Pen
0	0	1	d	1	N	d	0*		Display Off — Float DADD Bus If N=1
0	0	1	d	1	N	d	1*		Display On — Next Field (N=1) or Scan Line (N=0)
0	0	1	1	d	d	d	0*		Cursor Off
0	0	1	1	d	d	d	1*		Cursor On
0	1	0	N	N	N	N	N		Reset Interrupt/Status — Bit Reset where N=1
1	0	0	N	N	N	N	N		Disable Interrupt — Disable where N=1
0	1	1	N	N	N	N	N		Enable Interrupt — Enables Interrupts and Resets the Corresponding Interrupt/Status Bits where N=1
			V B	L Z	S S	R D	L P		
Delayed Commands									
1	0	1	0	0	1	0	0	A4	Reset at Pointer Address
1	0	1	0	0	0	1	0	A2	Write at Pointer Address
1	0	1	0	1	0	0	1	A9	Increment Cursor Address
1	0	1	0	1	1	0	0	AC	Read at Cursor Address
1	0	1	0	1	0	1	0	AA	Write at Cursor Address
1	0	1	0	1	1	0	1	AD	Read at Cursor Address and Increment Address
1	0	1	0	1	0	1	1	AB	Write at Cursor Address and Increment Address
1	0	1	1	1	0	1	1	BB	Write from Cursor Address to Pointer Address

*Any combination of these three commands is valid.
d = Don't Care

INSTANTANEOUS COMMANDS

The instantaneous commands are executed immediately after the trailing edge of the write pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits. However, a command should not be invoked if the RDFLG bit is low.

MASTER RESET

This command initializes the PVTC and may be invoked at any time to return the PVTC to its initial state. Upon power-up, two successive master reset commands must be applied to release the PVTC's internal power on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of reset and BLANK goes high. BLANK remains high until a "display on" command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The transparent mode, cursor off, display off, and light pen disable states are set.
4. The initialization register pointer is set to address IR0.

LOAD IR ADDRESS

This command is used to preset the initialization register pointer with the value "V" defined by D3-D0. Allowable values are 0 to 10.

ENABLE LIGHT PEN

After invoking this command, receipt of a light pen strobe input will cause the light pen register to be loaded with the current buffer memory address and the corresponding interrupt and status flag to be set. Once loaded, further loads are inhibited until either one of the light pen registers are read or a reset function is performed.

DISABLE LIGHT PEN

Light pen hits will not be recognized.

DISPLAY OFF

Asserts the BLANK output. The DADD0 through DADD13 display address bus outputs may be optionally placed in the high-impedance state by setting bit 2 to a one when invoking the command.

DISPLAY ON

Restores normal blanking operation either at the beginning of the next field (bit 2=1) or at the beginning of the next scan line (bit 2=0). Also returns the DADD0-DADD13 drivers to their active state.

CURSOR OFF

Disables cursor operation. Cursor output is placed in the low state.

CURSOR ON

Enables normal cursor operation.

RESET INTERRUPT/STATUS BITS

This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 – Light Pen
Bit 1 – Ready
Bit 2 – Split Screen
Bit 3 – Line Zero
Bit 4 – Vertical Blank

DISABLE INTERRUPTS

Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from asserting the $\overline{\text{INTR}}$ output. Bit position correspondence is as above.

ENABLE INTERRUPTS

Resets the selected interrupt and status register bits and writes the associated interrupt mask bits to a one. This enables the corresponding conditions to assert the $\overline{\text{INTR}}$ output. Bit position correspondence is as above.

DELAYED COMMANDS

This group of commands is utilized for the independent buffer mode of operation, although the "increment cursor" command can also be used in other modes. With the exception of the "write from cursor to pointer" and "increment cursor" commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking blanking interval. If invoked during a vertical retrace interval or a "display off" state, the command is executed immediately.

MOTOROLA

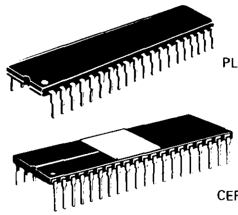
MC2673

Advance Information

HMOS

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

VIDEO ATTRIBUTES CONTROLLER (VAC)



P SUFFIX
PLASTIC PACKAGE
CASE 711

L SUFFIX
CERAMIC PACKAGE
CASE 715

VIDEO ATTRIBUTES CONTROLLER (VAC)

The MC2673A and MC2673B video attributes controllers (VAC) are bipolar LSI devices designed for CRT terminals and display systems that employ raster scan techniques. Each contains a high-speed video shift register, field and character attributes logic, attribute latch, cursor format logic, and half-dot shift control.

The VAC provides control of visual attributes on a field or character by character. Internal logic preserves field attribute data from character row to character row so that an attribute byte is not required at the beginning of each row. The MC2673B provides for reverse video, blank (non-display), blink, underline, and highlight attributes and a graphics mode attribute to work in conjunction with the MC2670 display character and graphics generator (DCGG). The MC2673A substitutes a light pen (strike-thru) attribute for the graphics attribute.

The horizontal dot frequency is the basic timing input to the VAC. Internally, this clock is divided down to provide a character clock output for system asynchronization. Up to ten bits of video dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the VIDEO output, the data is presented as a three-level signal representing low, medium, and high intensities. The three intensities are also encoded on two TTL compatible video outputs. Light or dark screen background can be selected.

- 25 MHz Video Dot Rate
- Three-Level Current Driven (75 Ohms) Video Output
- Three-Level Encoded TTL Video Outputs
- Character/Field Attribute Logic:
  - Reverse Video
  - Character Blank
  - Character Blink
  - Underline
  - Highlight
  - Light Pen Strike-Thru or Graphics Control
- Field Attributes Extend from Row to Row
- Light or Dark Field
- Cursor Reverse Video Logic
- Up to Ten Dots Per Character
- Composite Blanking for Light Field Retrace
- Optional Field Graphics Control Output
- High-Speed Bipolar Design
- 40-Pin Dual-in-Line Package
- TTL Compatible
- Compatible with the MC2672 PVTC and MC2670 DCGG
- Applications Include:
  - CRT Terminals
  - Word Processing Systems
  - Small Business Computers

PIN ASSIGNMENT

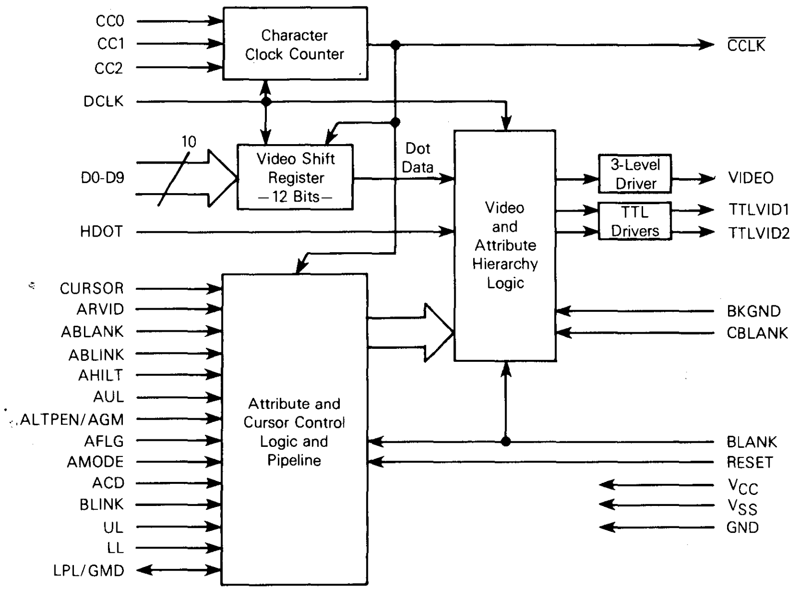
V_{SS}	1	40	V_{CC}
D3	2	39	D2
D4	3	38	D1
D5	4	37	D0
D6	5	36	$\overline{CCLK}$
D7	6	35	CC0
D8	7	34	CC1
D9	8	33	CC2
RESET	9	32	DCLK
BKGND	10	31	CBLANK
ACD	11	30	TTLVID1
AMODE	12	29	TTLVID2
AFLG	13	28	VIDEO
CURSOR	14	27	HDOT
BLANK	15	26	ABLANK
UL	16	25	ABLINK
BLINK	17	24	AUL
LL	18	23	AHILT
LPL/GMD	19	22	ARVID
GND	20	21	ALTPEN/AGM

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

ORDERING INFORMATION
(V_{CC}=5 V±5%, T_A=0° to 70°C)

Package Type	Light-Pen Attribute		Graphics Attribute	
	Frequency	Order Number	Frequency	Order Number
Ceramic L Suffix	18 MHz 25 MHz	MC2673A8L MC2673A5L	18 MHz 25 MHz	MC2673B8L MC2673B5L
Plastic P Suffix	18 MHz 25 MHz	MC2673A8P MC2673A5P	18 MHz 25 MHz	MC2673B8P MC2673B5P

VIDEO ATTRIBUTES CONTROLLER BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.5 to +6.0	V
Input Voltage	V_{in}	−0.5 to +6.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages 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 Plastic Package Ceramic Package	θ_{JA}	 50 50	°C/W

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistant, 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 $P_{PORT} \ll P_{PINT}$ 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:

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

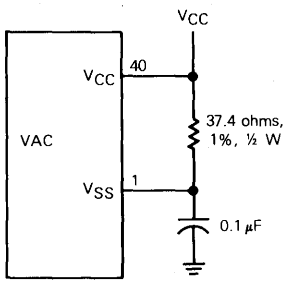
Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS ($T_A = 0°C$ to 70°C, $V_{CC} = 5.0$ V ±5%, see Figure 1)

Parameter	Symbol	Min	Typ	Max	Unit
Input Low Voltage	V_{IL}	–	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	–	V
Output Low Voltage (Except VIDEO) $I_{OL} = 4$ mA	V_{OL}	–	–	0.4	V
Output High Voltage (Except VIDEO) $I_{OH} = -400$ μA	V_{OH}	2.4	–	–	V
VIDEO Black Level $R_L = 150$ Ohms to GND	V_B	–	0	–	V
VIDEO Gray Level $R_L = 150$ Ohms to GND	V_G	–	0.45	–	V
VIDEO White Level $R_L = 150$ Ohms to GND	V_W	–	0.90	–	V
Input Low Current $V_{in} = 0.4$ V	I_{IL}	–	–	−400/ −800*	μA
Input High Current $V_{in} = 2.4$ V	I_{IH}	–	–	20/ 40*	μA
Power Supply Current $V_{in} = 0$ V, $V_{CC} =$ Max, $V_{SS} =$ Max	I_{CC}	–	–	80	mA
Bias Supply Current $V_{in} = 0$ V, $V_{CC} =$ Max, $V_{SS} =$ Max	I_{SS}	–	–	120	mA

*For DCLK input

FIGURE 1 — TEST DIAGRAM



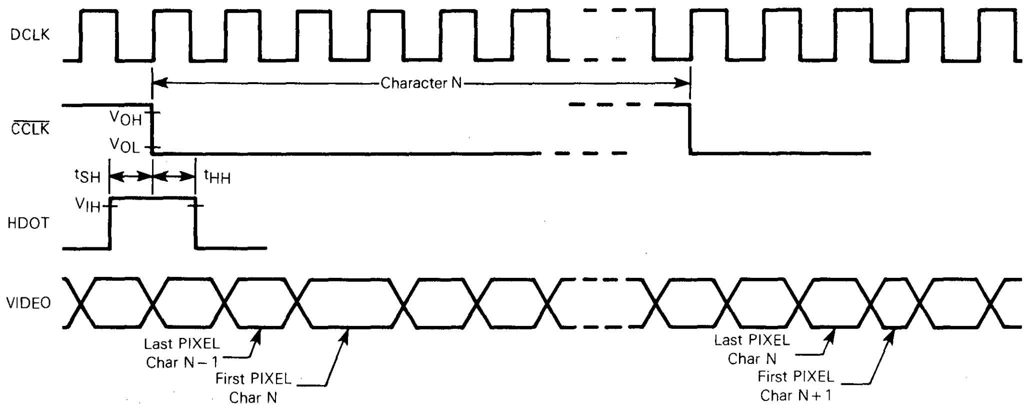
3

AC ELECTRICAL CHARACTERISTICS ($T_A = 0°C$ to $70°C$, $V_{CC} = 5$ V ±5%, see Figure 1)

Parameter	Symbol	25 MHz		18 MHz		Unit
		Min	Max	Min	Max	
Dot Clock Frequency (see Figure 2)	f_D	–	25	–	18	MHz
Dot Clock High (see Figure 2)	t_{DH}	15	–	22	–	ns
Dot Clock Low (see Figure 2)	t_{DL}	15	–	22	–	ns
BLANK to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{BS}	50	–	50	–	ns
BLINK, UL, LPL, LL (Ref. to BLANK) to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{SC}	20	–	20	–	ns
Attributes to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{SA}	45	–	55	–	ns
Dot Data D0-D9 to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{SD}	70	–	70	–	ns
CURSOR to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{SK}	50	–	50	–	ns
AFLG to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{FS}	50	–	65	–	ns
HDOT to $\overline{CCLK}$ Setup Time (see Figures 2, 3, 4, and 5)	t_{SH}	45	–	55	–	ns
BLINK, UL, LPL, LL (Ref. to BLANK) Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{HC}	20	–	20	–	ns
Attributes Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{HA}	20	–	20	–	ns
Dot Data D0-D9 Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{HD}	30	–	30	–	ns
CURSOR Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{HK}	20	–	20	–	ns
AFLG Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{FH}	30	–	30	–	ns
HDOT Hold Time from $\overline{CCLK}$ (see Figures 2, 3, 4, and 5)	t_{HH}	20	–	20	–	ns
BKGND to DCLK Setup Time (see Figure 6)	t_{SG}	15	–	15	–	ns
CBLANK to DCLK Setup Time (see Figure 6)	t_{SB}	15	–	15	–	ns
BKGND Hold Time from DCLK (see Figure 6)	t_{HG}	15	–	15	–	ns
CBLANK Hold Time from DCLK (see Figure 6)	t_{HB}	15	–	15	–	ns
GMD from DCLK Delay Time $C_L = 150$ pF (see Figures 5 and 7)	t_{DGM}	–	65	–	65	ns
CCLK from DCLK Delay Time* $C_L = 150$ pF (see Figures 5 and 7)	t_{DC}	–	65	–	65	ns
TTLVID1 and TTLVID2 from DCLK Delay Time $C_L = 150$ pF (see Figures 5 and 7)	t_{DV}	45	75	45	80	ns
VIDEO from DCLK Delay Time $C_L = 150$ pF (see Figures 5 and 7)	t_{DV}	–	240	–	240	ns

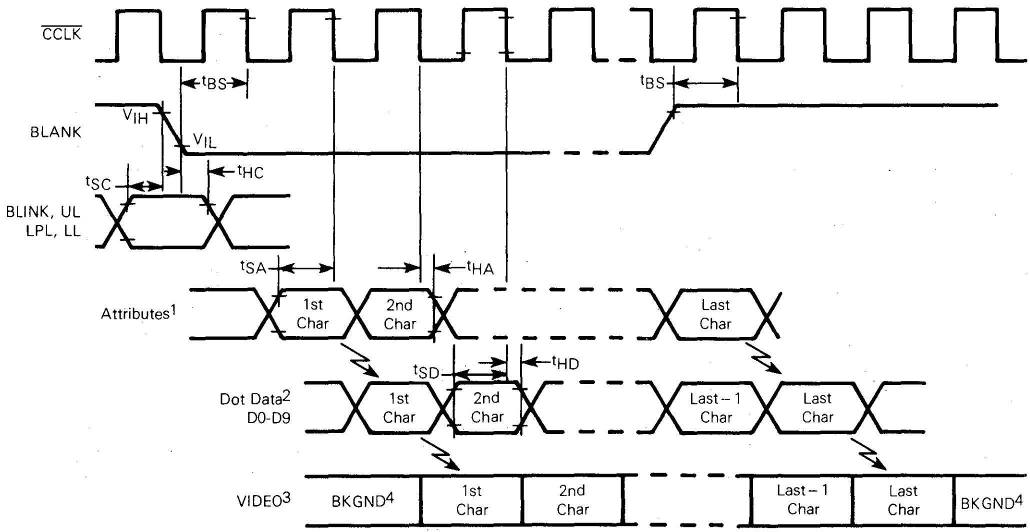
* C_L less than 150 picofarads could be faster.

FIGURE 2 — HALF-DOT SHIFT TIMING DIAGRAM



NOTE: Half-dot shift feature 18 MHz maximum.

FIGURE 3 — VAC PIPELINE TIMING DIAGRAM



NOTES

1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, and ALTPEN.
2. One $\overline{\text{CCLK}}$ delay for dot data (obtained from delay through character generator).
3. See Figure 7 for detail timing of VIDEO, TTLVID1, TTLVID2.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input.

FIGURE 4 — CURSOR PIPELINE TIMING DIAGRAM

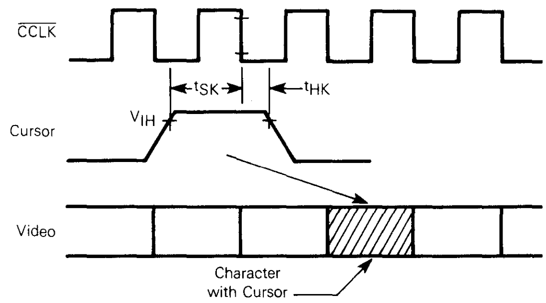
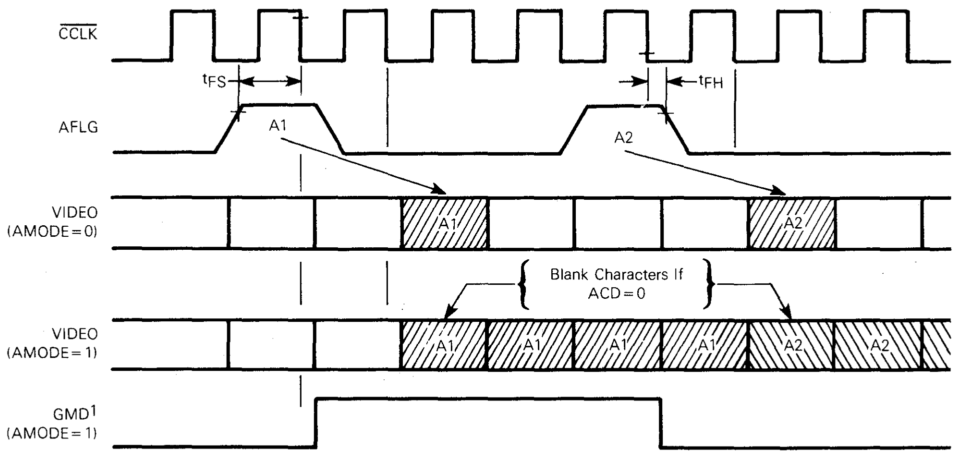


FIGURE 5 — CHARACTER (AMODE=0), FIELD (AMODE=1), AND GMD ATTRIBUTE TIMING DIAGRAM



NOTE

1. GMD output in MC2673B version only. See Figure 7 for detail timing.

FIGURE 6 — BKGND AND CBLANK TIMING DURING INACTIVE SCAN TIME (BLANK=1)

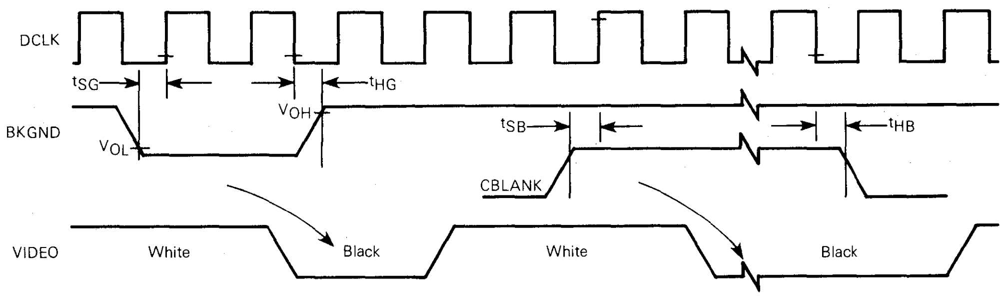
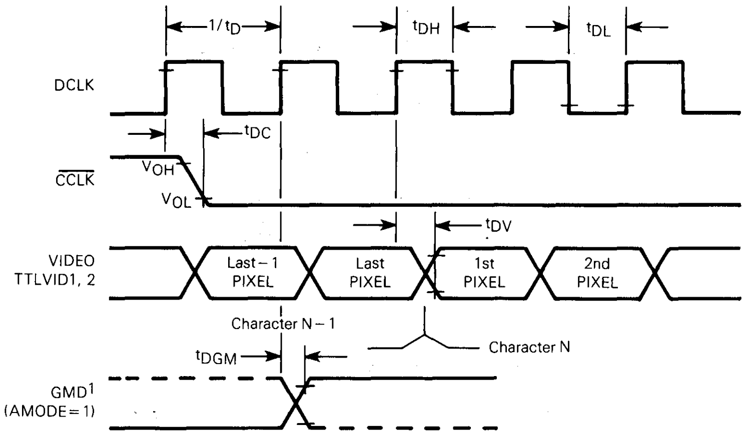


FIGURE 7 — VIDEO AND GMD PIPELINE TIMING DIAGRAM



NOTE
1. GMD output in MC2673B version only.

SIGNAL DESCRIPTION

The input and output signals for the VAC are described in the following paragraphs.

DOT CLOCK (DCLK)

This input controls the dot frequency and video shift rate.

CHARACTER CLOCK ($\overline{CCLK}$)

This output is a submultiple of DCLK. The frequency ranges from one sixth to one twelfth of DCLK, as determined by the state of the CC0-CC2 inputs.

CHARACTER CLOCK CONTROL (CC2-CC0)

The logic state of these three static inputs determine the internal divide factor for the CCLK output rate. Character clock rates of 6 through 12 dots per character may be specified.

DOT DATA INPUT (D0-D9)

These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the falling edge of each character clock.

HALF-DOT SHIFT (HDOT)

When this input is high, the serial video output is delayed by one-half dot time. This input is latched on the falling edge of each character clock.

CURSOR TIMING (CURSOR)

This input provides the timing for the cursor video. When high, it effectively reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.

BACKGROUND INTENSITY (BKGND)

This input specifies light or dark video during BLANK and character fields. Affects the intensities of all attributes.

SCREEN BLANK (BLANK)

When high, this input forces the video outputs to the level specified by the BKGND input (either high or low intensity). However, BLANK is not effective when composite blank (CBLANK) is high.

COMPOSITE BLANK (CBLANK)

This input is used with the TTL video outputs only. When high, CBLANK forces the video outputs to a low intensity state for retrace blanking. When BKGND input is low, or when using video outputs, this input may be tied low.

REVERSE VIDEO ATTRIBUTE (ARVID)

This input causes the intensity of the associated character or field video to be reversed. All other attributes are effectively reversed.

HIGHLIGHT ATTRIBUTE (AHILT)

This input causes all dot video (including underline) of the associated character or field to be highlighted with respect to the BKGND input and the reverse video attribute.

BLANK ATTRIBUTE (ABLANK)

This input generates a blank space in the associated character or field. The blank space intensity is determined by the BKGND input, the reverse video attribute, and the CURSOR input.

BLINK ATTRIBUTE (ABLINK)

This input causes the associated character or field video to be driven to the intensity determined by BKGND and the reverse video attribute when the BLINK input is high.

UNDERLINE ATTRIBUTE (AUL)

This input specifies a line to be displayed on the character or field. The line is specified by the underline (UL) input. All other attributes apply to the underline video.

LIGHT PEN ATTRIBUTE (ALTPEN)

This input of the MC2673A specifies a highlighted line to be displayed on the character or field. The line is specified by the LPL input.

ATTRIBUTE GRAPHICS MODE (AGM)

This input of the MC2673B is latched and synchronized to provide a field graphics mode output for the MC2670 DCGG.

ATTRIBUTE MODE (AMODE)

This input specifies character (AMODE=0) or field (AMODE=1) attributes mode.

ATTRIBUTES FLAG (AFLG)

This input, when high, causes the VAC to sample and latch the attributes inputs. If field attributes are specified (AMODE=1), the attributes are double buffered on a row basis. Thus, each scan line of every character row will start with the attributes that were valid at the end of the previous row.

ATTRIBUTE CONTROL DISPLAY (ACD)

In field attributes mode (AMODE=1), if ACD=0, the first character in each new attribute field (the attribute control character) will be suppressed and only the attributes will be displayed. If ACD=1, the first character and the attributes are displayed. This input has no effect in character mode (AMODE=0).

BLINK (BLINK)

This input is sampled on the falling edge of BLANK to provide the blink rate for the character blink attribute. It should be a submultiple of the frame rate.

UNDERLINE (UL)

This input indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.

LIGHT-PEN LINE (LPL)

For the MC2673A, this input indicates the scan line(s) for the light pen strike-thru attribute. Latched on the falling edge of BLANK.

GRAPHICS MODE (GMD)

For the MC2673B, this output provides a synchronized, latched, field graphics mode corresponding to the AGM input. This output can be used to control the GM input on the MC2670 DCGG.

LAST LINE (LL)

This input indicates the last scan line of each character row and is used internally to extend field attributes across row boundaries. Latched on the falling edge of BLANK. This input has no effect in character mode (AMODE=0).

VIDEO (VIDEO)

This is a three-level serial video output which corresponds to the composite dot pattern of characters, attributes, and cursor.

TTL VIDEO 1 (TTLVID1)

This output corresponds to the serial, non-highlighted video dot pattern.

TTL VIDEO 2 (TTLVID2)

This output corresponds to the highlighted serial video dot pattern. Should be used with TTLVID1 to decode a composite video of three intensities.

MANUAL RESET (RESET)

This active high input initializes the internal logic and resets the attribute latches. Normally used for testing.

V_{CC}, V_{SS}, AND GND

Power is supplied to the VAC using these three pins. V_{CC} is the +5 volts ±5% power input, V_{BB} is the bias supply (see Figure 1), and GND is the ground connection.

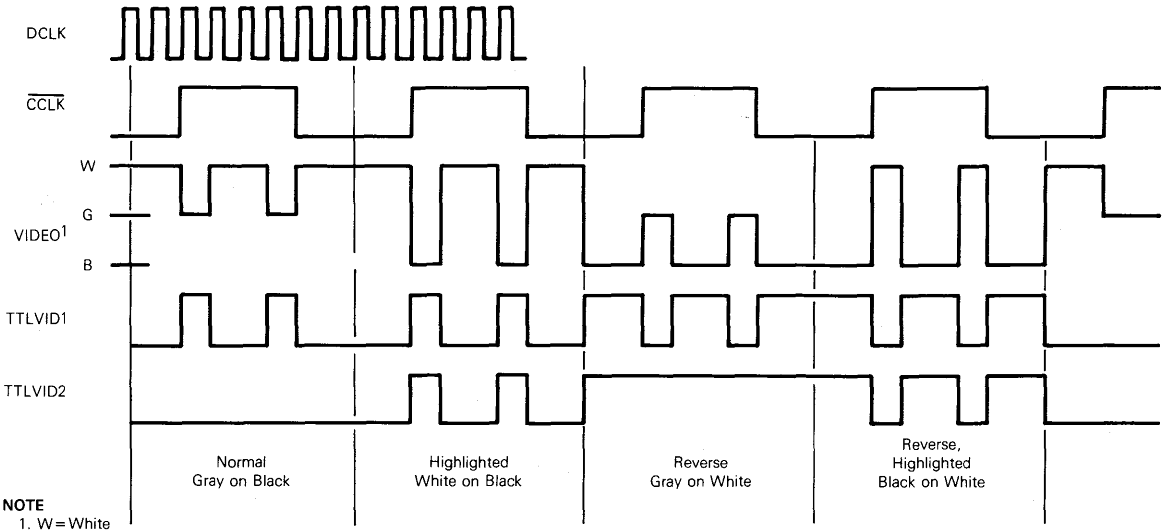
FUNCTIONAL DESCRIPTION

The VAC consists of four major sections. The high speed dot clock input is divided internally to provide a character clock for system timing. The parallel dot data is loaded into the video shift register on each character boundary and shifted into the video logic block at the dot rate. The six attribute inputs are latched internally and combined with the serial dot data to provide a three-level video source for the monitor.

A separate BLANK input defines the active screen area. When BLANK=0, the video levels are derived internally by the combinations of dot data, attributes, cursor, and the state of the BKGND input. Either black or white background can be selected. Symbols (dot data) are normally gray and can be highlighted to white or black as shown in Figure 8. Note that the VIDEO output is inverted as referenced to the TTL video outputs. The video output stages of the MC2673 are illustrated in Figure 9.

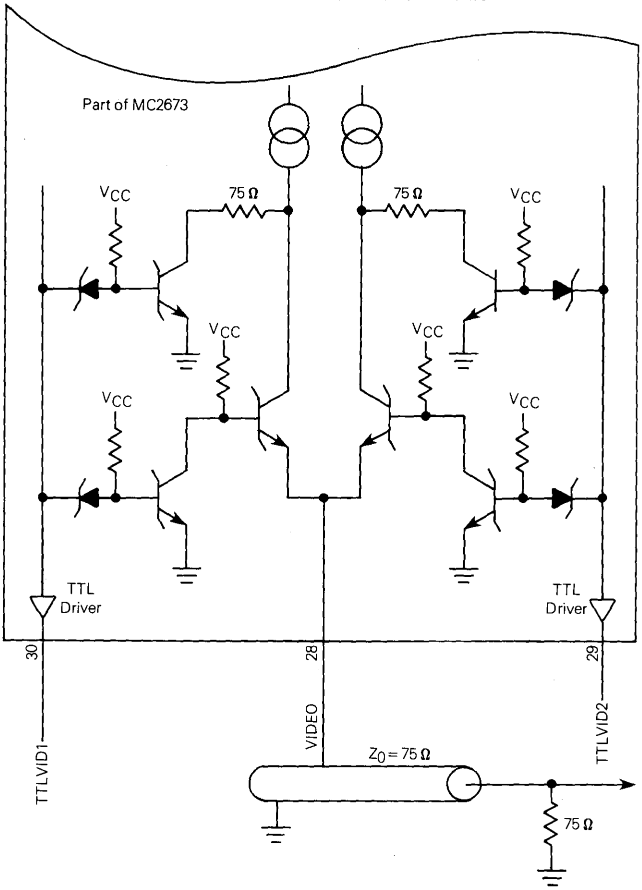
During the inactive screen area (BLANK=1), the video level produced by the TTL outputs in either white (BKGND=1) or black (BKGND=0). A separate composite blank (CBLANK) input is provided to suppress raster retrace video when white background is specified. During the inactive screen area (BLANK=1), the video level produced by the VIDEO output is either black (BKGND=1) or white (BKGND=0). For the latter case, raster retrace video suppression is accomplished by raising the BKGND input during horizontal and vertical retrace intervals. For black background, tie BKGND high. Tie CBLANK input low for both cases.

FIGURE 8 — ENCODED VIDEO OUTPUTS



NOTE
1. W = White
   G = Gray
   B = Black

FIGURE 9 — VIDEO OUTPUT STAGES OF THE MC2673



CHARACTER CLOCK COUNTER

The character clock counter divides the frequency on the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (CC0-CC2) as follows:

			Character Clock ($\overline{CCLK}$)	
CC2	CC1	CC0	Dots/Character	Duty Cycle
0	0	0	6	3/3
0	0	1	6	3/3
0	1	0	7	4/3
0	1	1	8	4/4
1	0	0	9	5/4
1	0	1	10	5/5
1	1	0	11	6/5
1	1	1	12	6/6

VIDEO SHIFT REGISTER

On each character boundary, the parallel data (D0-D9) is loaded into the video shift register. The data is shifted out least significant bit first (D0) by the DCLK. If 11 or 12 dots/character are specified (CC2-CC0 = 110 or 111), a zero (blank dot) is always shifted out before D0. For 12 dots/character, a zero is also shifted out after D9. The serial dot data is shifted into the video logic where it is combined with the cursor and attributes to encode three levels of video.

ATTRIBUTE AND CURSOR CONTROL

The VAC visual attributes capabilities include: reverse video, character blank, blink, underline, highlight, and light pen strike-thru. The six attributes and the three attribute control inputs (AMOD, AFLG, and ACD) are clocked into the VAC on the falling edge of $\overline{CCLK}$. If AFLG is high, the attributes are latched internally and are effective for either one character time (AMODE=0) or until another set of attributes is latched (AMODE=1). The attributes set is double buffered on a row-by-row basis internally. Using this technique, field attributes can extend across character row boundaries thereby eliminating the necessity of starting each row with an attribute set.

When field attribute mode is selected, (AMODE=1), the VAC will accomodate two attribute storage configurations. In one configuration, the attribute control data is stored in the refresh RAM, taking the place of the first character code in the field to be affected. For this mode, the ACD input is tied low and blank characters will be displayed in the screen positions occupied by the attribute data (see Figure 10). In the second configuration, (ACD=1), the character codes and attribute data are presented to the VAC in parallel. In this mode, dot data is displayed at each character position (see Figure 11).

The CURSOR and the attribute input signals are pipelined internally to allow for system propagations (one $\overline{CCLK}$ for refresh RAM, one $\overline{CCLK}$ for dot generator). The attribute timing signals BLINK, UL, LPL, and LL are clocked into the VAC at the beginning of each scan line by the falling edge of the BLANK input. Thus, these signals must be in their proper state at the falling edge of BLANK preceding the scan line at which they are to be active (see Figure 3).

FIGURE 10 — SYSTEM BLOCK DIAGRAM OF THE MC2673 IN FIELD ATTRIBUTE MODE USING THE NARROW RAM (8 WIDE) CONFIGURATION

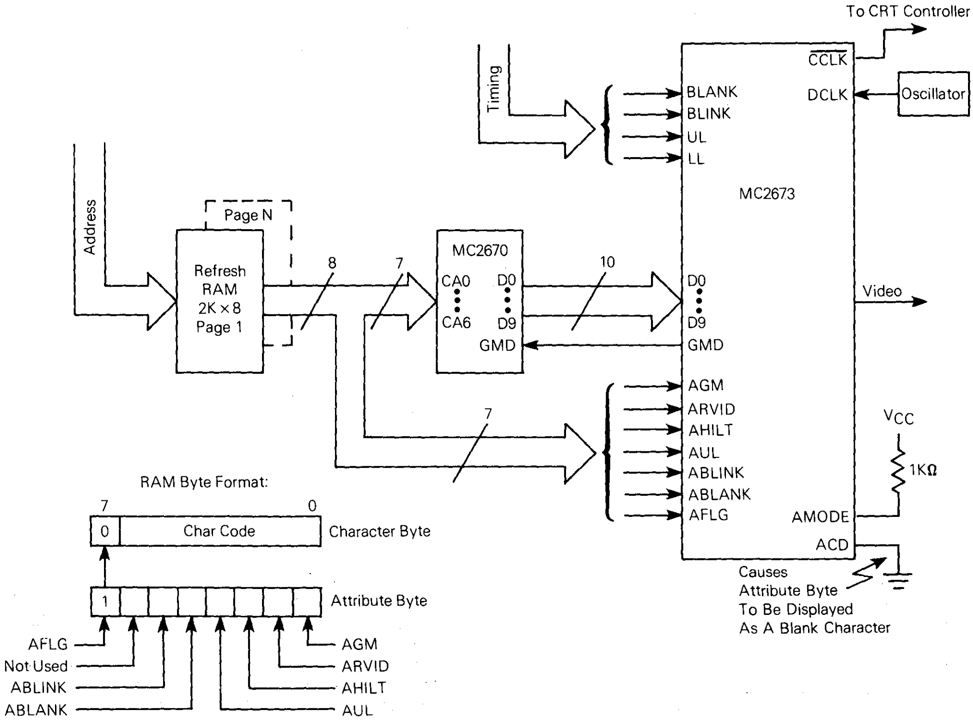
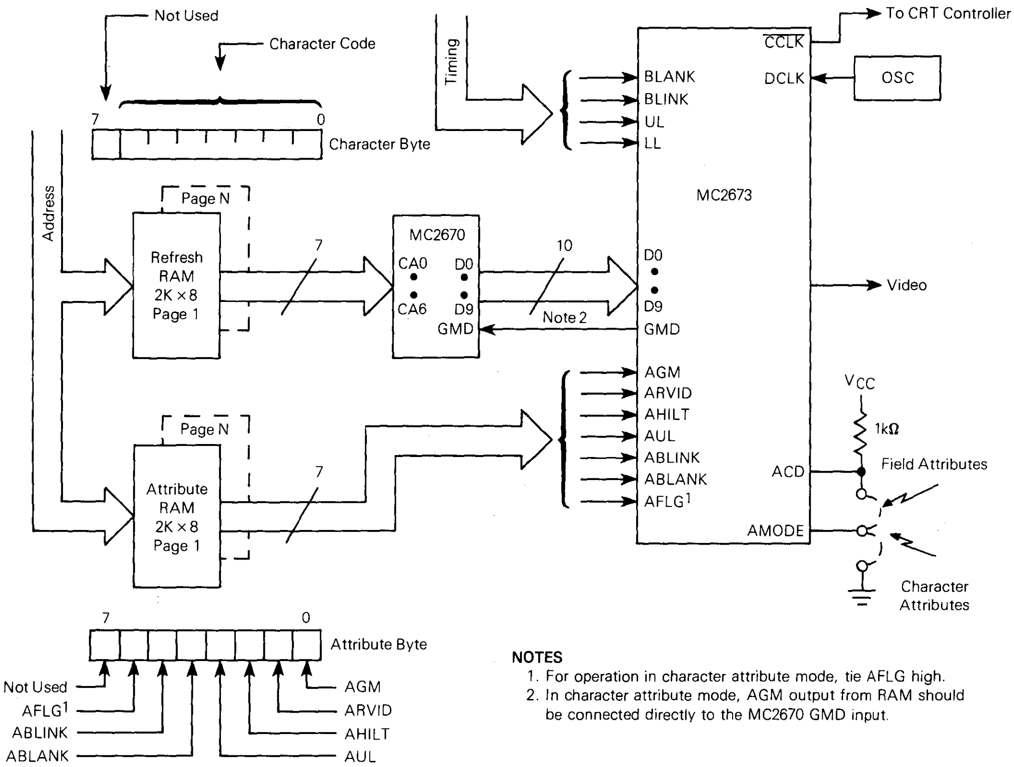


FIGURE 11 — SYSTEM BLOCK DIAGRAM OF THE MC2673 IN FIELD OR CHARACTER ATTRIBUTE MODE USING THE WIDE RAM CONFIGURATION



NOTES
1. For operation in character attribute mode, tie AFLG high.
2. In character attribute mode, AGM output from RAM should be connected directly to the MC2670 GMD input.

VIDEO LOGIC

The serial dot data and the pipeline cursor and attributes are combined to generate the three-level current source on the VIDEO output. The three levels (white, gray, and black) are also encoded on the two TTL compatible outputs TTLVID1 and TTLVID2. The three levels are encoded as shown below:

TTLVID2	TTLVID1	Intensity
0	0	Black (for CBLANK)
0	1	Gray (on black surround)
1	0	Gray (on white surround)
1	1	White

NOTE: The TTLVID1 output can be used independently to generate a two-level non-highlighted video.

The video is normally shifted out on the leading edge of the DCLK. When the HDOT input is asserted, the corresponding dot data is delayed by one-half DCLK. This half-dot shifting, when used on selected lines of character video, can be used to effect eyepleasing character rounding as shown in Figure 12.

ATTRIBUTE HIERARCHY

The video of each character block consists of four components as shown in Figure 13.

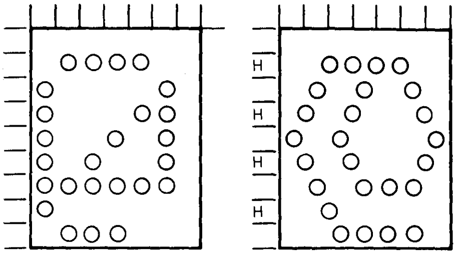
Symbol video is generated from the dot data inputs D0-D9. Underline video is enabled by the AUL attribute and is generated when the UL timing input is active. Underline and symbol video are always the same intensity.

Strike-thru video is enabled by the ALTPEN attribute and is generated when the LPL timing input is active. This video is always highlighted and takes precedence over the symbol and underline video. This feature applies to the MC2673A only.

Surround video is the absence of symbol, underline, and strike-thru video or the presence of the non-display attributes (ABLANK or ABLINK•BLINK).

The relative intensities of the four video components are determined by the remaining attributes (AHILT, ABLANK, ABLINK, ARVID) and the BKGND and CURSOR inputs are illustrated in Table 1.

FIGURE 12 — "AT" SYMBOL WITH AND WITHOUT HALF-DOT SHIFTING



H = Lines Shifted

FIGURE 13 — VIDEO COMPONENTS OF CHARACTER BLOCK

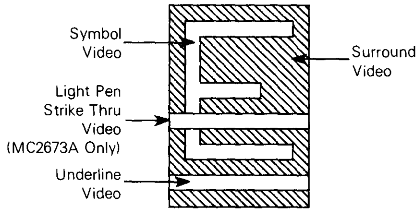


TABLE 1 — ATTRIBUTES HIERARCHY

Attributes and Control Inputs				Relative Video Intensities		
BKGND (See Note 1)	**Reverse (See Note 2)**	**Non-Display (See Note 3)**	**AHILT**	**Strike-Thru Video (See Figure 13)**	**Symbol or Underline Video (See Figure 13 and Note 4)**	**Surround Video (See Figure 13)**
0	0	0	0	W	G	B
0	0	0	1	W	W	B
0	0	1	d	B	B	B
0	1	0	0	B	G	W
0	1	0	1	B	B	W
0	1	1	d	W	W	W
1	0	0	0	B	G	W
1	0	0	1	B	B	W
1	0	1	d	W	W	W
1	1	0	0	W	G	B
1	1	0	1	W	W	B
1	1	1	d	B	B	B

d = don't care
W = white
B = black
G = gray

NOTES:
1. Reverse sense for VIDEO output.
2. Reverse = ARVID•$\overline{\text{CURSOR}}$ + $\overline{\text{ARVID}}$•CURSOR
3. Non-display = ABLANK + ABLINK•BLINK
4. Symbol and underline video are always the same intensity.

MOTOROLA

MC2674

Advance Information

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC)

The MC2674 advanced video display controller (AVDC) is a programmable device designed for use in CRT terminals and display systems that employ raster-scan techniques. The AVDC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the AVDC.

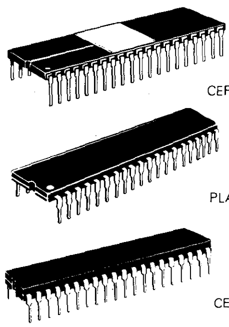
A minimum CRT terminal system configuration consists of an AVDC, an MC2671 keyboard and communication controller (PKCC), an MC2670 display character and graphics generator (DCGG), an MC2675 color/monochrome attributes controller (CMAC), a single-chip microcomputer such as the MC6809, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data buses.

- 4 MHz Character Rate
- 1 to 256 Characters Per Row
- 1 to 16 Raster Lines Per Character Row
- Bit Mapped Graphics Mode
- Programmable Horizontal and Vertical Sync Generators
- Interlaced or Non-Interlaced Operation
- Up to 64K RAM Address for Multiple-Page Operation
- Readable, Writeable, and Incrementable Cursor
- Programmable Cursor Size and Blink
- AC Line Lock
- Automatic Wraparound of RAM
- Automatic Split Screen
- Automatic Bidirectional Soft Scrolling
- Programmable Scan Line Increment
- Row Table Addressing Mode
- Double Height Tops and Bottoms
- Double Width Control Output
- Selectable Buffer Interface Modes
- Dynamic RAM Refresh
- Completely TTL Compatible
- Single +5-Volt Power Supply
- Power-On Reset Circuit
- Applications Include: CRT Terminals, Word Processing Systems, Small Business Computers, and Home Computers

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

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

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC)



L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT

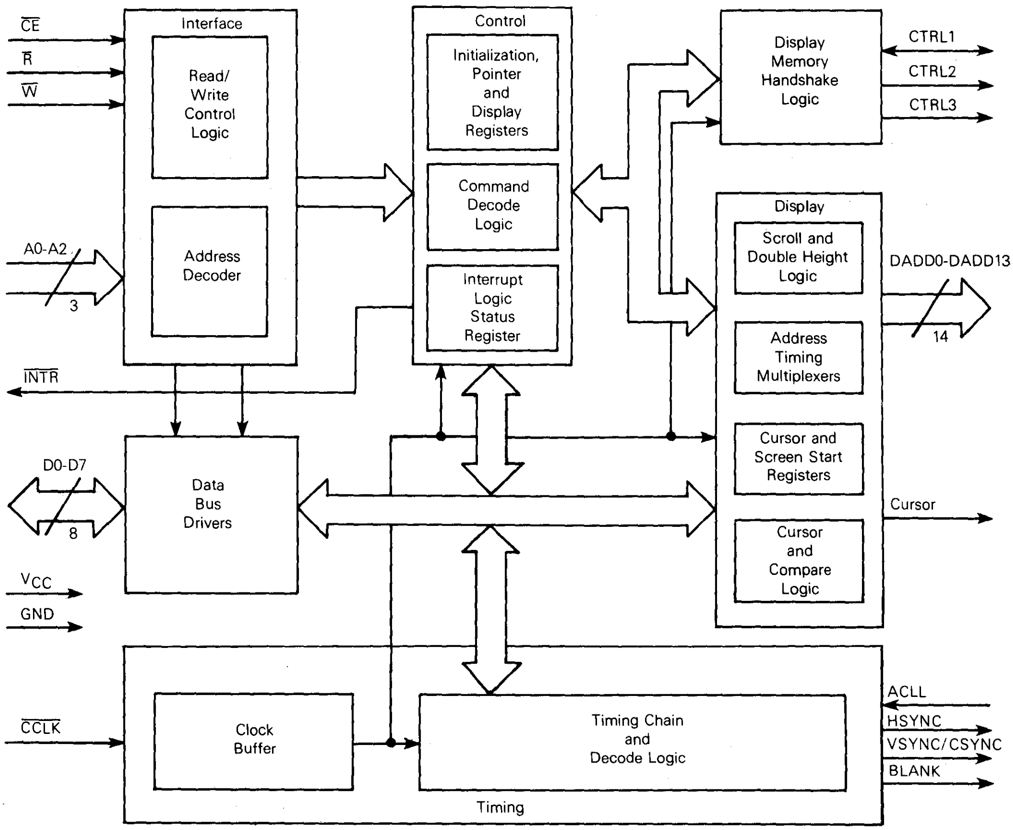
Signal	Pin	Pin	Signal
$\overline{R}$	1	40	V_{CC}
$\overline{CE}$	2	39	A2
$\overline{W}$	3	38	A1
CTRL1	4	37	A0
CTRL2	5	36	ACLL
CTRL3	6	35	$\overline{INTR}$
CURSOR	7	34	DADD0/LG
D0	8	33	DADD1/DADD14
D1	9	32	DADD2/DADD15
D2	10	31	DADD3/LR
D3	11	30	DADD4/LA0
D4	12	29	DADD5/LA1
D5	13	28	DADD6/LA2
D6	14	27	DADD7/LA3
D7	15	26	DADD8/FL
$\overline{CCLK}$	16	25	DADD9/DW
BLANK	17	24	DADD10/UL
VSYNC/CSYNC	18	23	DADD11/BLINK
HSYNC	19	22	DADD12/ODD
GND	20	21	DADD13/LL

3

ORDERING INFORMATION (V_{CC} = 5 V ± 5%, T_A = 0°C to 70°C)

Package Type	Frequency	Order Number
Plastic P Suffix	2.7 MHz 4.0 MHz	MC2674B3P MC2674B4P
Ceramic L Suffix	2.7 MHz 4.0 MHz	MC2674B3L MC2674B4L
Cerdip S Suffix	2.7 MHz 4.0 MHz	MC2674B3S MC2674B4S

BLOCK DIAGRAM



ABSOLUTE 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	Rating
Thermal Resistance			
Plastic Package	θ_{JA}	50	°C/W
Ceramic Package		50	

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

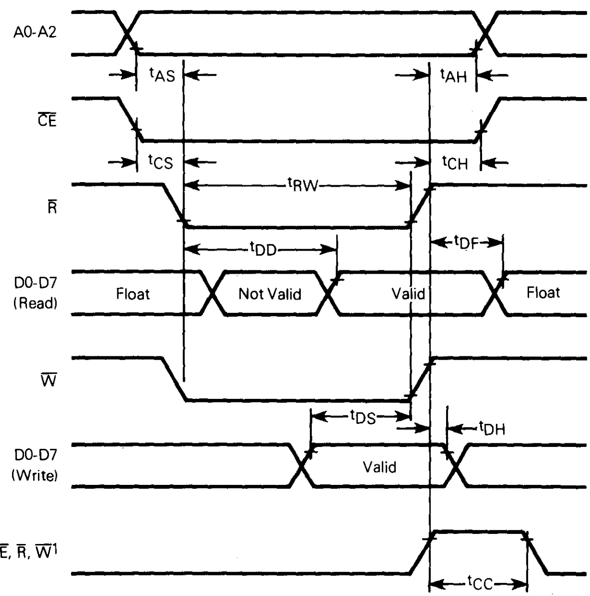
DC ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0 V ± 5%)

Parameter	Symbol	Min	Max	Unit
Input Low Voltage	V_{IL}	−0.3	0.8	V
Input High Voltage	V_{IH}	2.0	V_{CC}	V
Output Low Voltage (I_{OL} = 2.4 mA)	V_{OL}	–	0.4	V
Output High Voltage (Except $\overline{INTR}$ Output) (I_{OH} = −200 μA)	V_{OH}	2.4	–	V
Input Leakage Current (V_{in} = 0 to V_{CC})	I_{in}	−10	10	μA
Hi-Z (Off-State) Leakage Current (V_{CC} = 5.25 V, V_{in} = 0.4 to 2.4 V)	I_{TSI}	−10	10	μA
$\overline{INTR}$ Open-Drain Output Leakage Current (V_O = 0 to V_{CC})	I_{OD}	–	10	μA
Internal Power Dissipation (Measured at T_A = 0°C)	P_{INT}	–	800	mW

AC ELECTRICAL CHARACTERISTICS — BUS TIMING (T_A = 0°C to 70°C, V_{CC} = 5 V ± 5%)

Parameter	Symbol	2.7 MHz		4.0 MHz		Unit
		Min	Max	Min	Max	
A0-A2 Setup Time to $\overline{W}$, $\overline{R}$ Low	t_{AS}	30	–	30	–	ns
A0-A2 Hold Time from $\overline{W}$, $\overline{R}$ High	t_{AH}	0	–	0	–	ns
$\overline{CE}$ Setup Time to $\overline{W}$, $\overline{R}$ Low	t_{CS}	0	–	0	–	ns
$\overline{CE}$ Hold Time from $\overline{W}$, $\overline{R}$ High	t_{CH}	0	–	0	–	ns
$\overline{W}$, $\overline{R}$ Pulse Width	t_{RW}	250	–	200	–	ns
Data Valid after $\overline{R}$ Low	t_{DD}	–	200	–	200	ns
Data Bus Floating after $\overline{R}$ High	t_{DF}	–	100	–	100	ns
Data Setup Time to $\overline{W}$ High	t_{DS}	150	–	150	–	ns
Data Hold Time from $\overline{W}$ High	t_{DH}	10	–	5	–	ns
High Time from $\overline{CE}$ to $\overline{CE}$	t_{CC}					
Consecutive Commands		t_{CCP}	–	t_{CCP}	–	ns
Other Accesses		300	–	300	–	ns

BUS TIMING DIAGRAM



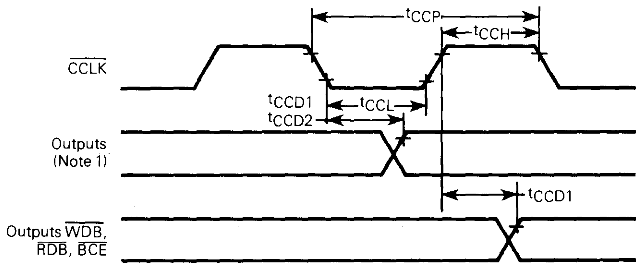
NOTES:
1. Any two must be high for t_{CC}.
2. All ac measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

AC ELECTRICAL CHARACTERISTICS — CHARACTER CLOCK ($\overline{CCLK}$) TIMING (T_A=0°C to 70°C, V_{CC}=5 V±5%)

Parameter	Symbol	2.7 MHz Min	2.7 MHz Max	4.0 MHz Min	4.0 MHz Max	Unit
$\overline{CCLK}$ Period	t_{CCP}	370	10000	250	10000	ns
$\overline{CCLK}$ High Time	t_{CCH}	125	–	100	–	ns
$\overline{CCLK}$ Low Time	t_{CCL}	125	–	100	–	ns
Output Delay Time from $\overline{CCLK}$ Edge						
DADD0-13, MBC	t_{CCD1}	40	175	40	150	ns
BLANK, HSYNC, VSYNC/CSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BACK}$, $\overline{BCE}$, $\overline{WDB}$, $\overline{RDB}$*	t_{CCD2}	40	225	40	200	ns

* $\overline{BCE}$, $\overline{WDB}$, and $\overline{RDB}$ delays track each other within 10 nanoseconds. Also, these output delays will tend to follow direction (minimum/maximum) of DADD0-DADD13 delays.

CCLK TIMING DIAGRAM



NOTES:
1. DADD0-DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, BACK.
2. $\overline{BCE}$ changes state on both $\overline{CCLK}$ edges.
3. All ac measurement points shown are 0.8 V to 2.0 V, unless otherwise specified.

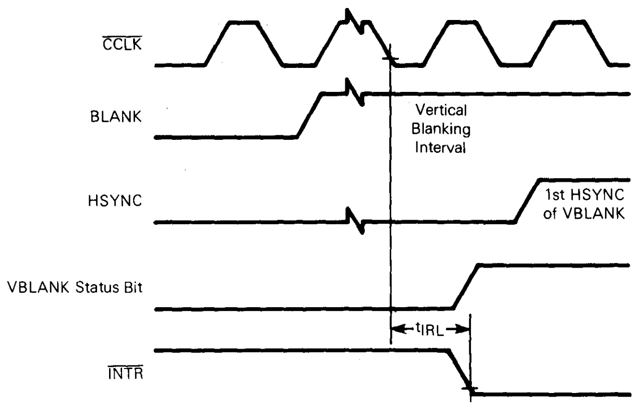
AC ELECTRICAL CHARACTERISTICS — OTHER TIMING (T_A=0°C to 70°C, V_{CC}=5 V±5%)

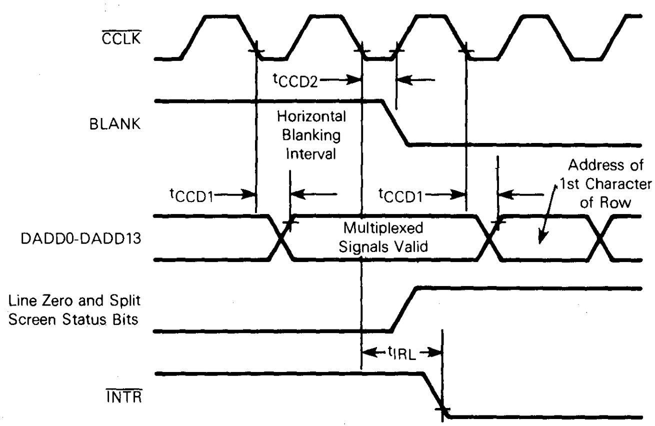
Parameter	Symbol	2.7 MHz Min	2.7 MHz Max	4.0 MHz Min	4.0 MHz Max	Unit
READY/RDFLG Low from $\overline{W}$ High*	t_{RDL}	t_{CCP}+30	–	t_{CCP}+30	–	ns
$\overline{BACK}$ High from $\overline{PBREQ}$ Low	t_{BAK}	225	–	200	–	ns
$\overline{BEXT}$ High from $\overline{PBREQ}$ High	t_{BXT}	225	–	200	–	ns
$\overline{INTR}$ Low from $\overline{CCLK}$ Low	t_{IRL}	225	–	200	–	ns
$\overline{INTR}$ High from $\overline{W}$, $\overline{R}$ High*	t_{IRH}	600	–	600	–	ns
ACLL from HSYNC	t_{AC}	3× t_{CCP}	–	3× t_{CCP}	–	ns

* Timing is illustrated and specified referenced to $\overline{W}$ and $\overline{R}$ inputs. Device may also be operated with $\overline{CE}$ as the "strobing" input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.

3

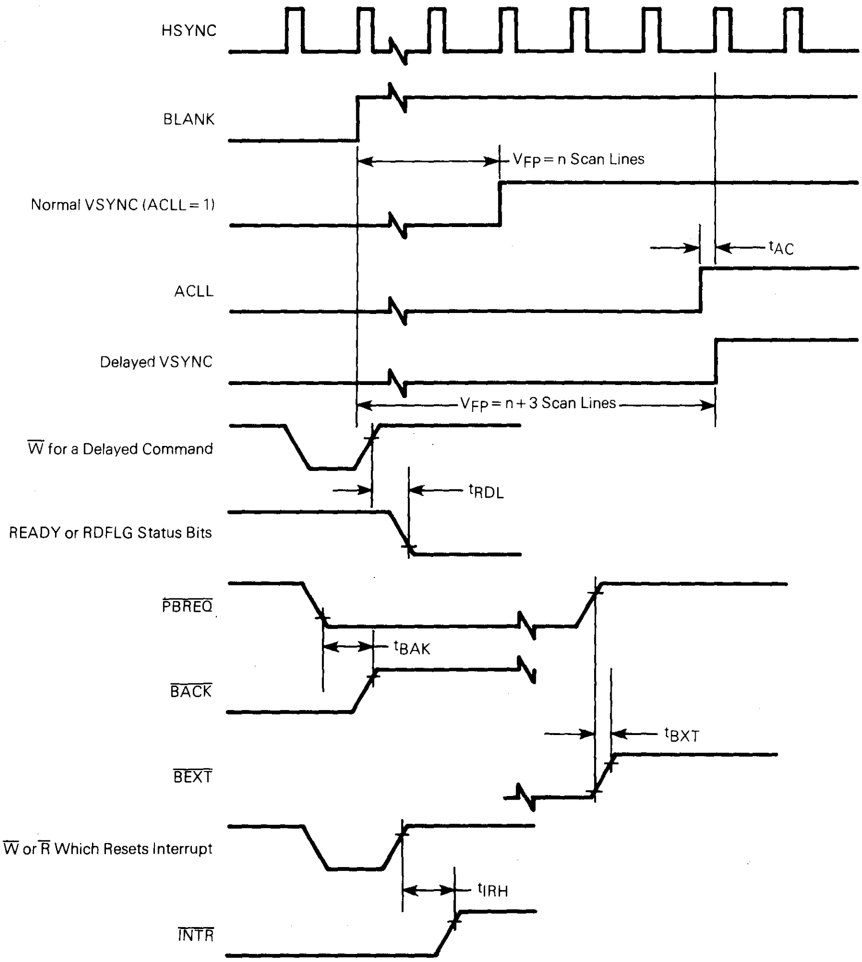
OTHER TIMING DIAGRAMS (Sheet 2 of 2)





NOTE: All ac measurement points shown are 0.8 V to 2.0 V, unless otherwise specified.

OTHER TIMING DIAGRAMS (Sheet 2 of 2)

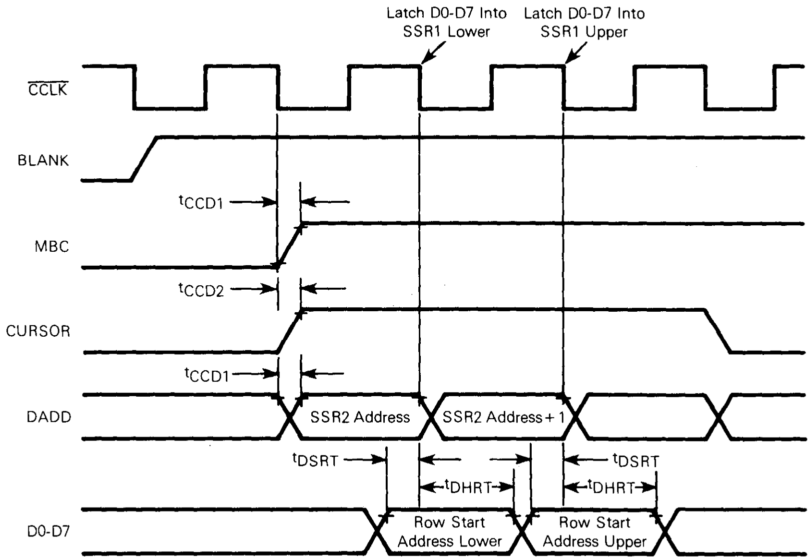


NOTE: All ac measurement points are 0.8 V to 2.0 V, unless otherwise specified.

AC ELECTRICAL CHARACTERISTICS — ROW TABLE INPUT TIMING (T_A = 0°C to 70°C, V_{CC} = 5 V ± 5%)

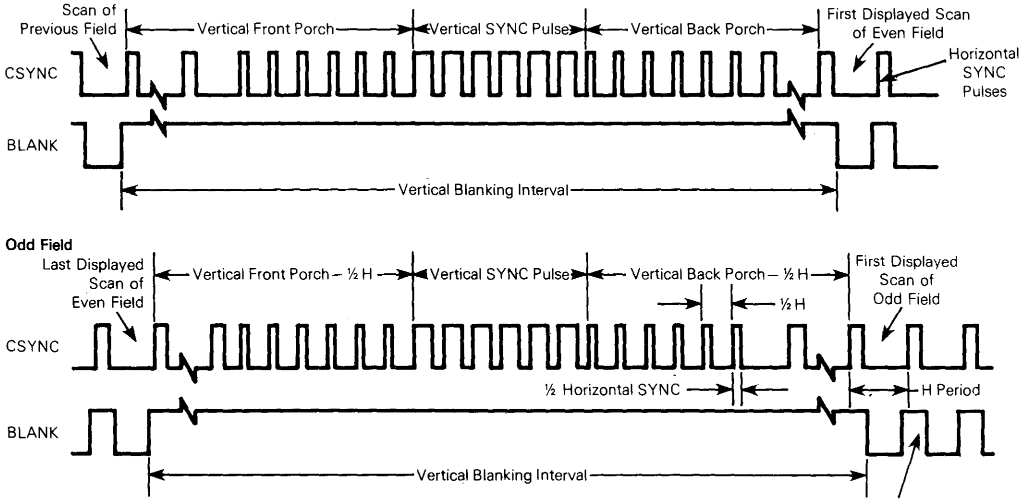
Parameter	Symbol	2.7 MHz Min	2.7 MHz Max	4.0 MHz Min	4.0 MHz Max	Unit
Data Setup Time to $\overline{CCLK}$ Low	t_{DSRT}	100	–	60	–	ns
Data Hold Time from $\overline{CCLK}$ Low	t_{DHRT}	60	–	60	–	ns

ROW TABLE FETCH I/O TIMING DIAGRAM



NOTE: All ac measurement points are 0.8 V to 2.0 V, unless otherwise specified.

COMPOSITE SYNC TIMING DIAGRAM



NOTES:
1. In non-interlaced operation the even field is repeated continuously. 2. In interlaced operation the even field alternates with the odd field.

SIGNAL DESCRIPTION

The input and output signals for the AVDC are described in the following paragraphs.

ADDRESS LINES (A0-A2)

These input lines are used to select AVDC internal register for read/write operations and for commands.

DATA BUS (D0-D7)

The 8-bit bidirectional three-state data bus controls all data, command, and status transfers between the CPU and the AVDC. Bit 0 is the least significant bit and bit 7 is the most significant bit. The direction of the transfer is controlled by the read ($\overline{R}$) and write ($\overline{W}$) inputs when chip enable ($\overline{CE}$) input is low. When the $\overline{CE}$ input is high, the data bus is in the three-state condition.

READ STROBE ($\overline{R}$)

This pin is an active low input. A low on this pin while $\overline{CE}$ is low causes the contents of the register selected by the address lines to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{R}$.

WRITE STROBE ($\overline{W}$)

This is an active low input. A low on this pin while $\overline{CE}$ is also low causes the contents of the data bus to be transferred to the register selected by the address lines. The transfer occurs on the trailing (rising) edge of $\overline{W}$.

CHIP ENABLE ($\overline{CE}$)

This is an active low input. When low, data transfers between the CPU and the AVDC are enabled on the data bus as controlled by the write strobe, read strobe, and address lines. When $\overline{CE}$ is high, effectively, the AVDC is isolated from the data bus and D0-D7 are placed in the three-state condition.

CHARACTER CLOCK ($\overline{CCLK}$)

This input is the timing signal derived from the video dot clock which is used to synchronize the AVDC's timing functions.

HORIZONTAL SYNC (HSYNC)

This active high output provides video horizontal sync pulses. The timing parameters are programmable.

VERTICAL SYNC/COMPOSITE SYNC (VSYNC/CSYNC)

A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.

BLANK (BLANK)

This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on display addresses DADD0 and DADD3 through DADD13 are valid on the trailing edge of BLANK.

CURSOR GATE (CURSOR)

This output becomes active for a specified number of scan lines when the address continued in the cursor register matches the address output on DADD0 through DADD13 for displayable character addresses. The first and last lines of the cursor and a blink option are programmable. When the row table addressing mode is enabled, this output is active for a portion of the blanking interval prior to the first scan line of a character row, while the AVDC is fetching the starting address for that row.

INTERRUPT REQUEST ($\overline{INTR}$)

This is an open-drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after a power-on reset or a master reset command.

AC LINE LOCK (ACLL)

If this input is low after the programmed vertical front porch interval, the vertical front porch will be lengthened by increments of horizontal scan line times until this input goes high.

HANDSHAKE CONTROL 1 (CTRL1)

In independent mode, provides an active low write data buffer ($\overline{WDB}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{PBREQ}$) input which indicates that the CPU desires to access the display memory.

HANDSHAKE CONTROL 2 (CTRL2)

In independent mode, provides an active low read data buffer ($\overline{RDB}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{BEXT}$) output which indicates that the AVDC has relinquished control of the display memory (DADD0-DADD13 are in the three-state condition) in response to a CPU bus request. $\overline{BEXT}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{BREQ}$) output which halts the CPU during a line DMA.

HANDSHAKE CONTROL 3 (CTRL3)

In independent mode, provides the active low buffer chip enable ($\overline{BCE}$) signal to the display memory. In transparent and shared modes, provides an active low bus acknowledge ($\overline{BACK}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.

DISPLAY ADDRESS (DADD0-DADD13)

These outputs are used by the AVDC to address up to 16K of display memory directly, or to 64K of memory by demultiplexing DADD14 and DADD15. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD0 through DADD13 and are valid at the trailing edge of BLANK. The following paragraphs describes the control signals.

LINE GRAPHICS (DADD0/LG) — This is the output which denotes bit-mapped graphics mode.

DISPLAY ADDRESS 14 (DADD1/DADD14) — This is the multiplexed address bit used to extend addressing to 64K.

DISPLAY ADDRESS 15 (DADD2/DADD15) — This is the multiplexed address bit used to extend addressing to 64K.

LAST ROW (DADD3/LR) — This is the output which indicates the last active character row of each field.

LINE ADDRESS (DADD4-DADD7/LA0-LA3) — These outputs provide the number of the current scan line count for each character row.

FIRST LINE (DADD8/FL) — This output is asserted during the blanking interval just prior to the first scan line of each character row.

DOUBLE WIDTH (DADD9/DW) — This output denotes a double width character row.

UNDERLINE (DADD10/UL) — This output is asserted during the blanking interval just prior to the scan line which matches the programmed underline position (line 0 through 15).

BLINK FREQUENCY (DADD11/BLINK) — Blink frequency provides an output divided down from the vertical sync rate.

ODD FIELD (DADD12/ODD) — This active high signal is asserted before each scan line of the odd field when interlace is specified. Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications.

LAST LINE (DADD13/LL) — This output is asserted during the blanking interval just prior to the last scan line of each character row.

V_{CC} AND GND

Power is supplied to the AVDC using these two pins. V_{CC} is the +5 volts ±5% power input and GND is the ground connection.

FUNCTIONAL DESCRIPTION

As shown in the block diagram, the AVDC contains the following major blocks: data bus buffer, interface logic, operation control, timing, display control, and buffer control. The major blocks are described in the following paragraphs.

DATA BUS BUFFER

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the AVDC.

INTERFACE LOGIC

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data buffer. The functions performed by the CPU read and write operations are shown in Table 1.

OPERATION CONTROL

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

TIMING

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

DISPLAY CONTROL

The display control section generates linear addressing of up to 16K bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning and address comparisons required for generation of timing signals, double-height tops and bottoms, smooth scrolling, and the split-screen interrupts.

BUFFER CONTROL

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described in **SYSTEM CONFIGURATIONS.**

TABLE 1 — AVDC ADDRESSING

A2	A1	A0	Read ($\overline{R}$=0)	Write ($\overline{W}$=0)
0	0	0	Interrupt Register	Initialization Registers*
0	0	1	Status Register	Command Register
0	1	0	Screen Start 1 Lower Register	Screen Start 1 Lower Register
0	1	1	Screen Start 1 Upper Register	Screen Start 1 Upper Register
1	0	0	Cursor Address Lower Register	Cursor Address Lower Register
1	0	1	Cursor Address Upper Register	Cursor Address Upper Register
1	1	0	Screen Start 2 Lower Register	Screen Start 2 Lower Register
1	1	1	Screen Start 2 Upper Register	Screen Start 2 Upper Register

*There are 15 initialization registers which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for additional accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the MC2670, MC2671, MC2674, and MC2675 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The AVDC supports four common system configurations of display-buffer memory, designated the independent, transparent, shared, and row-buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row-buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user programs IR0 bits 0 and 1 to select the mode best suited for the system environment. The CTRL1, CTRL2, and CTRL3 outputs perform different functions for each mode and are named accordingly in the description of each mode. mode.

INDEPENDENT MODE

The CPU-to-RAM interface configuration for this mode is illustrated in Figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not require address multiplexers. The CPU does not address the memory directly – the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The AVDC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

The CPU manages the data transfers by supplying commands to the AVDC. The commands used are:

1. Read/write at pointer address,
2. Read/write at cursor address (with optional increment of address), and
3. Write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues "write at cursor with/without increment" or "write at pointer" command.
5. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. AVDC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1. and 3. as above.
2. CPU issues "read at cursor with/without increment" or "read at pointer" command.
3. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and AVDC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the "write from cursor to pointer" command:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues "write from cursor to pointer" command.
5. AVDC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. AVDC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously asserted delayed command has been completed.

Two timing sequences are possible for the "read/write at cursor/pointer" commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in Figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately five character clocks (see Figure 4).

Timing for the "write from cursor to pointer" operation is shown in Figure 5. The memory is filled at a rate of one location per two character times. The command will execute only during blanking intervals and may require many horizontal or vertical blanking intervals to complete. Additional delayed commands can be asserted immediately after this command has completed.

Immediately commands can be asserted at any time regardless of the state of the ready state/interrupt.

3

FIGURE 1 — CRT TERMINAL BLOCK DIAGRAM

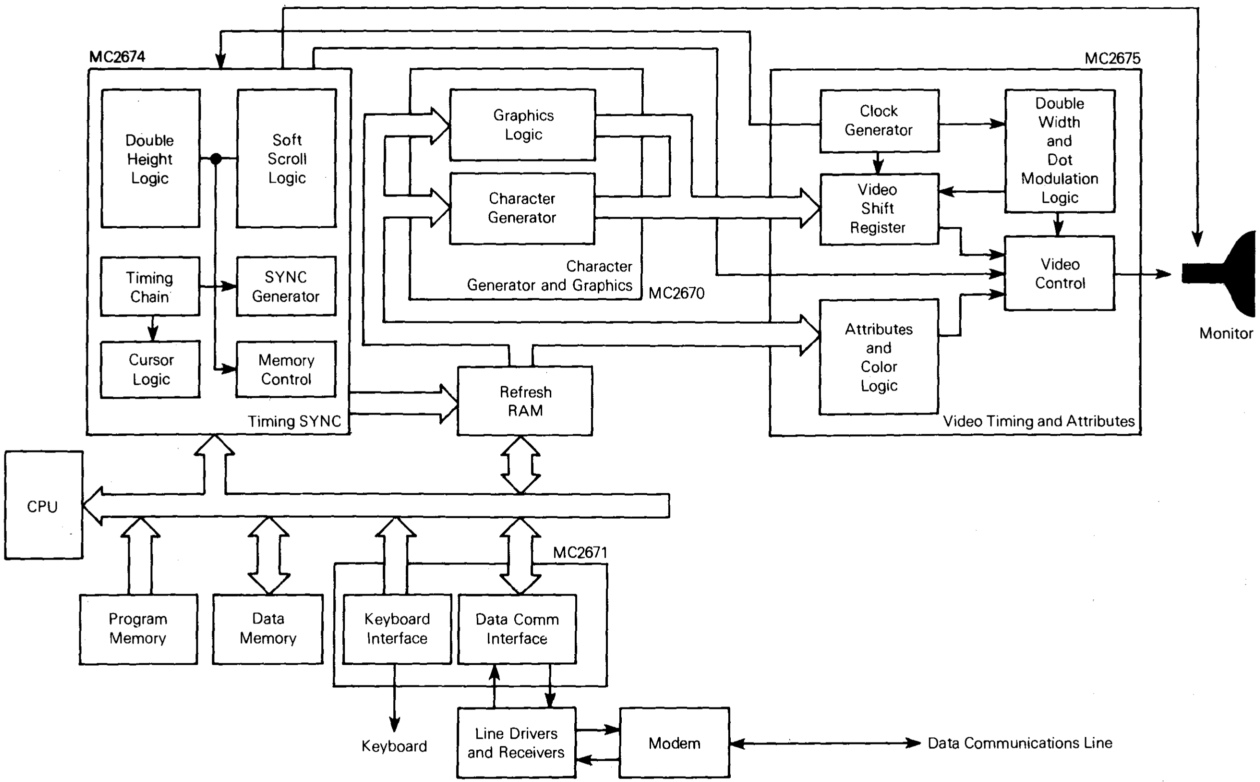


FIGURE 2 — INDEPENDENT BUFFER MODE CONFIGURATION

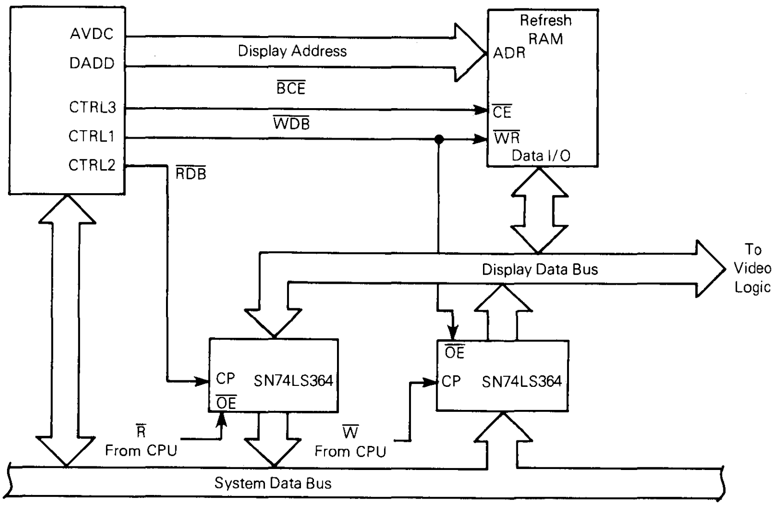
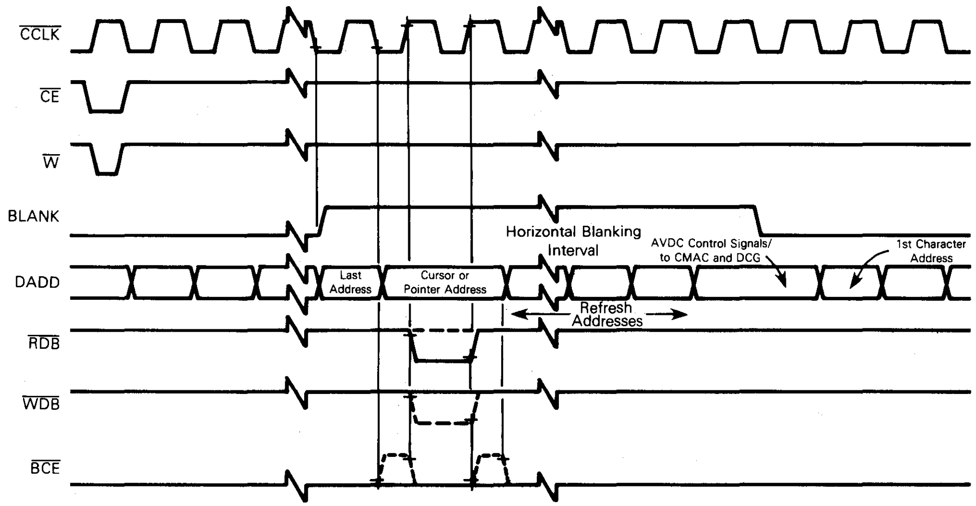


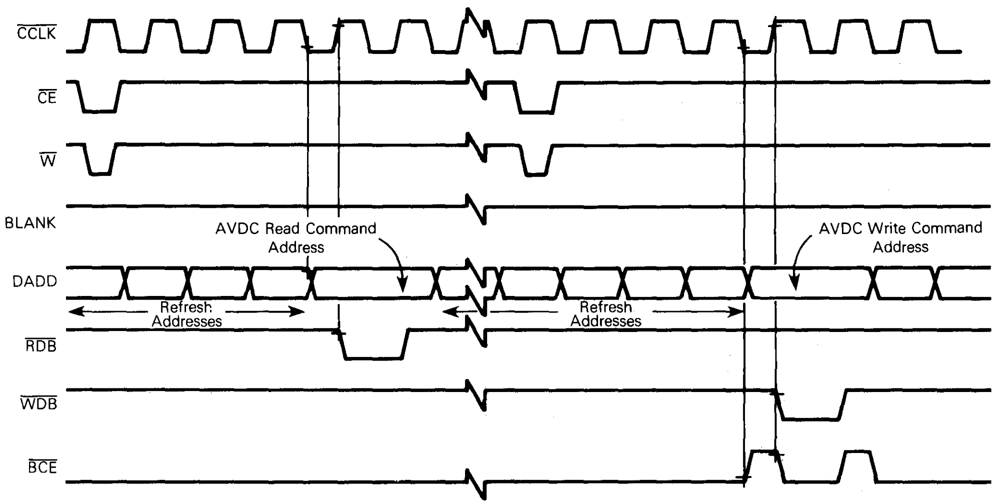
FIGURE 3 — READ/WRITE AT CURSOR/POINTER COMMAND TIMING
(Command Received During Active Display Window)



NOTES:

1. Write waveforms shown in dotted lines.
2. If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated.
3. Measurement points shown at 0.8 V to 2.0 V, unless otherwise noted.

FIGURE 4 — READ/WRITE AT CURSOR/POINTER COMMAND TIMING
(Command Received While Display Is Blanked)



NOTE: Measurement points shown at 0.8 V to 2.0 V, unless otherwise noted.

SHARED AND TRANSPARENT BUFFER MODES

In these modes, the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see Figure 6). The processor bus request ($\overline{\text{PBREQ}}$) control signal informs the AVDC that the CPU is requesting access to the display buffer. In response to this request, the AVDC raises bus acknowledge ($\overline{\text{BACK}}$) until its bus external ($\overline{\text{BEXT}}$) output has freed the display address and data buses for CPU access. $\overline{\text{BACK}}$, which can be used as a "hold" input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the AVDC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the AVDC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in Figures 7, 8, and 9.

ROW BUFFER MODE

Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the AVDC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The AVDC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The control signal $\overline{\text{BREQ}}$ informs the CPU that character addresses and the MBC signal will start at the next falling edge of BLANK. The CPU must release the address and data buses before this time to prevent bus contention. After the row of character data is transferred to the CPU, $\overline{\text{BREQ}}$ returns high to grant memory control back to the CPU.

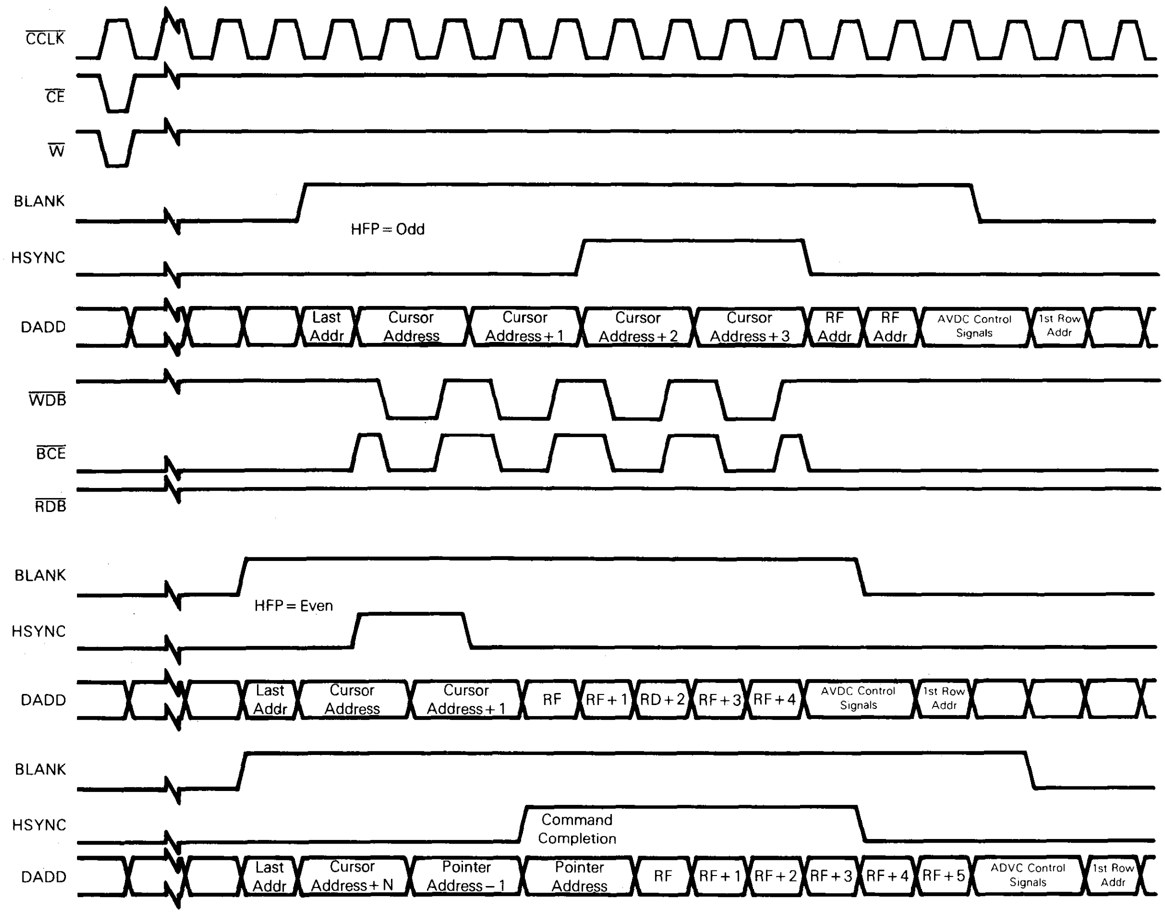
ROW TABLE ADDRESS MODE

In this mode, each character row in the screen image memory has a unique starting address. This provides greater flexibility with respect to screen operations, such as editing, than the sequential addressing mode. The row table, Figure 12, is a list of starting addresses for each character row and may reside anywhere in the AVDC's addressable memory space. Each entry in the table consists of two bytes: the first byte contains the eight least significant bits of the row starting address and the second byte contains, in its six least significant bits, the six most significant bits of the row starting address. The function of the two most significant bits of the second byte is selected by programming IR0[7]. They may be used either as row attribute bits to control double width and double height for that character row, or as an additional two address bits to extend the usable display memory to 64K.

The first address of the row table operation is designated in screen start register 2 (SSR2). If row table addressing is enabled via IR2[7], the AVDC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row, while simultaneously incrementing the contents of SSR2 by two so as to point to the next table entry. The fetching of the row starting address from the row table is indicated by the assertion of the CURSOR output during BLANK. The address read from the table by the AVDC is loaded into screen start register 1 (SSR1) for use internally. Since the contents of SSR2 changes as the table entries are fetched, it must be reinitialized to point to the first table entry during each vertical retrace interval.

Row table addressing is intended primarily for use in conjunction with the row buffer mode of operation and requires no additional circuitry in that case. It may also be used with

FIGURE 5 — WRITE FROM CURSOR TO POINTER COMMAND TIMING



NOTE:
If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated.

FIGURE 6 — AVDC SHARED OR TRANSPARENT BUFFER MODES

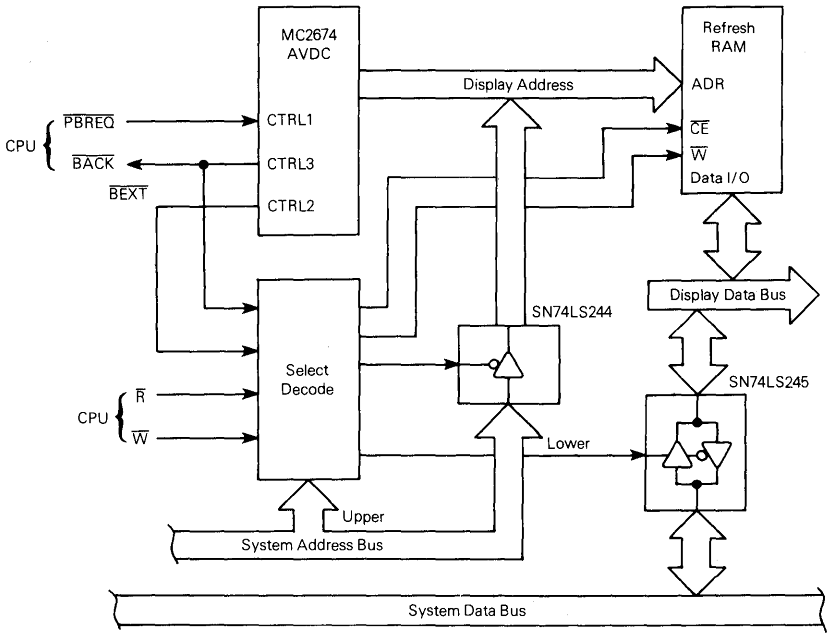
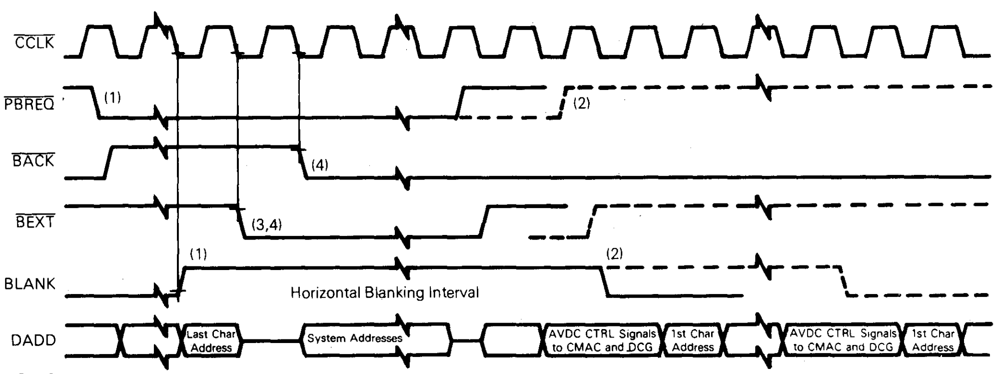


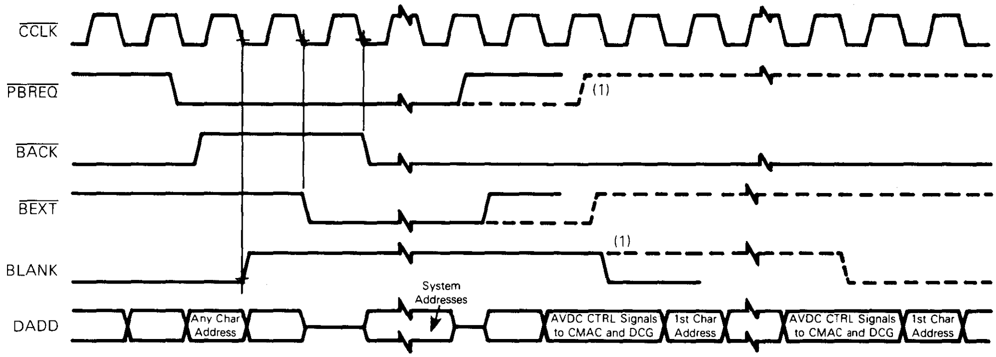
FIGURE 7 — TRANSPARENT BUFFER MODE TIMING



NOTES:

1. $\overline{PBREQ}$ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If $\overline{PBREQ}$ is negated after the next to last $\overline{CCLK}$ of the horizontal blanking interval, the next scan line will also be blanked.
3. Accesses during vertical blank or "display off" are granted only at the beginning of the horizontal front porch.
4. If row table addressing is enabled, CPU access is delayed by two character clocks prior to the first scan line of each character row.
5. Measurement points shown at 0.8 V to 2.0 V, unless otherwise noted.

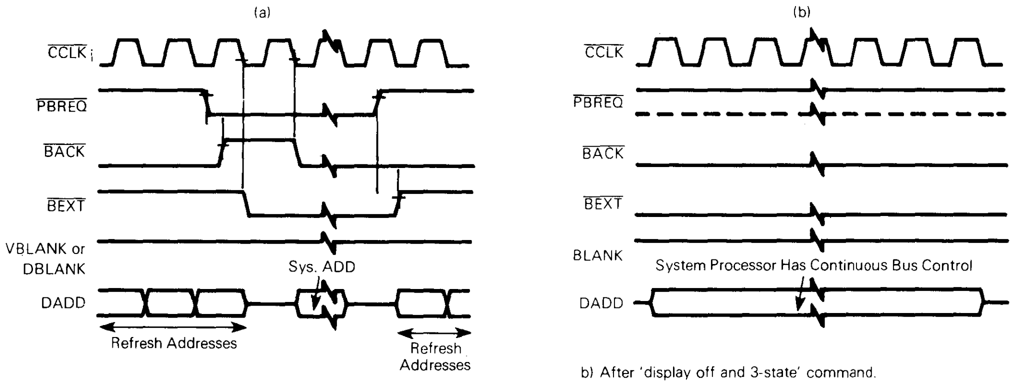
FIGURE 8 — SHARED BUFFER MODE TIMING



NOTES:

1. If $\overline{PBREQ}$ is negated after the next to last $\overline{CCLK}$ of the horizontal blanking interval, the next scan line will also be blanked.
2. Measurement points shown at 0.8 V to 2.0 V, unless otherwise noted.

FIGURE 9 — SHARED AND TRANSPARENT MODE TIMING



b) After 'display off and 3-state' command.

a) During Vertical Blank or after 'display off' command in shared mode only. See Figure 7 for transparent timing.

NOTE: Measurment points shown at 0.8 V to 2.0 V, unless otherwise noted.

the other modes, but circuitry must be added to route the data from the display memory to the data bus inputs of the AVDC. Additionally, when not operating in row buffer mode, care must be taken to assure that the CPU does not attempt to access the AVDC while it is reading the row table. One way of preventing this is to latch prior to reading or writing the AVDC. The AVDC should only be accessed if the latch is low, indicating that the last line of the row is not active.

Figure 13 illustrates a typical hardware implementation for use in conjunction with independent and transparent modes, and Figure 14 shows the timing for row table operation.

OPERATION

After power is applied, the AVDC will be in an inactive state. Two consecutive "master reset" commands are necessary to release this circuitry and ready the AVDC for operation. Two register groups exist within the ADC; the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, pointer address, scrolling region, double height and width condition, and screen format. These are loaded first and normally require no modification except for certain special visual

FIGURE 10 — ROW BUFFER MODE CONFIGURATION

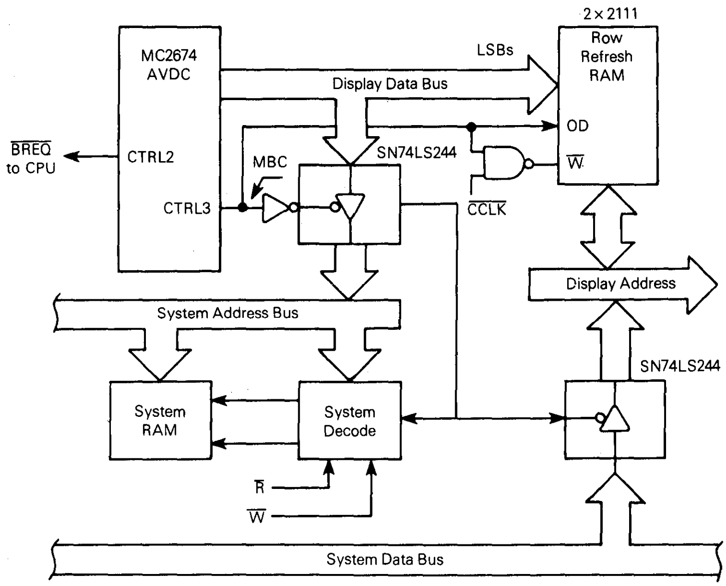
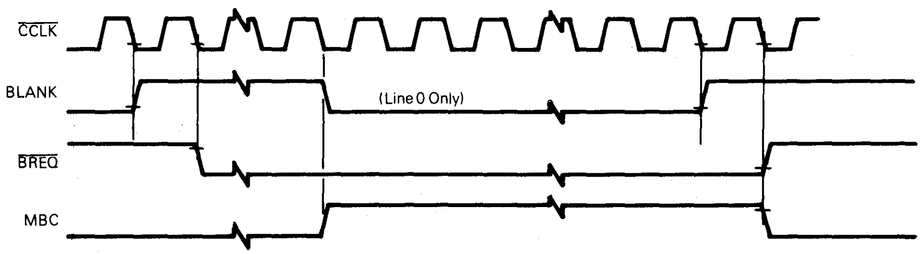


FIGURE 11 — ROW BUFFER MODE TIMING



NOTES:
1. If row table addressing is enabled, $\overline{BREQ}$ will be asserted at the middle of the last scan line of the prior row, and MBC will be asserted at the beginning of BLANK.
2. Measurement points shown at 0.8 V to 2.0 V, unless otherwise noted.

FIGURE 12 — ROW TABLE ADDRESS FORMAT

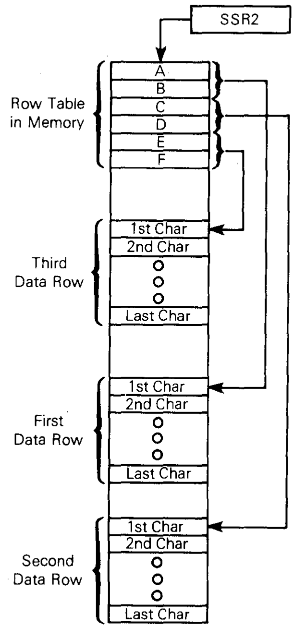


FIGURE 13 — ROW TABLE MODE CONFIGURATION (NON-ROW BUFFER MODES)

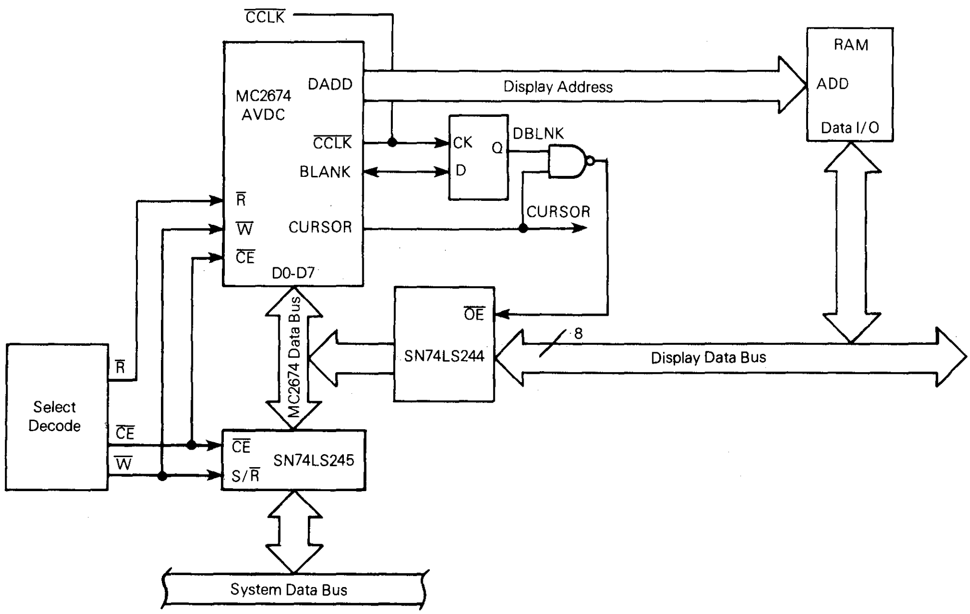
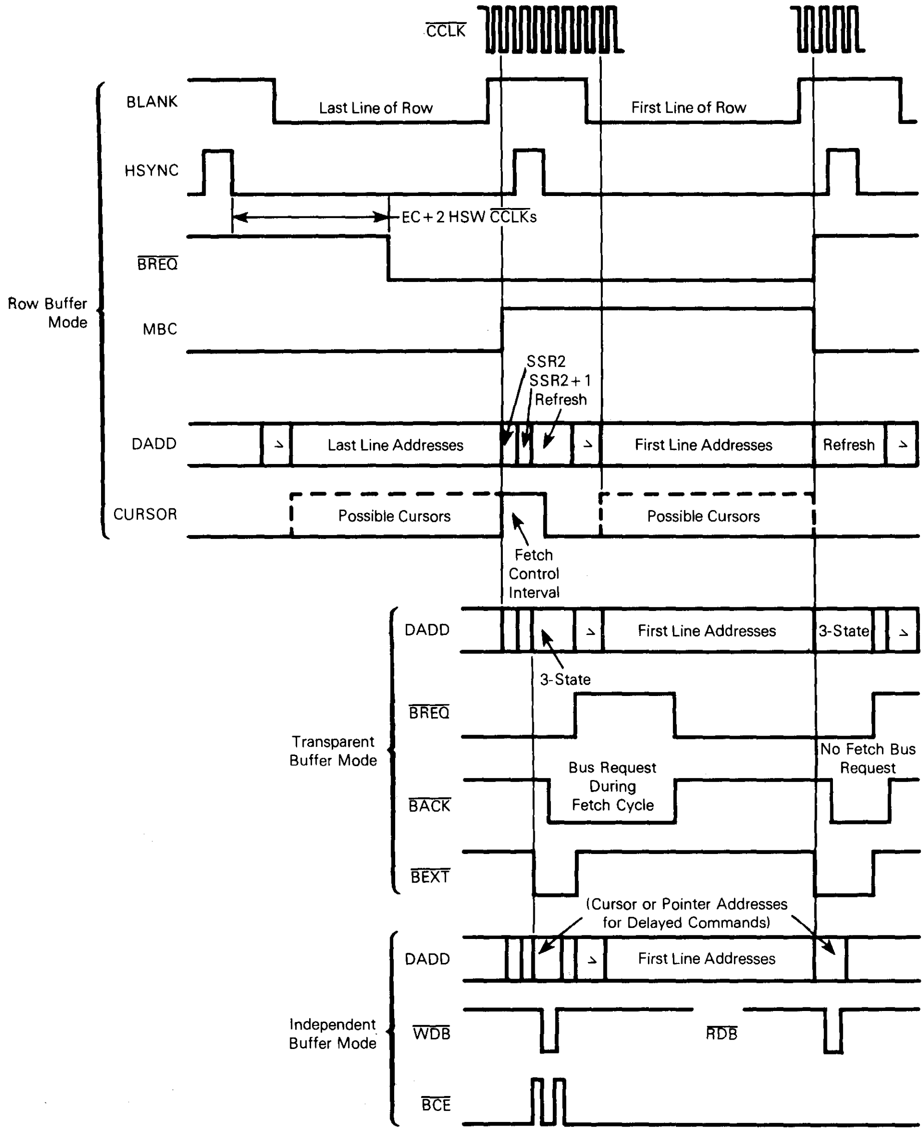


FIGURE 14 — ROW TABLE MODE TIMING



⦣ = Multiplexed Control Signals
EC = Equalizing Constant
HSW = Horizontal SYNC Width

effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the split screen addresses associated with the scrolling area or an alternate memory. These may require modification during operation.

After initial loading of the two register groups, the ADC is ready to control the monitor screen. Prior to executing the AVDC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the AVDC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the AVDC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the AVDC supply the "handshaking" information necessary for the CPU to effect real time display changes in the proper time frame if required.

INITIALIZATION REGISTERS

There are 15 initialization registers (IR0-IR14) which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for further accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the "load IR address pointer" command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in Figure 15.

FIGURE 15 — INITIALIZATION REGISTER FORMATS (Sheet 1 of 4)

IR0

7	6–3		2	1–0
Double Height/ Width	Scan Lines Per Character Row		Sync Select	Buffer-Mode Select
	Non-Interlaced	Interlaced		
	0000 = 1 Line 0001 = 2 Lines 0010 = 3 Lines • • 1110 = 15 Lines 1111 = 16 Lines	0000 = 2 Lines 0001 = 4 Lines 0010 = 6 Lines • • 1110 = 30 Lines 1111 = Undefined	0 = VSYNC 1 = CSYNC	00 = Independent 01 Transparent 10 = Shared 11 = Row

IR1

7	6–0
Interlace Enable	Equalizing Constant
0 = Non-Bit 1 = Interrupt	0000000 = 1 $\overline{CCLK}$ 0000001 = 2 $\overline{CCLK}$ • • 1111110 = 127 $\overline{CCLK}$ 1111111 = 128 $\overline{CCLK}$ Calculated from: $EC = 0.5\,(H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}) - 2(H_{SYNC})$

IR2

7	6–3	2–0
Row Table	Horizontal Sync Width	Horizontal Back Porch
0 = Off 1 = On	0000 = 2 $\overline{CCLK}$ 0001 = 4 $\overline{CCLK}$ • • 1110 = 30 $\overline{CCLK}$ 1111 = 32 $\overline{CCLK}$	000 = Not Allowed 001 = 3 $\overline{CCLK}$ • • 110 = 23 $\overline{CCLK}$ 111 = 27 $\overline{CCLK}$

FIGURE 15 — INITIALIZATION REGISTER FORMATS (Sheet 2 of 4)

IR3

7 6 5	4 3 2 1 0
Vertical Front Porch	Vertical Back Porch
000 = 4 Scan Lines 001 = 8 Scan Lines • • 110 = 28 Scan Lines 111 = 32 Scan Lines	00000 = 4 Scan Lines 00001 = 6 Scan Lines • • 11110 = 64 Scan Lines 11111 = 66 Scan Lines

IR4

7	6 5 4 3 2 1 0
Character Blink Rate	Active Character Rows Per Screen
0 = 1/64 VSYNC 1 = 1/128 VSYNC	0000000 = 1 Row 0000001 = 2 Rows • • • 1111110 = 127 Rows 1111111 = 128 Rows

IR5

7 6 5 4 3 2 1 0
Active Characters Per Row
00000010 = 3 Characters 00000011 = 4 Characters • • 11111110 = 255 Characters 11111111 = 256 Characters

IR6

7 6 5 4	3 2 1 0
First Line of Cursor	Last Line of Cursor
0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15	0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15

IR7

7 6	5	4	3 2 1 0
Light Pen Line	Cursor Blink	Cursor Rate	Underline Position
00 = Scan Line 3 01 = Scan Line 1 10 = Scan Line 5 11 = Scan Line 7	0 = Off 1 = On	0 = 1/32 1 = 1/64	0000 = Scan Line 0 0001 = Scan Line 1 • • 1110 = Scan Line 14 1111 = Scan Line 15

FIGURE 15 — INITIALIZATION REGISTER FORMATS (Sheet 3 of 4)

IR8

7	6	5	4	3	2	1	0
Display Buffer First Address LSBs							
H'000' = 0 H'001' = 1 • • H'FFE' = 4,094 H'FFF' = 4,095 NOTE: MSBs are in IR9[3:0]							

IR9

7	6	5	4	3	2	1	0
Display Buffer Last Address				Display Buffer First Address MSBs			
0000 = 1,023 0001 = 2,047 • • 1110 = 15,359 1111 = 16,383				See IR8			

IR10

7	6	5	4	3	2	1	0
Display Pointer Address Lower							
See IR11							

IR11

7	6	5	4	3	2	1	0
LZ Down	LZ Up	Display Pointer Address Upper					
0 = Off 1 = On	0 = Off 1 = On	H'0000' = 0 H'0001' = 1 • • H'3FFF' = 16,383					

IR12

7	6	5	4	3	2	1	0
Scroll End	Split Register 1						
0 = Off 1 = On	0000000 = Row 1 0000001 = Row 2 • • 1111111 = Row 128						

FIGURE 15 — INITIALIZATION REGISTER FORMATS (Sheet 4 of 4)

	Bit 7	Bits 6–0
IR13	Scroll End	Split Register 2
	0 = Off 1 = On	0000000 = Row 1 0000001 = Row 2 • • 1111111 = Row 128

	Bits 7–6	Bits 5–4	Bits 3–0
IR14	Double 1	Double 2	Lines to Scroll
	00 = Normal 01 = Double Width 10 = Double Width and Tops 11 = Double Width and Bottoms	00 = Normal 01 = Double Width 10 = Double Width and Tops 11 = Double Width and Bottoms	0000 = 1 0001 = 2 • • 1110 = 15 1111 = 16

DOUBLE HEIGHT/WIDTH ENABLE (IR0[7]) — When this bit is set, the value in IR14[7:6] is used to control the double height and width conditions of each character row. Assertion of this bit also allows IR14[7:6] to be programmed in two ways:

1. By the CP writing to IR14 directly.
2. When the contents of screen start register 1 (SSR1) upper are changed, either by the CPU writing to this register or by the automatic loading of SSR1 when operating in row table mode, the two most significant bits of SSR1 upper are copied into IR14[7:6]. Thus, the most significant bits of each row table entry can be used to control double height and double width attributes on a row-by-row basis.

IR14[5:4] are not active when this bit is set. When this bit is reset, the double height and width attributes operate as described in IR[14].

SCAN LINES PER CHARACTER ROW (IR0[6:3]) — Both interlaced and non-interlaced scanning are supported by the AVDC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the AVDC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0-LA3 and ODD pins.

VSYNC/CSYNC (IR0[2]) — This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

BUFFER MODE SELECT (IR0[1:0]) — Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See SYSTEM CONFIGURATIONS.

INTERLACE ENABLE (IR1[7]) — Specifies interlaced or non-interlaced timing operation. Two modes of interlaced operation are available, depending on whether L0-L3 or ODD, L0-L2 are used as the line address for the character generator. The resulting displays are shown in Figure 16.

For "interlaced sync" operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The AVDC outputs successive line numbers in ascending order on the LA0-LA3 lines, one per scan line for each field.

The "interlaced sync and video" format doubles the character density on the screen. The AVDC outputs successive line numbers in ascending order on the odd and LA0-LA2 lines, one per scan line for each field.

EQUALIZING CONSTANT (IR1[6:0]) — This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{\text{CCLK}}$s) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{SP}}{2} - 2\,(H_{SYNC})$$

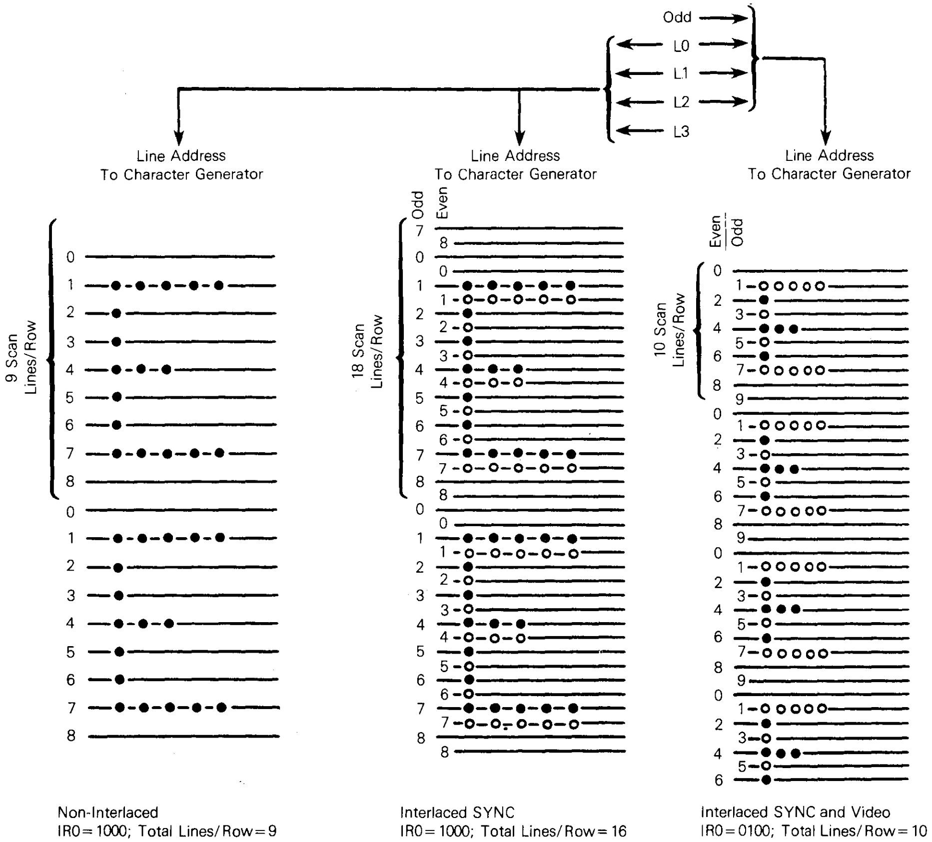
The definition of the individual parameters is illustrated in Figure 17.

Note that when using the MC2675 CMAC, it will delay the blank pulse three $\overline{\text{CCLK}}$s relative to the HSYNC pulse.

ROW TABLE MODE ENABLE (IR2[7]) — Assertion/negation of this bit causes the AVDC to begin/terminate operating in row table mode starting at the next character row. See **ROW TABLE ADDRESS MODE.** By using the split interrupt capability of the AVDC, this mode can be enabled and disabled on a particular character row. This allows a combination of row table and sequential addressing to be utilized to provide maximum flexibility in generating the display.

HORIZONTAL SYNC PULSE WIDTH (IR2[6:3]) — This field specifies the width of the HSYNC pulse in $\overline{\text{CCLK}}$ periods.

FIGURE 16 — INTERLACED DISPLAY MODES



HORIZONTAL BACK PORCH (IR2[2:0]) — This field defines the number of $\overline{\text{CCLK}}$s between the trailing edge of HSYNC and the trailing edge of BLANK.

VERTICAL FRONT PORCH (IR3[7:3]) — Specifies the number of scan line periods between the rising edges of BLANK and VSYNC during the vertical retrace interval. The vertical front porch will be extended in increments of scan lines if the ACLL input is low at the end of the programmed value.

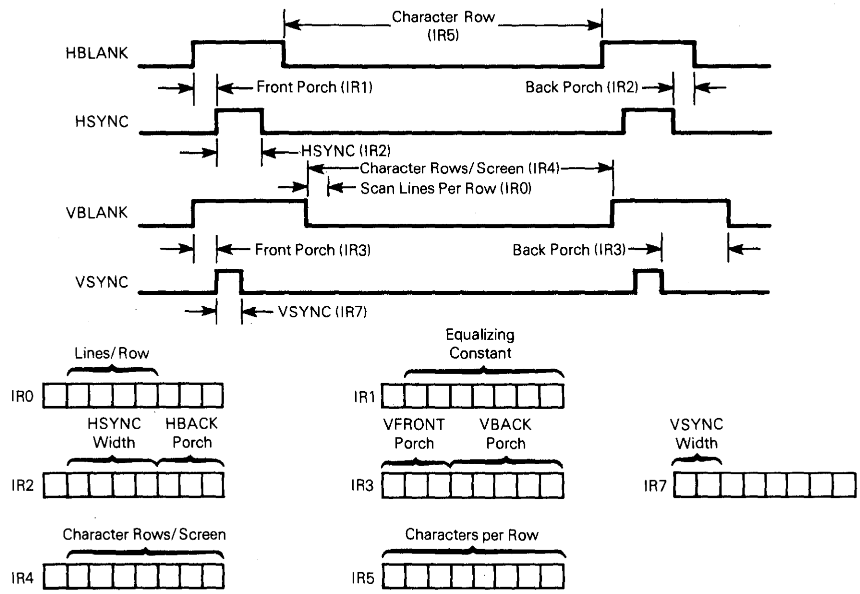
CHARACTER BLINK RATE (IR4[7]) — Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/64 or 1/128 of the vertical field rate. The timing signal has a duty cycle of 50% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

CHARACTER ROWS PER SCREEN (IR4[6:0]) — This field defines the number of character rows to be displayed. The value multiplied by the scan lines per character row, plus the vertical front porch, the vertical back porch values, and the vertical sync pulse width is the vertical scan period in scan lines.

ACTIVE CHARACTERS PER ROW (IR5[7:0]) — This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in CCLKs.

FIRST AND LAST SCAN LINE OF CURSOR (IR6[7:4], IR6[3:0]) — These two field specify the height and position of the cursor on the character block. The "first" line is the topmost line when scanning from the top to the bottom of the screen.

FIGURE 17 — HORIZONTAL AND VERTICAL TIMING



VERTICAL SYNC PULSE WIDTH (IR7[7:6]) — This field specifies the width of the VSYNC pulse in scan line periods.

CURSOR BLINK ENABLE (IR7[5]) — This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR7[4]). The blink duty cycle for the cursor is 50%.

CURSOR BLINK RATE (IR7[4]) — The cursor blink rate can be specified at 1/32 or 1/64 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

UNDERLINE POSITION (IR7[3:0]) — This field defines which scan line of the character row will be used for the underline attribute by the MC2675 CMAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

DISPLAY BUFFER FIRST ADDRESS (IR9[3:0]), IR8[7:0] AND DISPLAY BUFFER LAST ADDRESS (IR9[7:4]) — These two fields define the area within the buffer memory where the display data will reside. When the data at the "display buffer last address" is displayed, the AVDC will wraparound and obtain the data to be displayed at the next screen position from the "display buffer first address". If "last address" is the end of a character row and a new screen start address has been loaded into the screen start register, or if "last address" is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the automatic split screen or split screen interrupt features of the AVDC.

DISPLAY POINTER ADDRESS LOWER (IR10[7:0] AND DISPLAY POINTER ADDRESS UPPER (IR11[5:0]) — These two fields define a buffer memory address for AVDC controlled accesses in response to "read/write at pointer" commands. They also define the last buffer memory address to be written for the "write from cursor to pointer" command.

SCAN LINE ZERO DURING SCROLL DOWN (IRZ11[7]) — This field specifies normal scan line count or all scan line zero counts for the new character row that occurs at the top of the scrolling area during soft scroll down operation. If the character generator provides blanks during scan line zero, this will cause the new row to be automatically blanked on the display. This feature can be used, if necessary, to blank the new row until the CPU places "blank data" into the display buffer.

SCAN LINE ZERO DURING SCROLL UP (IR11[6]) — This field specifies normal scan line count or all scan line counts for the new character row that occurs at the bottom of the scrolling area during soft scroll up operation.

SCROLL START (IR12[7]) — This bit is asserted when soft scroll is to take place. The scrolling area begins at the row specified in split register 1 (IR12[6:0]). If set, the first row to scroll scan line count will be reduced by the value in the lines to scroll register (IR14[3:0]). The scan line count of this row will start at the programmed offset value. When this bit is asserted, scroll end IR13[7] must be set before split register 2.

SPLIT REGISTER 1 (IR12[6:0]) — Split register 1 can be used to provide special screen effects such as soft (scan line by scan line) scrolling, double height/width rows, or to change the normal addressing sequence of the display memory. The contents of this field is compared, in real time, to the current row number. Upon a match, the AVDC sets the split screen 1 status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of the scan line zero of the split screen character row. If enabled by the SPL1 bit of screen start register 2, an automatic split screen to the address specified in screen start register 2 will be made for the designated character row. During a scroll operation, this field defines the first character row of the scrolling area.

SCROLL END (IR13[7]) — This field specifies that the row programmed in split register 2 (IR13[6:0]) is to be the last scrolling row of the scrolling area. Note that this bit must be asserted for a valid row only when the scroll start bit IR12[7] is also asserted.

SPLIT REGISTER 2 (IR13[6:0]) — This field is similar to the split register 1 field except for the following:

1. Split screen 2 status bit is set.
2. During a scroll operation, this field defines the last character row of the scrolling area. This row will be followed by a partial row. The LTSR (IR14) value replaces the normal scan lines/row value for the partial row, thus keeping the total scan lines/screen the same.
3. If enabled by the SPL2 bit of screen start register 2, an automatic split to the address contained in screen start register 2 will occur in one of two ways:
   a) If not scrolling an automatic split will occur for the next character row.
   b) If scrolling, the automatic split will occur after the partial row being scrolled onto or off the screen.
4. The specified double width and height conditions (IR14) are also asserted in two possible ways:
   a) Automatic split will assert the programmed condition for the current row.
   b) During soft scroll operation the programmed conditions are asserted for the partial row scrolling onto or off the screen.

DOUBLE 1 (IR14[7:6]) — This field specifies the conditions (double width/height or normal) of the row designated in split register 1 (IR12[6:0]). When double height tops or bottoms has been specified, the AVDC will automatically toggle between tops and bottoms until another split 1 or 2 occurs which changes the double height/width condition. If a double height top row is specified, the scan line count will start at zero and increment the scan line every other scan line. If a double height bottom row is specified, the AVDC will start a one half the normal scan line total. If double width is specified, the AVDC will assert the DADD9/DW output at the falling edge of blank. This condition will also remain active until the next split 1 or 2. When IR0[7] = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14[7:6] and the automatic toggling between tops and bottoms is disabled.

DOUBLE 2 (IR14[5:4]) — This field specifies the conditions (double width/height or normal) of the row designated in split register 2 (IR13[6:0]). Not used with IR0[7] = 1.

LINES TO SCROLL (IR14[3:0]) — This field defines the scan line increment to be used during a soft scroll operation. This value will only be used when scroll start (IR12[7]) and scroll end (IR13[7]) are enabled.

TIMING CONSIDERATIONS

Normally, the contents of the initialization registers are not changed during normal operation. However, this may be necessary to implement special display features such as multiple cursors and horizontal scrolling. Table 2 describes timing details for these registers which should be considered when implementing these features.

3

TABLE 2 — TIMING CONSIDERATIONS

Parameter	Timing Considerations
First Line of Cursor Last Line of Cursor Underline Line	These parameters must be established at a minimum of two character times prior to their occurrence.
Double Height Character Rows Double Width Character Rows Rows to Scroll	Set/reset prior to the row specified in split 1 or 2 registers.
Cursor Blink Cursor Blink Rate Character Blink Rate	New values become effective within one field after values are changed.
Split Register 1 Split Register 2	Change anytime prior to line zero of desired row.
Character Rows Per Screen	Change only during vertical blanking period.
Vertical Front Porch	Change prior to first line of VFP.
Vertical Back Porch	Change prior to four line after VSYNC.
Screen Start Register 1 Row Table Mode Enable	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used.

DISPLAY CONTROL REGISTERS

There are seven registers in this group, each with an individual address. Their formats are illustrated in Figure 18. The command register is used to invoke one of 19 possible AVDC commands as described in COMMANDS. The remaining registers in the group store address values which specify the cursor location, the location of the first character to be be displayed on the screen, and any split screen address locations. The user initializes these registers after powering on the system and changes their values to control the data which is displayed.

SCREEN START REGISTERS 1 AND 2

The screen start 1 registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then advanced sequentially at the character clock rate for the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC is loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the entire process repeats again.

During vertical blanking, the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, refreshing continues from the display buffer first address.

The sequential operation described above will be modified upon the occurrence of any of three events. First, if during the incrementing of the memory address counter the "display buffer last address" (IR9[7:4]) is reached, the MAC will be loaded from the "display buffer first address" register (IR9[3:0] and IR8[7:0]) at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see Figure 19a).

The sequential row to row addressing can also be modified via split register 1 (IR12) and split register 2 (IR13), under CPU control, or by enabling the row table addressing mode. If bit 6 of screen start register 2 upper (SPL1) is set, the screen start register 2 contents will be loaded automatically into the RSR at the beginning of the first scan line of the row designated by split register 1 (IR12[6:0]). If bit 7 of screen start 2 upper (SPL2) is set, the screen start register 2 contents is automatically loaded into the RSR at the end of the last scan line of the row designated by split register 2 (IR13[6:0]). SPL1 and SPL2 are write only bits and will read as zero when reading screen start register 2.

If the contents of screen start register 1 (upper, lower, or both) are changed during any character row (e.g., row 'n'), the starting address of the next character row (row 'n+1') will be the new value of the screen start register and addressing will continue sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the AVDC is useful in controlling the CPU initiated operations. Note that in order to obtain the correct screen display, screen start register 1 must be reloaded with the original (origin of display) value prior to the end of the vertical retrace. See Figure 19b.

FIGURE 18 — DISPLAY CONTROL REGISTER FORMATS (Sheet 1 of 2)

7	6	5	4	3	2	1	0
Command Code							
See **COMMANDS** for Command Codes							

Command Registers (Write Only)

7	6	5	4	3	2	1	0
		Upper Register					
DADD15	DADD14	Most Significant Bits					

FIGURE 18 — DISPLAY CONTROL REGISTER FORMATS (Sheet 2 of 2)

7	6	5	4	3	2	1	0
Lower Register (Least Significant Bit)							
H'0000' = 0 H'0001' = 1 Through H'3FFE' = 16,382 H'3FFF' = 16,383		NOTE: Most significant bits are in upper register [5:0]					

NOTES:
1. Bits 7 and 6 of upper register are not used in the cursor address register.
2. Bits 7 and 6 of upper register are always zero when read by the CPU.
3. When IR0[7] = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14[7:6] to control the double width and double height attributes of the display as follows:

7	6	Attribute
0	0	None
0	1	Double Width Only
1	0	Double Width and Double Height Tops
1	1	Double Width and Double Height Bottoms

Screen Start 1 Register (Read and Write) and Cursor Address Registers (Read and Write)

7	6	5	4	3	2	1	0
		Upper Register					
SPL2 0 = Off 1 = On	SPL1 0 = Off 1 = On	Most Significant Bits					

7	6	5	4	3	2	1	0
Lower Register (Least Significant Bit)							
H'0000' = 0 H'0001' = 1 Through H'3FFE' = 16,382 H'3FFF' = 16,383		NOTE: Most significant bits are in upper register [5:0]					

NOTE:
Bit 7 and bit 6 are always zero when read by the CPU.

Screen Start 2 Registers (Read and Write)

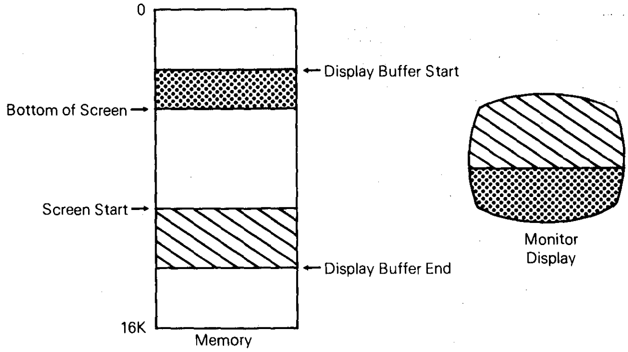
When row table addressing mode is enabled, the first address of the row table is designated in SSR2. The AVDC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row and loads it into SSR1 for use as the starting address of the next row. Since the contents of SSR2 changes as the table entries are fetched, it must be re-initialized to point to the first table entry during each vertical retrace interval.

The values of the two most significant bits of SSR1 upper are multiplexed onto the DADD1/DADD14 and DADD2/DADD15 outputs during the falling edge of BLANK. If IR0[7] = 0, these two bits act as memory page select bits which may be used to extend the display memory addressing range of the AVDC up to 64K. In that case, these two bits act as a two-bit counter which is incremented each time that "wraparound" occurs (see above). Note that the counter is incremented at the falling edge of BLANK and that for proper display operation the wraparound address should be programmed to occur at the last character position of a row. Also, the first address accessed in the new page will be the address contained in the display buffer first address register (IR9[3:0] and IR8[7:0]).

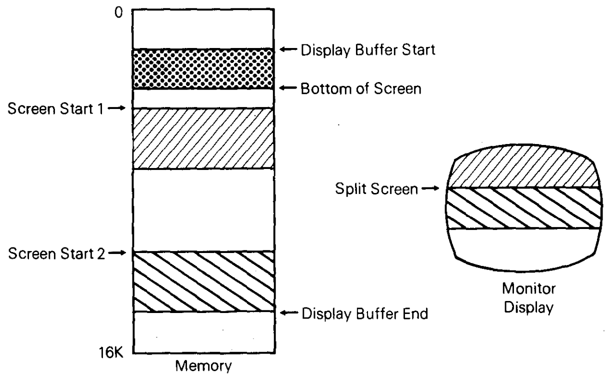
CURSOR ADDRESS REGISTERS

The contents of these registers define the buffer memory address of the cursor. The cursor output will be asserted when the memory address counter matches the value of the

FIGURE 19 — DISPLAY ADDRESSING OPERATION



(a) Display Memory Wraparound



(b) Display Memory Split Screen With Wraparound

cursor address registers for the scan lines specified in IR6. The cursor address registers can be read or written by the CPU or incremented via the "increment cursor address" command. In independent buffer mode, these registers define a buffer memory address for AVDC controlled access in response to "read/write at cursor with/without increment" commands, or the first address to be used in executing the "write from cursor to pointer" command.

INTERRUPT/STATUS REGISTERS

The interrupt and status registers provide information to the CPU to allow it to interact with the AVDC to effect desired changes that implement various display operations. The interrupt register provides information on five display operations. The interrupt register provides information on five possible interrupt conditions, as shown in Figure 20. These conditions can be selectively enabled or disabled (masked) from causing interrupts by certain AVDC commands. An interrupt condition which is enabled (masked bit equal to one) will cause the $\overline{\text{INTR}}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set upon the occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{\text{INTR}}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupt conditions plus the RDFLG bit. For this register, however, the contents are not affected by the state of the mask bits.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a "reset interrupt/status bits" command. The bits are also reset by a "master reset" command and upon power-up.

FIGURE 20 — INTERRUPT AND STATUS REGISTER FORMAT

7	6	5	4	3	2	1	0
		RDFLG	VBLANK	Line Zero	Split 1	Ready	Split 2
Not Used Always Read as 0		0 = Busy 1 = Ready	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Busy 1 = Ready	0 = No 1 = Yes

RDFLG (I/SR[5]) — This bit is present in the status register only. A zero indicates that the AVDC is currently executing the previously issued delayed command. A one indicates that the AVDC is ready to accept a new delayed command.

VBLANK (I/SR[4]) — Indicates the beginning of a vertical blanking interval. Set to one at the beginning of the first scan line of the vertical front porch.

LINE ZERO (I/SR[3]) — Set to one at the beginning of the first scan line (line 0) of each active character row.

SPLIT SCREEN 1 (I/SR[2]) — This bit is set when a match occurs between the current character row number and the value contained in split register 1, IR12[6:0]. The equality condition is only checked at the beginning of line zero of each character row.

READY (I/SR[1]) — The delayed commands affect the display and may require the AVDC to wait for a blanking interval before enacting the command. This bit is set to one when execution of a delayed command has been completed. No other delayed command should be invoked until the prior delayed command is completed.

SPLIT SCREEN 2 (I/SR[0]) — This bit is set when a match occurs between the current character row number and the value contained in split register 2 (IR13[6:0]).

COMMANDS

The AVDC commands are divided into two classes: the instantaneous commands which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in Table 3. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

TABLE 3 — AVDC COMMAND FORMATS

D7	D6	D5	D4	D3	D2	D1	D0	Hex	Command
									Instantaneous Commands
0	0	0	0	0	0	0	0		Master Reset
0	0	0	1	V	V	V	V		Load IR Pointer with Value V (V=0 to 14)
0	0	1	d	d	d	1	0*		Disable Graphics
0	0	1	d	d	d		1*		Enable Graphics
0	0	1	d	1	N		0*		Disable Off — Float DADD Bus if N=1
0	0	1	d	1	N		1*		Disable On — Next Field (N=1) or Scan Line (N=0)
0	0	1	1	d	d		0*		Cursor Off
0	0	1	1	d	d		1*		Cursor On
0	1	0	N	N	N		N		Reset Interrupt/Status: Bit Reset where N=1
1	0	0	N	N	N		N		Disable Interrupt: Disable where N=1
0	1	1	N	N	N		N		Enable Interrupt: Enables Interrupts where N=1
			V B	L Z	S P 1	R D Y	S P 2		Interrupt Bit Assignments
									Delayed Commands
1	0	1	0	0	1	0	0	A4	Read at Pointer Address
1	0	1	0	0	0	1	0	A2	Write at Pointer Address
1	0	1	0	1	0	0	1	A9	Increment Cursor Address
1	0	1	0	1	1	0	0	AC	Read at Cursor Address
1	0	1	0	1	0	1	0	AA	Write at Cursor Address
1	0	1	0	1	1	0	1	AD	Read at Cursor Address and Increment Address
1	0	1	0	1	0	1	1	AB	Write at Cursor Address and Increment Address
1	0	1	1	1	0	1	1	BB	Write from Cursor Address to Pointer Address
1	0	1	1	1	1	0	1	BD	Read from Cursor Address to Pointer Address

NOTES:
* Any combination of these three commands is valid.
d = Don't care.

INSTANTANEOUS COMMANDS

The instantaneous commands are executed immediately after the trailing edge of the write pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits and can be invoked at any time.

MASTER RESET

This command initializes the AVDC and can be invoked at any time to return the AVDC to its initial state. Upon power-up, two successive master reset commands must be applied to release the AVDC's internal power-on circuits. In transparent and shared buffer modes, the CTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of the command and BLANK goes high. After command completion, HSYNC and VSYNC will begin operation and BLANK will remain high until a "display on" command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The row buffer mode, cursor-off, display-off, and line graphics disable states are set.
4. The initialization register pointer is set to address IR0.
5. IR2[7] is reset.

LOAD IR ADDRESS

This command is used to preset the initialization register pointer with the value "V" defined by D3-D0. Allowable values are 0 to 14.

ENABLE GRAPHICS

After invoking this command, the AVDC will increment the MAC to the next consecutive memory address for each scan line even if more than one scan line per row is programmed. This mode can be used for bit-mapped graphics where each location in the display buffer within the defined area contains the bit pattern to be displayed. This command is row buffered and should be asserted during the character row prior to the row where this feature is required. This allows the user to enter and exit graphics mode on character row boundaries.

To perform split screen operations while in graphics mode use SSR2 only.

DADD0/LG is asserted during the trailing edge of BLANK for each scan line while this mode is active.

DISABLE GRAPHICS

Normal addressing resumes at the next row boundary.

DISPLAY OFF

Asserts the BLANK output. The DADD0 through DADD13 display address bus outputs can be optionally placed in the three-state condition by setting bit 2 to a one when invoking the command.

DISPLAY ON

Restores normal blanking operation either at the beginning of the next field (bit 2=1) or at the beginning of the next scan line (bit 2=0). Also returns the DADD0-DADD13 drivers to their active state.

CURSOR OFF

Disables cursor operation. Cursor output is placed in the low state.

CURSOR ON

Enables normal cursor operation.

RESET INTERRUPT/STATUS BITS

This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 – Split 2
Bit 1 – Ready
Bit 2 – Split 1
Bit 3 – Line Zero
Bit 4 – Vertical Blank

DISABLE INTERRUPTS

Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from being set in the interrupt register and asserting the INTR output. Bit position correspondence is as above.

ENABLE INTERRUPTS

This command writes the associated interrupt mask bit to a one. This enables the corresponding conditions to be set in the interrupt register and asserts the INTR output. Bit position correspondence is as above.

DELAYED COMMANDS

This group of commands is utilized for the independent buffer mode of operation, although the "increment cursor" command can also be used in other modes. With the exception of the "write from cursor to pointer" and "increment cursor" commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a "display off" state, the command is executed immediately.

The "increment cursor" command is executed immediately after it is issued and requires approximately three $\overline{CCLK}$ periods for completion. The "write from cursor to pointer" command executes during blanking intervals. The AVDC will execute as many writes as possible during each blanking interval. If the command is not completed during the current blanking interval, the command will be held in suspension during the next active portion of the screen and continues during the next blanking interval until the command is completed.

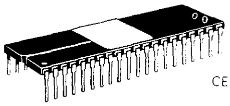
MOTOROLA

MC2675

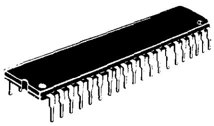
Advance Information

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

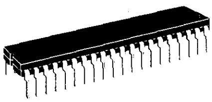
COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC)



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC)

The MC2675 color/monochrome attributes controller (CMAC) is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a programmable dot clock divider to generate a character clock, a high speed shift register to serialize input dot data into a video stream, latches, logic to apply visual attributes to the resulting display, and logic to display a cursor on the display.

The CMAC provides control of visual attributes on a character-by-character basis for two operating modes: monochrome and color. The monochrome mode provides reverse video, blank, highlight, and two general purpose user definable attributes. In this mode, the display characters can be specified to appear on either a light or dark screen background. Retrace video supression can be automatically or externally controlled. The color mode provides eight colors for foreground (character) video and eight colors for background video together with a luminance output for external color set selection or to simultaneously drive a monochrome monitor. Additionally, both modes provide double width, underline, blink, dot stretching, and dot width attributes. In monochrome mode, the MC2675 emulates the attributes characteristics of Digital Equipment Corporation VT100 terminal.

The horizontal dot frequency is the basic timing input to the CMAC. This clock is divided internally to provide a character clock output for system synchronization. Up to ten bits of dot data are parallel loaded into the video shift register on each character boundary. The two TTL video data outputs in monochrome mode are encoded to provide four video intensities (black, gray, white, and highlight). The video data in color mode is encoded to provide eight foreground colors and shifted out on three TTL outputs, together with the luminance output.

Applications include CRT terminals, word processing systems, small business computers.

- 25 MHz and 18 MHz Video Dot Rate Versions*
- Four Video Intensities Encoded on Two TTL Outputs (Monochrome Mode)
- Eight Foreground and Background Colors Encoded on Three TTL Outputs (Color Mode)
- Internally Latched Character Atrributes:

Reverse Video	Two General Purpose
Blank	Eight Foreground Colors
Blink	Eight Background Colors
Underline	Dot Width Control
Highlight	Double Width Characters

- VT100 Compatible Attributes
- Reverse Video Cursor with Optional White Cursor in Color Mode
- Up to Ten Dots Per Character
- Light or Dark Background in Monochrome Mode — Automatic Retrace Blanking
- Programmable Dot Stretching
- TTL Compatible
- 40-Pin Dual-in-Line Package

*For faster versions consult factory.

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{BB}	1	40	V_{CC}
D1	2	39	D0
D3	3	38	D2
D5	4	37	D4
D7	5	36	D6
D8	6	35	C1
RBLANK	7	34	C0
CURSOR	8	33	$\overline{\text{CCLK}}$
CMODE	9	32	DCLK
DOTS	10	31	DOTM
BLINK	11	30	M/C
BLANK	12	29	BLUE/TTLV2
UL	13	28	RED/TTLV1
ADOUBLE	14	27	GREEN/GP1
RESET	15	26	LUM/GP2
ABLINK	16	25	AREDF/AHILT
AGREENF/BKGND	17	24	ADOTM
AGREENB/ARVID	18	23	ABLUEF/ABLANK
AUL	19	22	ABLUEB/AGP2
GND	20	21	AREDB/AGP1

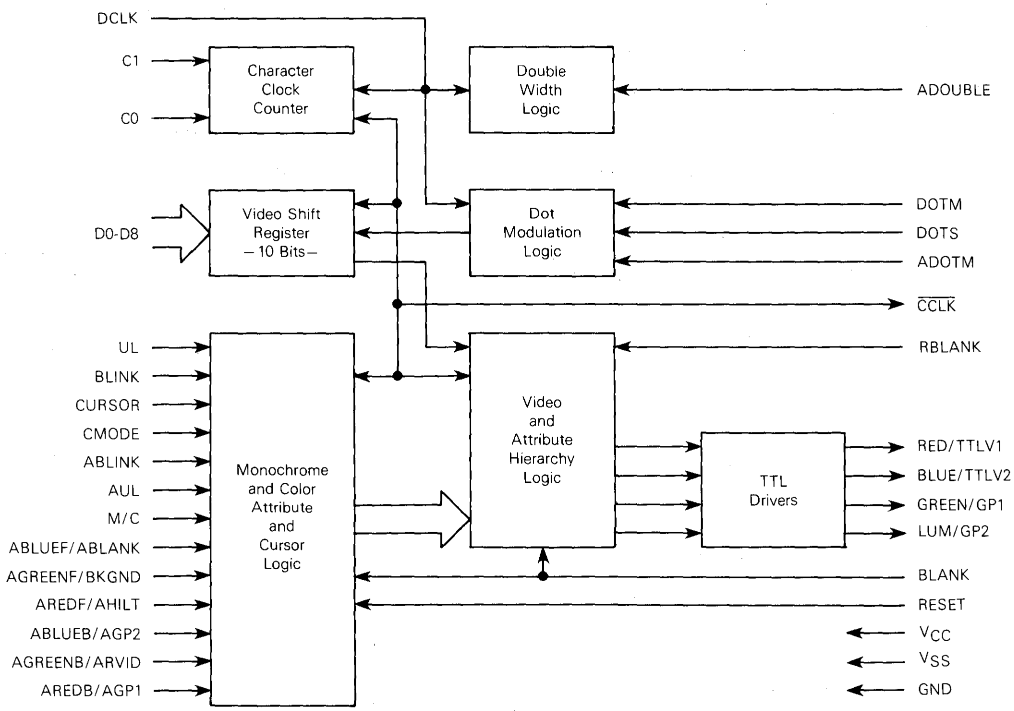
This document contains information on a new product. Specifications and information herein are subject to change without notice.

ORDERING INFORMATION (V_{CC} = 5 V ±5%, 0°C to 70°C)

Package Type	Dots Per Character	Frequency (MHz)	Order Number
Ceramic L Suffix	7, 8, 9, 10	18	MC2675B8L
	7, 8, 9, 10	25	MC2675B5L
	6, 8, 9, 10	18	MC2675C8L
	6, 8, 9, 10	25	MC2675C5L
Plastic P Suffix	7, 8, 9, 10	18	MC2675B8P
	7, 8, 9, 10	25	MC2675B5P
	6, 8, 9, 10	18	MC2675C8P
	6, 8, 9, 10	25	MC2675C5P
Cerdip S Suffix	7, 8, 9, 10	18	MC2675B8S
	7, 8, 9, 10	25	MC2675B5S
	6, 8, 9, 10	18	MC2675C8S
	6, 8, 9, 10	25	MC2675C5S

3

BLOCK DIAGRAM



ABSOLUTE 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	Rating
Thermal Resistance			
Plastic Package	θ_{JA}	50	°C/W
Ceramic Package		50	

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

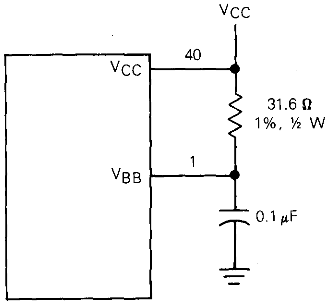
$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

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

DC ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0 V ±5%)

Parameter		Symbol	Min	Max	Unit
Input Low Voltage		V_{IL}	−0.3	0.8	V
Input High Voltage		V_{IH}	2.0	V_{CC}	V
Output Low Voltage (I_{OL} = 4 mA)		V_{OL}	–	0.4	V
Output High Voltage (I_{OH} = −400 μA)		V_{OH}	2.4	–	V
Input Low Current (V_{in} = 0.4 V)	DCLK	I_{IL}	–	−800	μA
	All Other Inputs		–	−400	
Input High Current (V_{in} = 2.4 V)	DCLK	I_{in}	–	40	μA
	All Other Inputs		–	20	
V_{CC} Supply Current (V_{in} = 0 V, V_{CC} = Max)		I_{CC}	–	80	mA
V_{BB} Supply Current (See Figure 1)		I_{BB}	–	120	mA

FIGURE 1 — RECOMMENDED V_{BB} TEST CIRCUIT



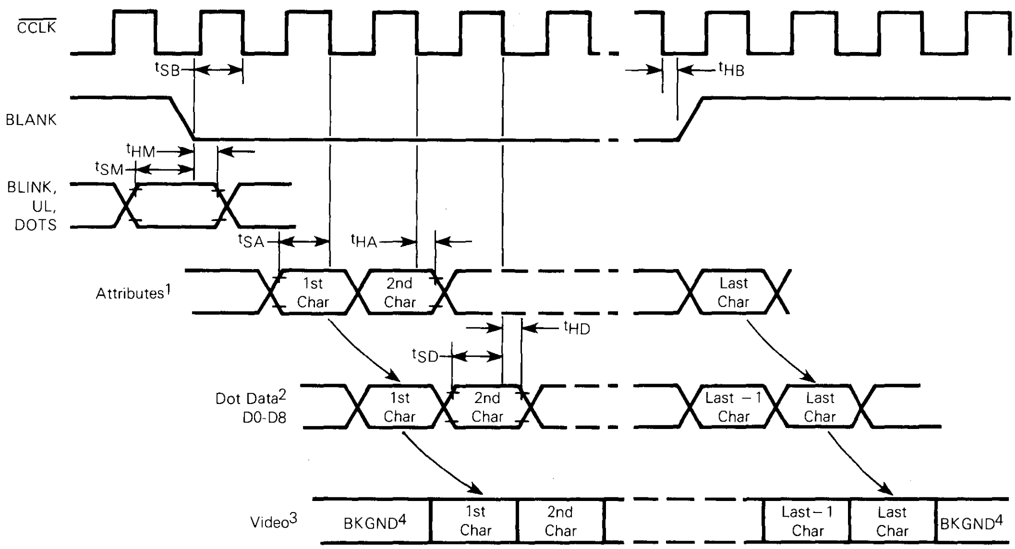
3

AC ELECTRICAL CHARACTERISTICS — DOT CLOCK TIMING (T_A = 0°C to 70°C, V_{CC} = 5.0 V ±5%)

Parameter	Symbol	25 MHz Min	25 MHz Max	18 MHz Min	18 MHz Max	Unit
Dot Clock High Time	t_{DH}	15	–	22	–	ns
Dot Clock Low Time	t_{DL}	15	–	22	–	ns
BLANK to $\overline{CCLK}$ Setup Time	t_{SB}	40	–	50	–	ns
Attributes to $\overline{CCLK}$ Setup Time	t_{SA}	40	–	50	–	ns
D0-D9 to $\overline{CCLK}$ Setup Time	t_{SD}	60	–	70	–	ns
CURSOR to $\overline{CCLK}$ Setup Time	t_{SK}	40	–	50	–	ns
C0 and C1 to DCLK Setup Time	t_{SC}	20	–	20	–	ns
RBLANK to DCLK Setup Time	t_{SR}	20	–	20	–	ns
BLINK, UL, DOTS, to BLANK Setup Time	t_{SM}	20	–	20	–	ns
BLANK from $\overline{CCLK}$ Hold Time	t_{HB}	20	–	20	–	ns
Attributes from $\overline{CCLK}$ Hold Time	t_{HA}	20	–	20	–	ns
D0-D8 from $\overline{CCLK}$ Hold Time	t_{HD}	30	–	30	–	ns
CURSOR from $\overline{CCLK}$ Hold Time	t_{HK}	20	–	20	–	ns
C0 and C1 from DCLK Hold Time	t_{HC}	20	–	20	–	ns
RBLANK from DCLK Hold Time	t_{HR}	20	–	20	–	ns
BLINK, UL, DOTS, from BLANK Hold Time	t_{HM}	20	–	20	–	ns
$\overline{CCLK}$ from DCLK Delay Time (C_L = 50 pF)	t_{DC}	–	55	–	70	ns
Other Outputs from DCLK Delay Time (C_L = 50 pF)	t_{DV}	30	60	35	70	ns

NOTE: All voltage measurements are referenced to ground. For testing, all input signals swing between 0.4 volts and 2.4 volts with a transition time of 3 nanoseconds maximum. All time measurements are referenced at input voltages of 0.8 volts and 2.0 volts and at output voltages of 0.8 volts and 2.0 volts as appropriate.

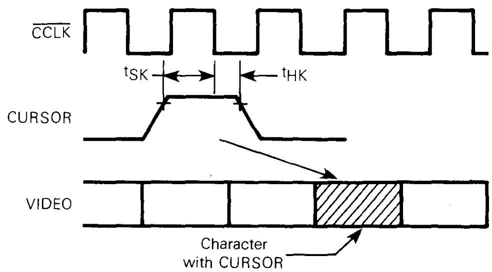
CMAC PIPELINE TIMING DIAGRAM



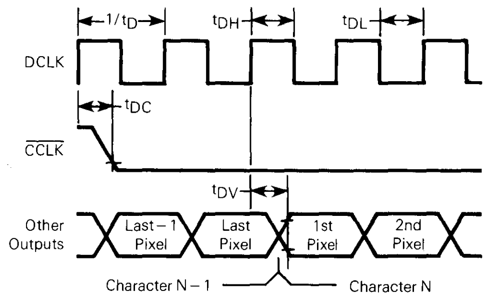
NOTES:

1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, ADOUBLE, ADOTM, two general purpose, and foreground/background colors.
2. One $\overline{CCLK}$ delay for dot data (obtained from delay through character generator).
3. For detail timing of video outputs, see Output Pipeline Timing Diagram.
4. Non-active scan time. Video reverts to polarity selected by the BKGND input in monochrome mode.

CURSOR PIPELINE TIMING DIAGRAM

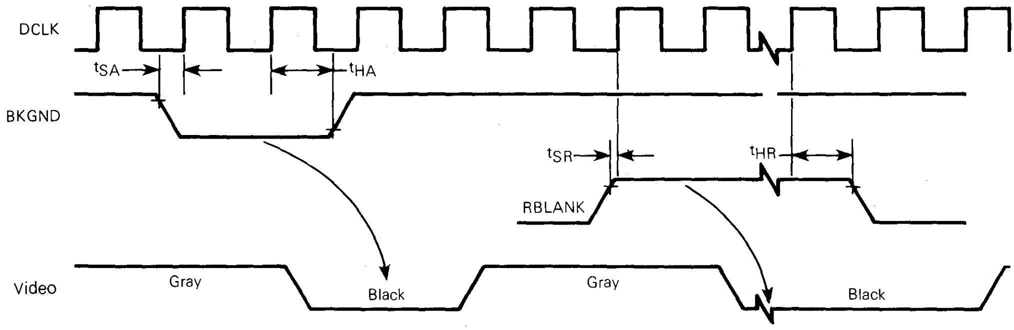


OUTPUT PIPELINE TIMING DIAGRAM

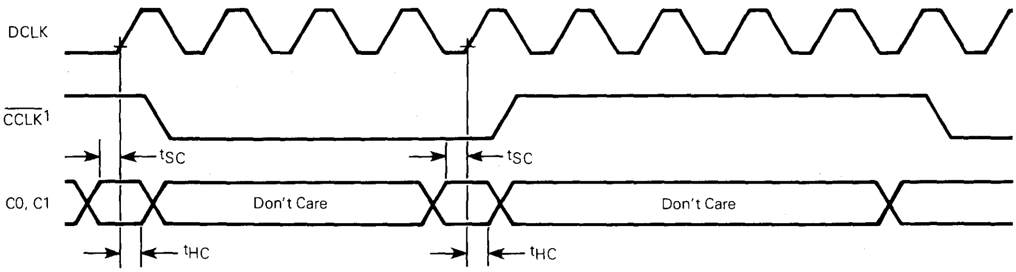


3

BKGND AND RBLANK TIMING DIAGRAM DURING INACTIVE SCAN TIME (BLANK=1) — MONOCHROME MODE



CLOCK DIVIDER TIMING DIAGRAM



NOTE:
1. The high and low times of $\overline{CCLK}$ may be controlled independently.

SIGNAL DESCRIPTION

The input and output signals for the CMAC are described in the following paragraphs.

V_{CC}, V_{BB}, AND GND

Power is supplied to the CMAC using these three pins. V_{CC} is the +5 volts ±5% power input, V_{BB} is the bias supply current (refer to Figure 1), and GND is the ground connection.

DOT CLOCK (DCLK)

This dot frequency input controls the video output shift rate.

CHARACTER CLOCK ($\overline{CCLK}$)

This output is a submultiple of DCLK. The period ranges from seven to ten DCLK periods per cycle and is determined by the state of the character clock control (C0-C1) inputs.

RED/TTL VIDEO 1 (RED/TTLV1)

In color mode, this output provides the red gun serial video. In monochrome mode, it should be used with the blue/TTL video 2 output to decode four video intensities.

BLUE/TTL VIDEO 2 (BLUE/TTLV2)

In color mode, this output provides the blue gun serial video. In monochrome mode, it should be used with the read/TTL video 1 output to decode four video intensities.

GENERAL/GENERAL PURPOSE 1 (GREEN/GP1)

In color mode, this output provides the green gun serial video. In monochrome mode, it is a general purpose TTL output which is asserted if the AREDB/AGP1 input is asserted when the corresponding character dot data is loaded into the video shift register.

LUMINANCE/GENERAL PURPOSE 2 (LUM/GP2)

In color mode, this output is the logical OR of the RGB foreground video. It is low during a blanking interval and during the foreground portion of the cursor display. In monochrome mode, it is a general purpose TTL output which is asserted if the ABLUEB/AGP2 input is asserted when the corresponding character dot data is loaded into the video shift register.

UNDERLINE TIMING (UL)

Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.

BLINK TIMING (BLINK)

This input is sampled on the falling edge of BLANK to provide the blink rate for the blink attribute. Should be a submultiple of the frame rate.

SCREEN BLANK (BLANK)

When high, this input forces the video outputs to the specified background color in color mode and to the level specified by the BKGND input (either black or gray) in monochrome mode.

RETRACE BLANK (RBLANK)

This input is used to force the video outputs to a low during retrace periods. If pulled high, it will automatically suppress video during the retrace periods when BLANK is high. The user may also pulse this input while BLANK is high to selectively suppress raster video.

GREEN FOREGROUND/BACKGROUND INTENSITY (AGREENF/BKGND)

In color mode, this input activates the GREEN/GP1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input specifies gray or black screen background.

BLUE FOREGROUND/BLANK ATTRIBUTE (ABLUEF/ABLANK)

In color mode, this input activates the BLUE/TTLV2 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input generates a blank space for the associated character. The blank space intensity is controlled by the AGREENF/BKGND input, the reverse video attribute and cursor input.

RED FOREGROUND/HIGHLIGHT ATTRIBUTE (AREDF/AHILT)

In color mode, this input activates the RED/TTLV1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input highlights the associated character (including underline).

CURSOR TIMING (CURSOR)

This input provides the timing for the cursor video. In color mode, with CURSOR and CMODE high, the RGB outputs are driven high (white cursor). If CMODE is low, or in monochrome mode, this input reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.

CURSOR MODE (CMODE)

Used in color mode only. When CURSOR and CMODE are high, the RGB outputs are driven high (white cursor). When CURSOR is high and CMODE is low, the RGB outputs are logically inverted (reverse video cursor).

UNDERLINE ATTRIBUTE (AUL)

Specifies a line to be displayed in the character block. The specific line(s) are specified by the UL input. All other attributes apply to the underline video.

BLINK ATTRIBUTE (ABLINK)

In color mode, this active high input will drive the foreground RGB combination to the background RGB combination. In monochrome mode, the associated character or background is driven to the intensity determined by BKGND, reverse video attribute, and the cursor input.

DOUBLE WIDTH ATTRIBUTE (ADOUBLE)

This active high input causes the associated character video to be shifted out of the serial shift register at one-half the dot frequency (DCLK). The $\overline{\text{CCLK}}$ output is not affected.

RED BACKGROUND/GENERAL PURPOSE ATTRIBUTE 1 (AREDB/AGP1)

In color mode, this input activates the RED/TTLV1 output during the background portion of the associated character block. In monochrome mode, it activates the GREEN/GP1 output for the associated character block.

BLUE BACKGROUND/GENERAL PURPOSE ATTRIBUTE 2 (ABLUEB/AGP2)

In color mode, this input activates the BLUE/TTLV2 output during the background portion of the associated character block. In monochrome mode, it activates the LUM/GP2 output for the associated character block.

GREEN BACKGROUND/REVERSE VIDEO ATTRIBUTE (AGREENB/ARVID)

In color mode, this input activates the GREEN/GP1 output during the background portion of the associated character block. In monochrome mode, it causes the associated character block video intensities to be reversed.

DOT DATA INPUT (D0-D8)

These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the trailing (falling) edge of each character clock ($\overline{\text{CCLK}}$).

CHARACTER CLOCK CONTROL (C0-C1)

The states of these two static inputs determine the internal divide factor for the $\overline{\text{CCLK}}$ output rate.

RESET (RESET)

This active high input initializes the internal logic and resets the attribute latches.

MONOCHROME/COLOR MODE (M/C)

This input selects whether the CMAC operates in monochrome or color mode. A low selects color mode and a high selects monochrome mode.

DOT MODULATION ATTRIBUTE (ADOTM)

When DOTM and this input are high, the active dot width of the associated character video is one DCLK. When DOTM is high and this input is low, the active dot width of the associated character video is two DCLKs.

DOT WIDTH MODULATION (DOTM)

When this input is high, two DCLKs are used for each dot shifted through the shift register. When this input is low, one DCLK is used.

DOT STRETCHING (DOTS)

This input is sampled at the falling edge of BLANK. When this input is high, one extra dot is appended to individual dots or groups of dots of the input parallel data and then transferred through the shift register. When this input is low, normal transfer of input parallel data results.

FUNCTIONAL DESCRIPTION

The CMAC consists of seven major sections (refer to the block diagram). The high speed dot clock input is applied to a programmable divider to provide a character clock output for system timing. Parallel dot data is loaded into the video shift register on character boundaries and shifted into the video logic block at the dot rate specified by the dot modulation section. The appropriate attribute control inputs are selected by the mode select logic, latched internally on character boundaries, and combined with the serial dot data to provide monochrome or color video outputs. System block diagrams of the MC2675 in color mode and monochrome mode are provided in Figures 2 and 3.

The BLANK input defines the active screen area. In color mode, the video outputs are forced to the specified background color when this signal is asserted; in monochrome mode the video outputs are forced to the states defined by the BKGND input, i.e., black if dark background is selected and gray if light background is selected. A separate RBLANK input allows the user to select the amount of border around the active area when operating in color mode or in monochrome mode with light background. This input can be

FIGURE 2 — SYSTEM BLOCK DIAGRAM OF MC2675 IN COLOR MODE

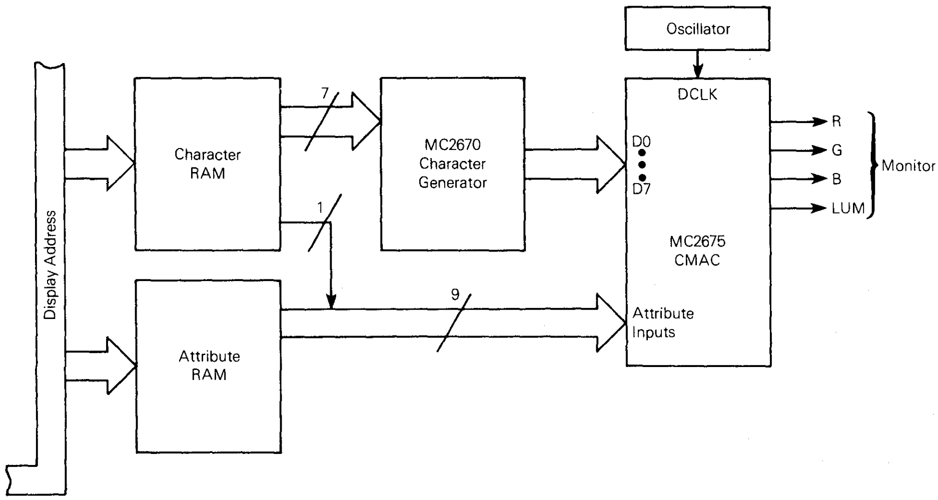
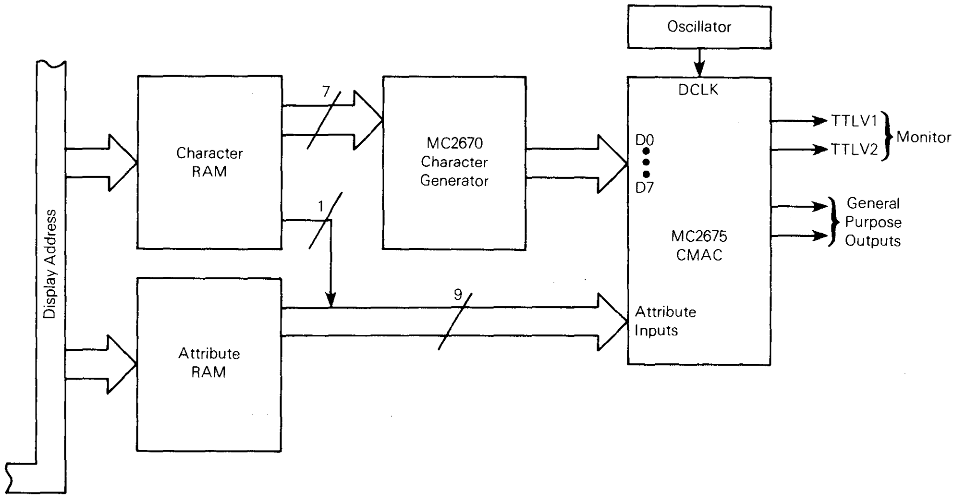


FIGURE 3 — SYSTEM BLOCK DIAGRAM OF MC2675 IN MONOCHROME MODE



tied high, in which case the area outside the active area will be dark, or it may be pulsed during BLANK periods to externally control the border widths.

In color mode, eight colors for the character (foreground) and eight colors for the background (area other than character) can be selected by the attribute inputs. In monochrome mode, the intensities of foreground and background are a function of the attribute and BKGND inputs, i.e., characters may be black, gray, white, or highlight (very white) while background may be black, gray, or white (see Table 1).

CHARACTER CLOCK COUNTER

The character clock counter divides the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (C1-C0) as follows:

C1	C0	MC2675B Dots/ Character	MC2675B $\overline{CCLK}$ Duty Cycle*	MC2675C Dots/ Character	MC2675C $\overline{CCLK}$ Duty Cycle*
0	0	10	5/5	10	5/5
0	1	7	4/3	6	3/3
1	0	8	4/4	8	4/4
1	1	9	5/4	9	5/4

*High/Low

TABLE 1 — MONOCHROME MODE ATTRIBUTE CHARACTERISTICS

REV*	AHILT	ABLINK**	Foreground Video	Background Video
0	0	0	W	B
0	0	1	W/G	B
0	1	0	H	B
0	1	1	H/W	B
1	0	0	B	G
1	0	1	B/W	G/B
1	1	0	B	W
1	1	1	B/H	W/B

*REV=(BKGND) XOR (AVRID):

BKGND	ARVID	REV
0	0	0
0	1	1
1	0	1
1	1	0

**For blinking, the video outputs are shown as zero/one, where zero and one are the blink timing input states.

NOTES:

1. Foreground includes underline when underlining is specified by AUL=1.
2. When ABLANK=1, foreground component becomes same as background component.
3. Codes for video outputs are as follows:

Code	TTLV2	TTLV1	Beam Intensity
B	0	0	Black
G	0	1	Gray
W	1	0	White
H	1	1	Highlight

The number of dot clocks/character is normally the number of dots/character as listed above. However, when dot width control is specified, the DCLK input is divided by two before it is applied to the character clock counter resulting in the number of dot clocks/character being double those listed above, although the number of displayed dots/character remains the same. See **DOT MODULATION LOGIC.**

VIDEO SHIFT REGISTER

On each character boundary, the parallel input dot data (D0-D8) is loaded into the video shift register. The data is shifted out least significant bit first (D0) at the DCLK rate. If ten dots/character are specified (C1-C0 = 00), the tenth dot will be the same as D8. The serial dot data from the video shift register is routed to the video logic where it is combined with the cursor and attribute control bits to produce the video data outputs.

MODE SELECT, ATTRIBUTE, AND CURSOR CONTROL

The mode select logic multiplexes the monochrome and color attribute inputs and outputs as specified by the M/C input. The monochrome mode provides blank, reverse video, highlight, and two general purpose attributes. The latter may be used, with external logic, to combine other attributes (e.g., overscore) into the video stream. The color mode provides RGB foreground and background color attributes. Both modes provide double width characters, blink, underline, dot width control, and dot stretching.

The cursor and attribute inputs are pipelined internally to allow for system pipeline propagations. The cursor input signal is delayed internally by two $\overline{CCLK}$s (one for RAM and one for the character generator), while the attribute inputs are delayed for one $\overline{CCLK}$ to account for the delay of the character data through the character generator latches. The attribute timing inputs (BLINK, UL, and DOTS) are clocked into the MC2675 at the beginning of each scan line time by the falling edge of BLANK. Thus, these inputs must be their proper state at the falling edge of BLANK preceding the scan line where they are required to be active. The BLANK signal itself is also delayed internally to provide for the RAM and character generator delays. Internal delays cause the video outputs to be delayed relative to $\overline{CCLK}$.

VIDEO LOGIC

Each character block consists of the three components shown in Figure 4. Symbol video is generated from the dot data inputs D0-D8. Underline video is enabled by the AUL attribute and is generated during the scan lines for which the UL input is active. Underline and symbol video are always the same intensity or color, and other attributes (e.g., ABLINK) apply to them equally. The combination of underline and symbol video is also referred to as foreground video. Background video is the area of the character block corresponding to the absence of foreground video. The assertion of the non-display attribute (ABLANK) causes the entire character block to be displayed as background.

In monochrome mode, the serial dot data and pipelined cursor and attributes are combined to generate four video intensities (black, gray, white, and highlight) which are encoded on the TTLV1 and TTLV2 outputs as follows:

TTLV2	TTLV1	Video Intensity
0	0	Black
0	1	Gray
1	0	White
1	1	Highlight

FIGURE 4 — CHARACTER BLOCK DEFINITION

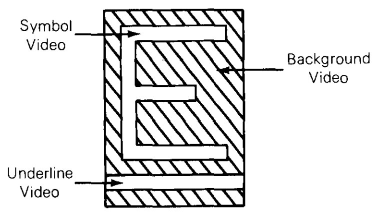


Table 1 describes the relationship between attributes and video intensity of the foreground and background components of the character block in monochrome mode.

In color mode, the colors of the foreground and background components are specified by the corresponding attribute inputs; AREDF, AGREENF, and ABLUEF dictate the color of the foreground components while AREDB, AGREENB, and ABLUEB do the same for the background component. In this mode, the serial dot data and pipelined cursor and attributes are combined to generate four video outputs. The RED, GREEN, and BLUE outputs separately contain the corresponding foreground and background components. The LUM output is the logical OR of the foreground colors and can be used to drive a separate monochrome monitor or to select a different set of colors for the foreground.

DOT MODULATION LOGIC

The dot modulation logic controls the video shift register to supply dot stretching and dot width control.

Dot stretching is controlled by the DOTS input which is sampled each scan line at the trailing (falling) edge of BLANK. If DOTS is asserted at that time, all characters on the following scan line will have dot stretching applied. Dot stretching causes an extra dot to be added to individual dots or groups of dots as shown in Figures 5 and 6. Dot stretching can be used to:

1. Compensate for low video backwidth monitors (since the minimum active displayed segment with dot stretching is two DCLKs).
2. Assure crisp black characters when operating in white background mode.
3. Provide thick characters as a means of distinguishing areas of the display.

FIGURE 5 — DOT MODULATION TIMING

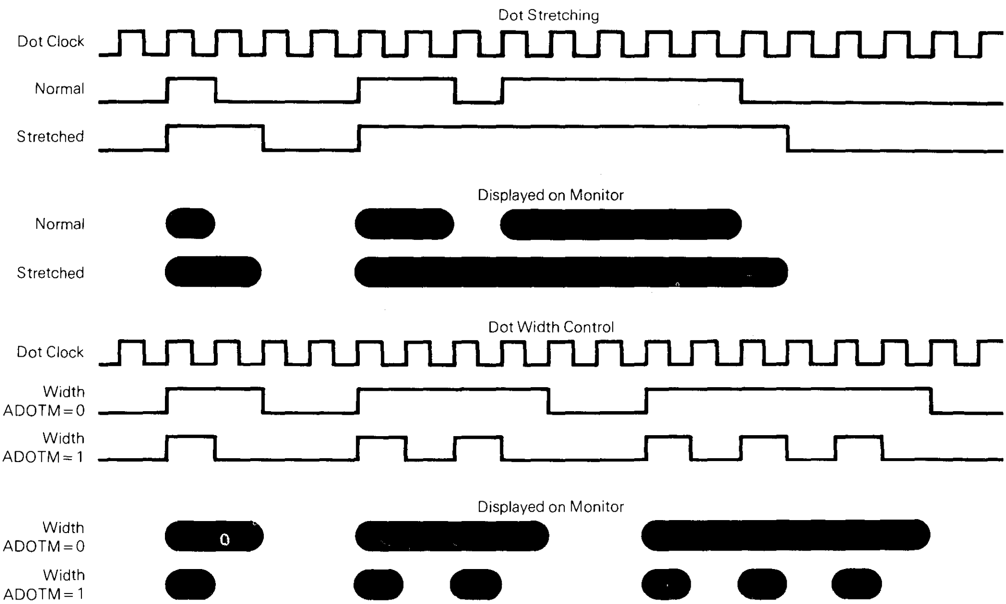
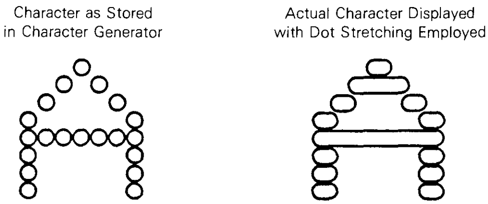


FIGURE 6 — DOT STRETCHING



Dot width is controlled by the DOTM and ADOTM inputs. DOTM is tied either high, which enables the features on the entire display, or low, which disables the feature. With ADOTM high, the dot width of characters can be selectively controlled by assertion of the ADOTM attribute input. When operating in this mode, the dot clock input is divided by two before being applied to other circuits in the CMAC. This affects the $\overline{\text{CCLK}}$ output.

When dot width control is enabled as above, two DCLKs are used for each video dot period. Asserting ADOTM for a particular character will cause each active video dot of the displayed character to be turned on for one DCLK and off for the other DCLK, while if ADOTM is negated for that character, the active video dot for that character will be turned on (black background) or off (white background) for both DCLK times (see Figures 5 and 7). Only the character video components of the character block are modulated. Underline video and background are not affected by on-time modulation. Width control can be used to:

1. Make horizontal lines and vertical lines appear the same brightness on the display.
2. Provide two different brightness levels for characters without requiring a monitor with analog brightness inputs.

However, note that the effects produced by this feature are highly dependent on the video amplifier characteristics of the monitor used.

FIGURE 7 — DOT WIDTH CONTROL

Normal Character Display Without Width Control | Actual Character Display with Width Control

DOUBLE WIDTH LOGIC

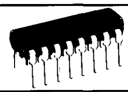
The double width logic controls the rate at which dots are shifted through the video shift register. When the ADOUBLE input is asserted, the associated character video will be shifted at one half the DCLK rate, and the dot information for the next character will be loaded into the shift register two $\overline{\text{CCLK}}$s later. The $\overline{\text{CCLK}}$ output is not affected. If a double width character is specified at the last location of a character row, the second half of the double width character (one $\overline{\text{CCLK}}$) will extent into the horizontal front porch.

MOTOROLA

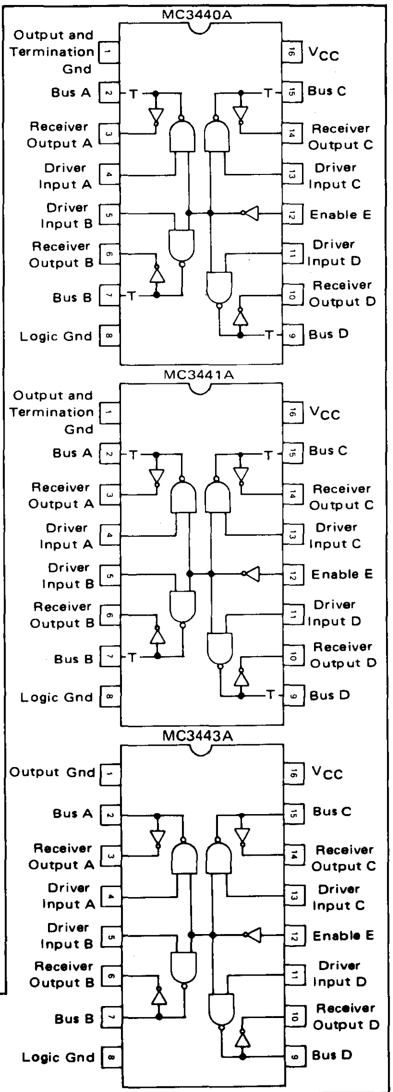
MC3440A
MC3441A
MC3443A

QUAD INTERFACE BUS TRANSCEIVERS

SILICON MONOLITHIC INTEGRATED CIRCUITS



P SUFFIX
PLASTIC PACKAGE
CASE 648-05



— T — = Bus Termination

QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVERS

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

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

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

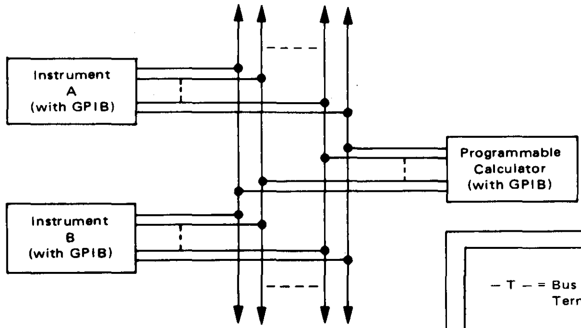
The MC3443A is identical to the MC3441A except that the terminations have been omitted. As such it is pin compatible, and functionally equivalent to the SN75138. It does offer the advantage of receiver input hysteresis.

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

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

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

TYPICAL APPLICATION — GPIB MEASUREMENT SYSTEM



3

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, $4.5\ V \leq V_{CC} \leq 5.5\ V$ and $0 \leq T_A \leq 70^{\circ}C$, typical values are at $T_A = 25^{\circ}C$, $V_{CC} = 5.0\ V$)

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

SWITCHING CHARACTERISTICS ($V_{CC} = 5.0\ V$, $T_A = 25^{\circ}C$)

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

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

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

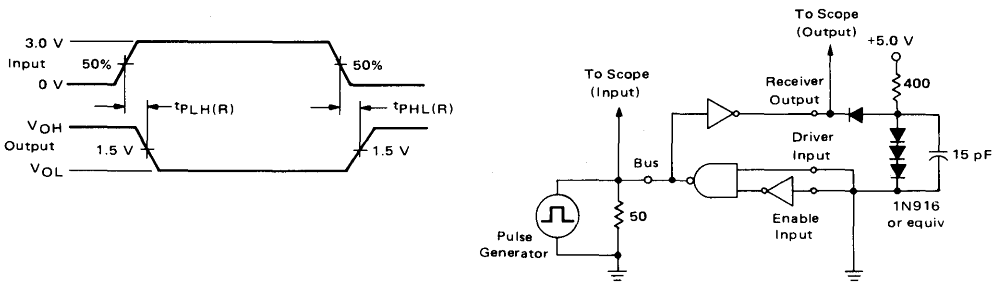


FIGURE 2 — TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)

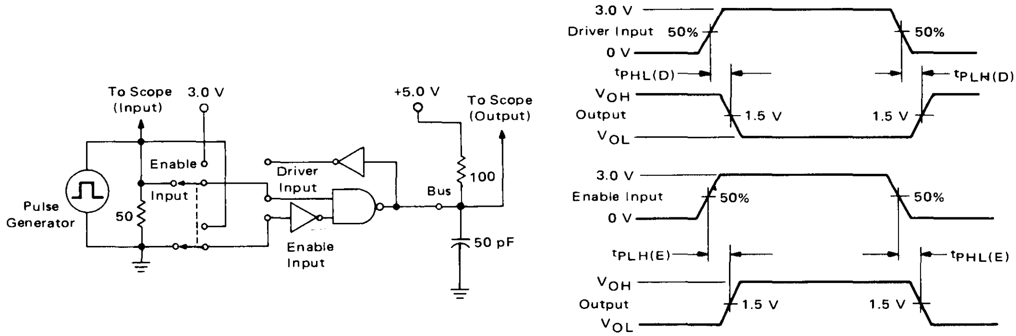
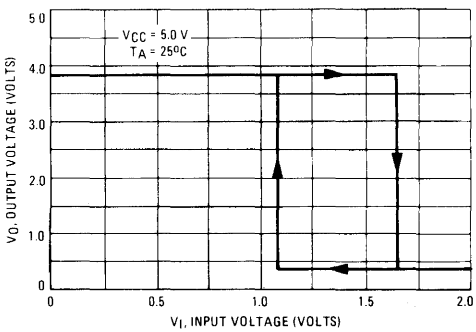
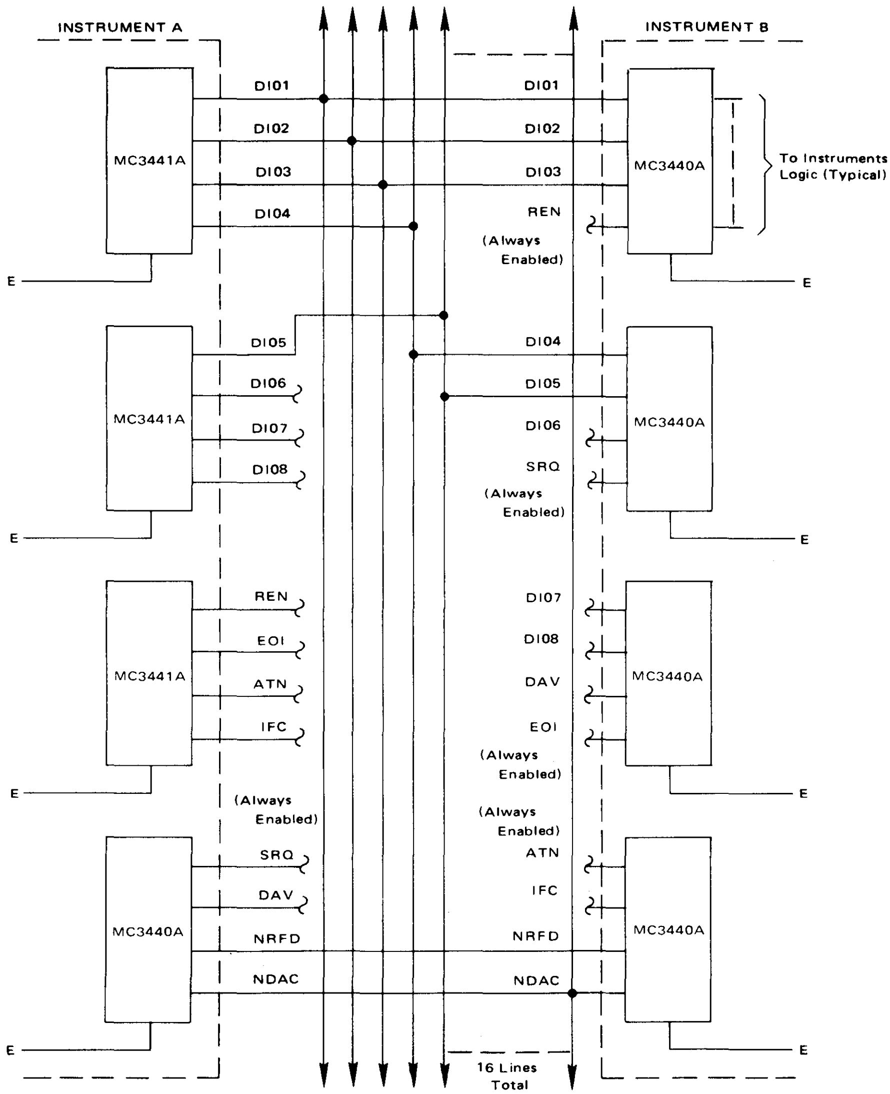


FIGURE 3 — TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS



GENERAL PURPOSE INTERFACE BUS APPLICATION



GPIB SIGNALS:

8 Line Data Bus: DI01 – DI08
5 General Interrupt Transfer Control Bus:
- REN – Remote Enable
- SRQ – Service Request
- EOI – End or Identify
- ATN – Attention
- IFC – Interface Clear

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

16 Total Signal Lines

MC3446A

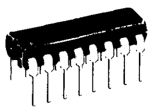
QUAD GENERAL-PURPOSE INTERFACE BUS (GPIB) TRANSCEIVER

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

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

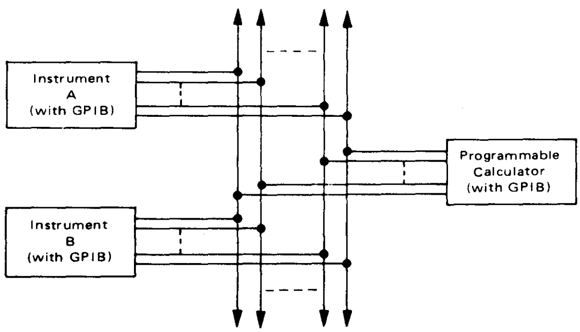
QUAD INTERFACE BUS TRANSCEIVER

SILICON MONOLITHIC INTEGRATED CIRCUIT

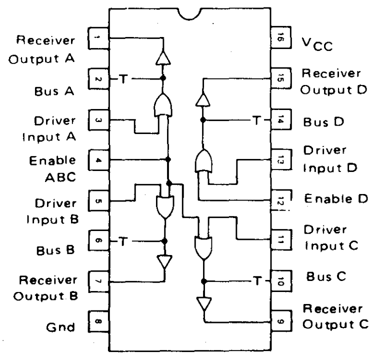


P SUFFIX
PLASTIC PACKAGE
CASE 648-05

TYPICAL MEASUREMENT SYSTEM APPLICATION



PIN CONNECTIONS



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

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

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

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

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

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

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

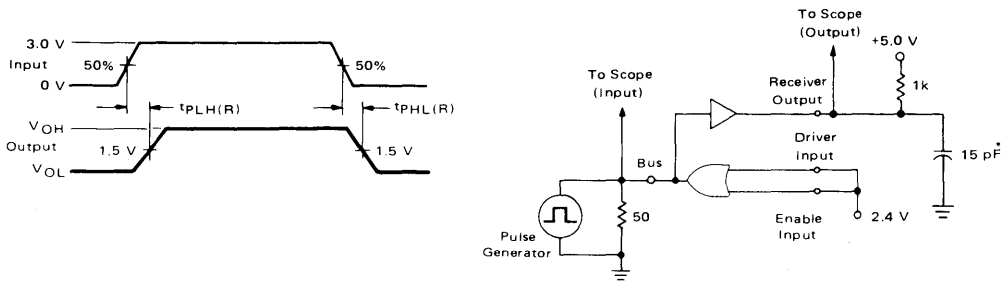
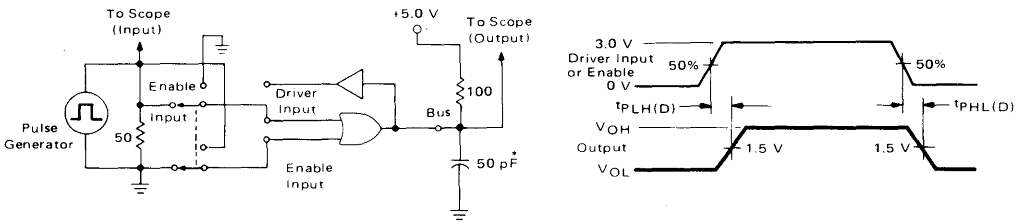


FIGURE 2 — TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER AND COMMON ENABLE INPUTS TO OUTPUT (BUS)



* Includes Probe and Jig Capacitance

FIGURE 3 — TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

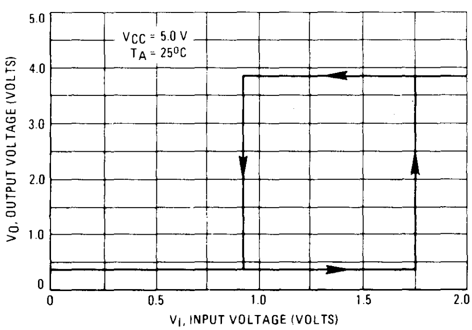
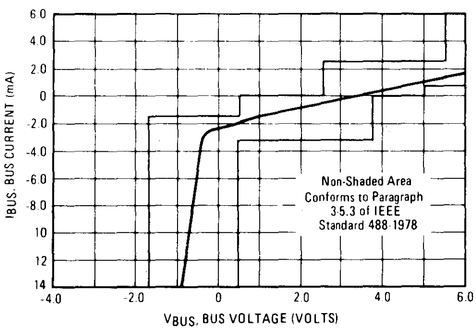


FIGURE 4 — TYPICAL BUS LOAD LINE



MOTOROLA

MC3447

BIDIRECTIONAL INSTRUMENTATION BUS (GPIB) TRANSCEIVER

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

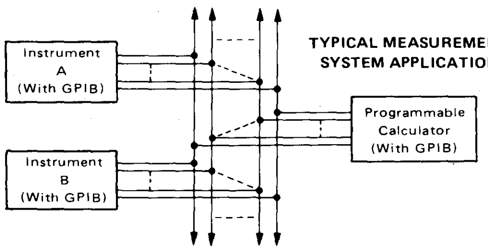
Low power consumption has been achieved by trading a minimum of speed for low current drain on non-critical channels. A fast channel is provided for critical ATN and EOI paths.

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

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

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

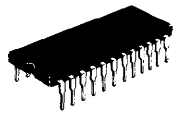
Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
Junction Temperature	T_J	150	°C
Operating Ambient Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C



TYPICAL MEASUREMENT SYSTEM APPLICATION

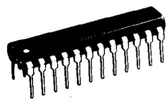
OCTAL BIDIRECTIONAL BUS TRANSCEIVER WITH TERMINATION NETWORKS

SILICON MONOLITHIC INTEGRATED CIRCUIT

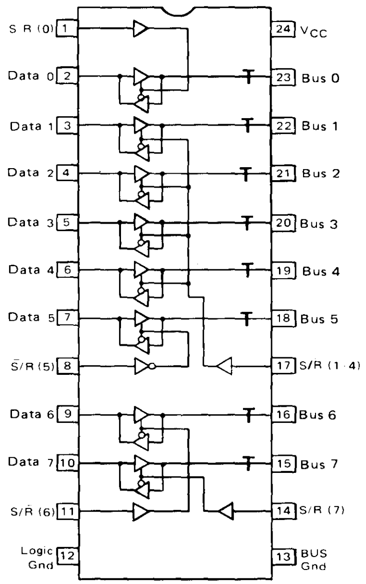


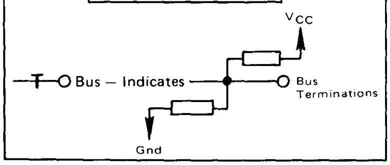
L SUFFIX
CERAMIC PACKAGE
CASE 623-05

P3 SUFFIX
PLASTIC PACKAGE
CASE 724-02



PIN ASSIGNMENTS





ELECTRICAL CHARACTERISTICS

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

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

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

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

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

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

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

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

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

PROPAGATION DELAY TEST CIRCUITS AND WAVEFORMS

FIGURE 1 — BUS INPUT TO DATA OUTPUT (RECEIVER)

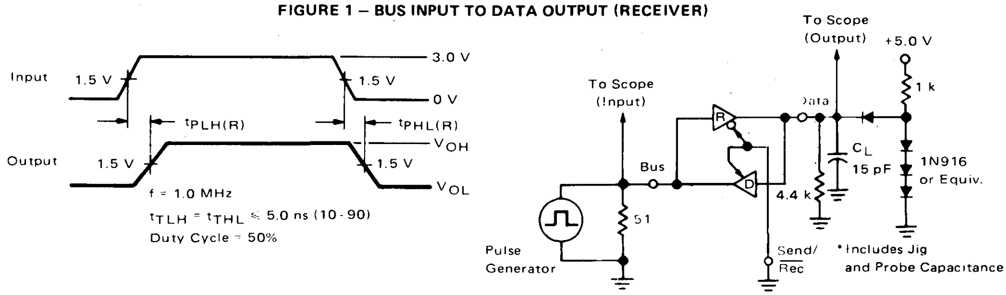


FIGURE 2 — DATA INPUT TO BUS OUTPUT (DRIVER)

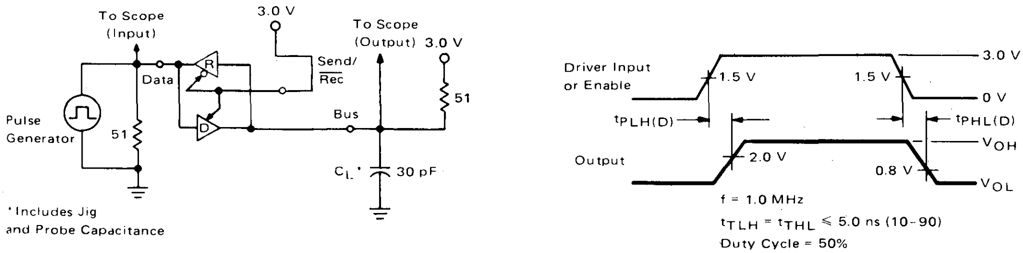


FIGURE 3 — SEND/$\overline{\text{RECEIVE}}$ INPUT TO BUS OUTPUT (DRIVER)

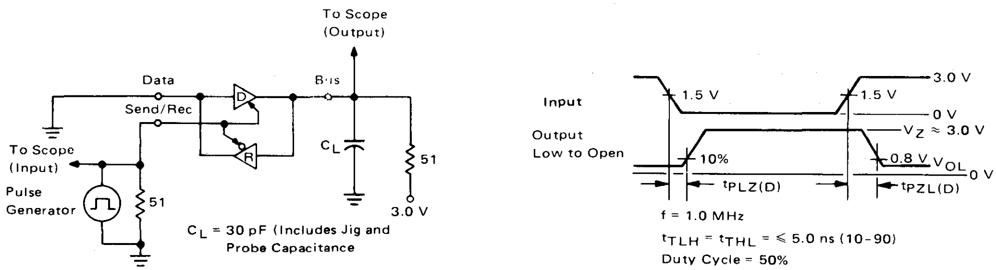


FIGURE 4 — SEND/$\overline{\text{RECEIVE}}$ INPUT TO DATA OUTPUT (RECEIVER)

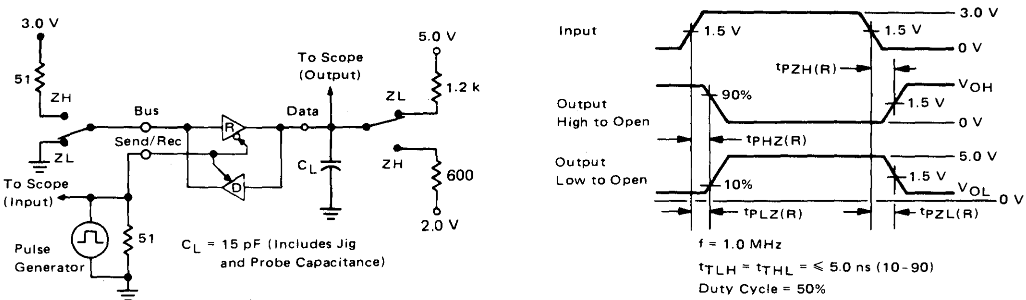


FIGURE 5 — TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

FIGURE 6 — TYPICAL BUS LOAD LINE

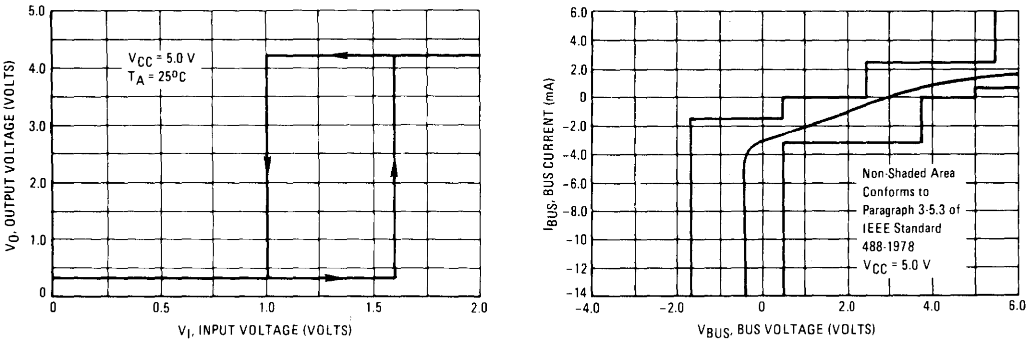


FIGURE 7 — SUGGESTED PRINTED CIRCUIT BOARD LAYOUT USING MC3447s AND MC68488

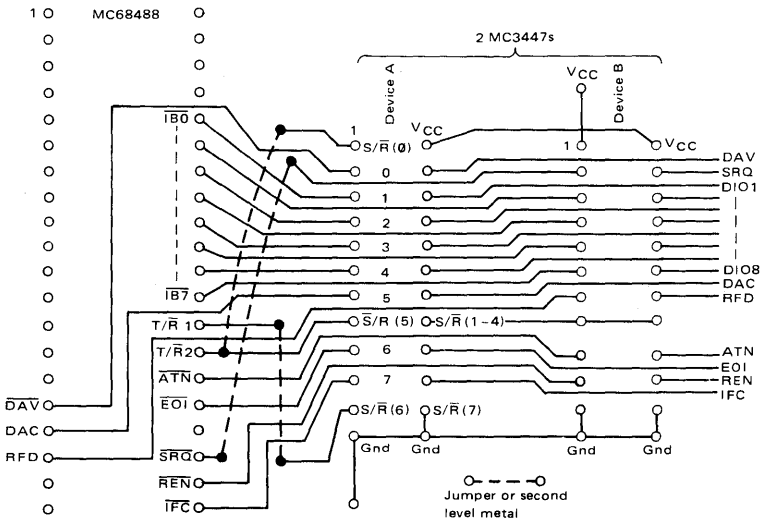
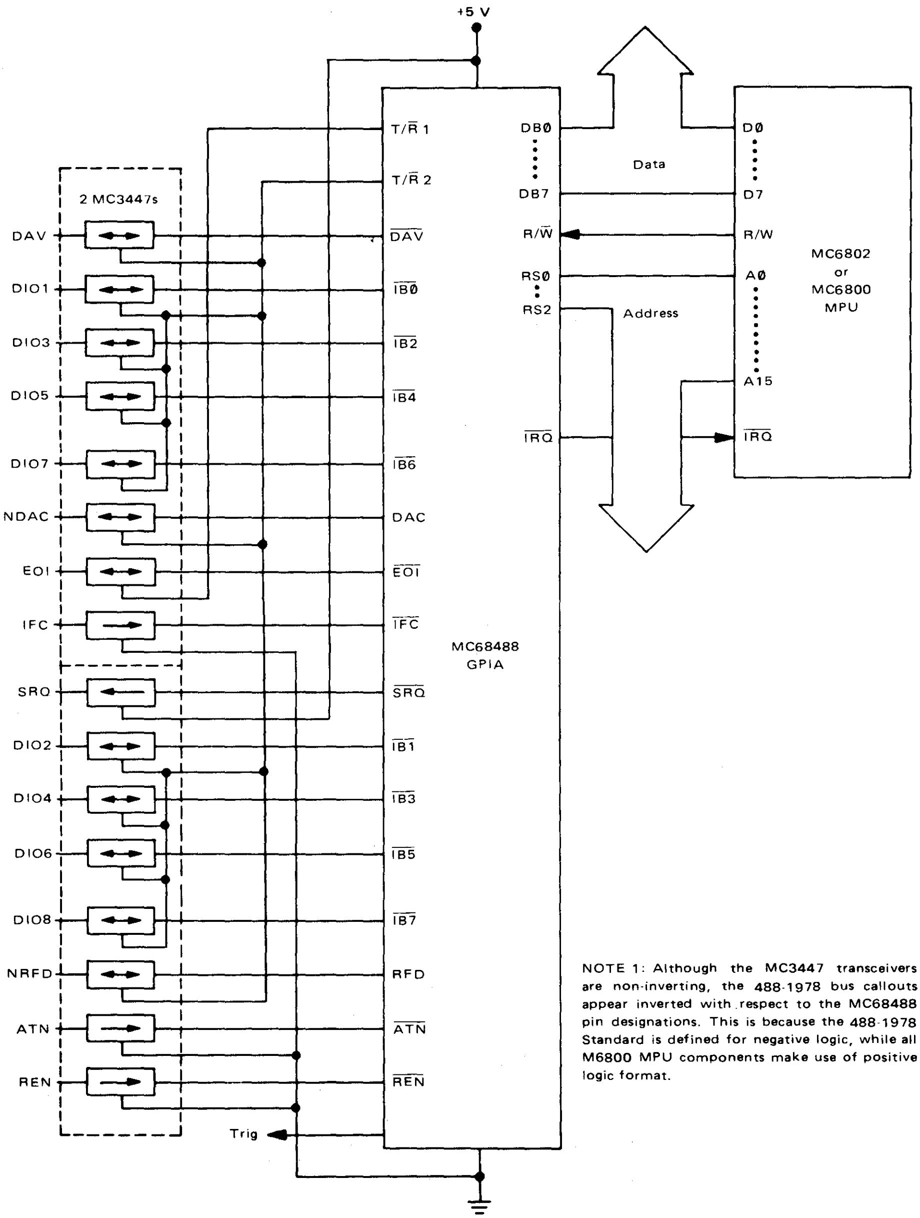


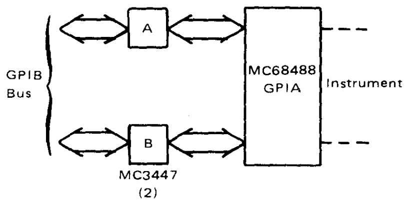
FIGURE 8 – SIMPLE SYSTEM CONFIGURATION



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

FIGURE 9 – SUGGESTED PIN DESIGNATIONS FOR USE WITH MC68488

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



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.

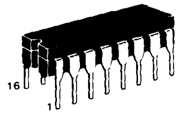
Each driver/receiver pair forms the complete interface between the bus and an instrument. Either the driver or the receiver of each channel is enabled by its corresponding Send/Receive input with the disabled output of the pair forced to a high impedance state. An additional option allows the driver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

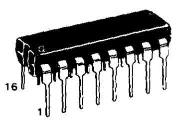
(1) Selection of the "Open Collector" configuration, in fact, selects an open collector device with a passive pull-up load/termination which conforms to Figure 7, IEEE 488-1978 Bus Standard.

QUAD THREE-STATE BUS TRANSCEIVER WITH TERMINATION NETWORKS

SILICON MONOLITHIC INTEGRATED CIRCUIT



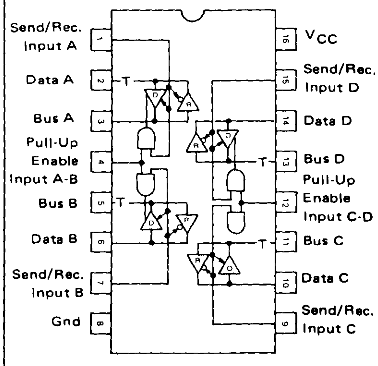
L SUFFIX
CERAMIC PACKAGE
CASE 620-02



P SUFFIX
PLASTIC PACKAGE
CASE 648-05

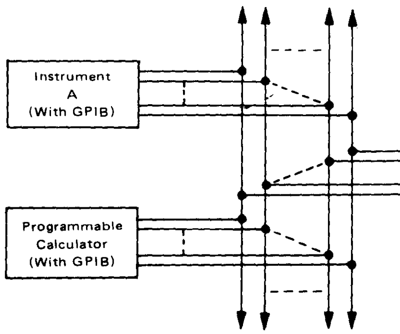
MAXIMUM RATINGS ($T_A = 25^{\circ}C$ unless otherwise noted)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	5.5	Vdc
Driver Output Current	$I_{O(D)}$	150	mA
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



–T– = Bus Termination

TYPICAL MEASUREMENT SYSTEM APPLICATION



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 $\leq V_{CC} \leq$ 5.25 V and 0 $\leq T_A \leq$ 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 $\leq V_{(BUS)} \leq$ 5.5 V)		0.7	–	2.5	
($V_{(BUS)}$ = 0.5 V)		-1.3	–	-3.2	
(V_{CC} = 0 V, 0 V $\leq V_{(BUS)} \leq$ 2.75 V)		–	–	+0.04	
Receiver Input Hysteresis ($V_{I(S/R)}$ = 0.8 V)	–	400	600	–	mV
Receiver Input Threshold					V
($V_{I(S/R)}$ = 0.8 V, Low to High)	$V_{ILH(R)}$	–	1.6	1.8	
($V_{I(S/R)}$ = 0.8 V, High to Low)	$V_{IHL(R)}$	0.8	1.0	–	
Receiver Output Voltage – High Logic State ($V_{I(S/R)}$ = 0.8 V, $I_{OH(R)}$ = -800 μA, $V_{(BUS)}$ = 2.0 V)	$V_{OH(R)}$	2.7	–	–	V
Receiver Output Voltage – Low Logic State ($V_{I(S/R)}$ = 0.8 V, $I_{OL(R)}$ = 16 mA, $V_{(BUS)}$ = 0.8 V)	$V_{OL(R)}$	–	–	0.5	V
Receiver Output Short Circuit Current ($V_{I(S/R)}$ = 0.8 V, $V_{(BUS)}$ = 2.0 V)	$I_{OS(R)}$	-15	–	-75	mA
Driver Input Voltage – High Logic State ($V_{I(S/R)}$ = 2.0 V)	$V_{IH(D)}$	2.0	–	–	V
Driver Input Voltage – Low Logic State ($V_{I(S/R)}$ = 2.0 V)	$V_{IL(D)}$	–	–	0.8	V
Driver Input Current – Data Pins ($V_{I(S/R)}$ = $V_{I(E)}$ = 2.0 V)					μA
(0.5 $\leq V_{I(D)} \leq$ 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 $\leq V_{I(S/R)} \leq$ 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 $\leq V_{I(E)} \leq$ 2.7 V)	$I_{I(E)}$	-200	–	20	
($V_{I(E)}$ = 5.5 V)	$I_{IB(E)}$	–	–	100	
Driver Input Clamp Voltage ($V_{I(S/R)}$ = 2.0 V, $I_{IC(D)}$ = -18 mA)	$V_{IC(D)}$	–	–	-1.5	V
Driver Output Voltage – High Logic State ($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V, I_{OH} = -5.2 mA)	$V_{OH(D)}$	2.5	–	–	V
Driver Output Voltage – Low Logic State (Note 1) ($V_{I(S/R)}$ = 2.0 V, $I_{OL(D)}$ = 48 mA)	$V_{OL(D)}$	–	–	0.5	V
Output Short Circuit Current ($V_{I(S/R)}$ = 2.0 V, $V_{IH(D)}$ = 2.0 V, $V_{IH(E)}$ = 2.0 V)	$I_{OS(D)}$	-30	–	-120	mA
Power Supply Current					mA
(Listening Mode – All Receivers On)	I_{CCL}	–	63	85	
(Talking Mode – All Drivers On)	I_{CCH}	–	106	125	

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

Characteristic	Symbol	Min	Typ	Max	Unit
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)

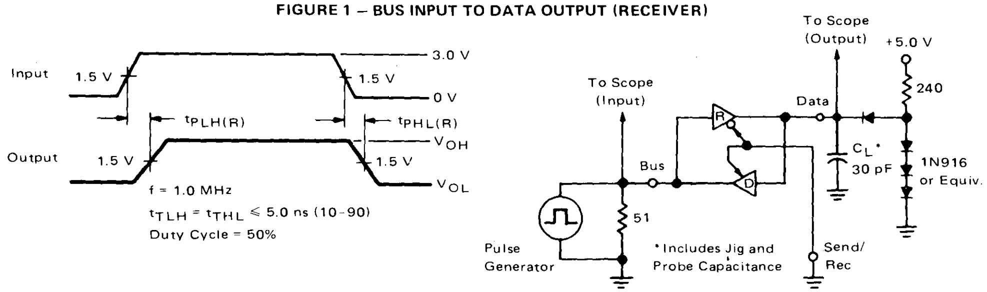


FIGURE 2 – DATA INPUT TO BUS OUTPUT (DRIVER)

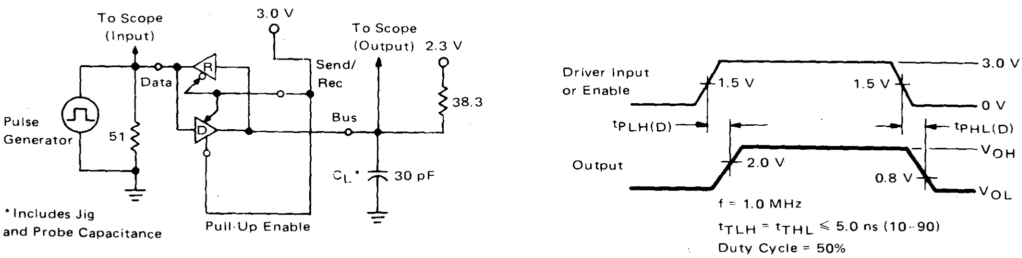


FIGURE 3 – SEND/RECEIVE INPUT TO BUS OUTPUT (DRIVER)

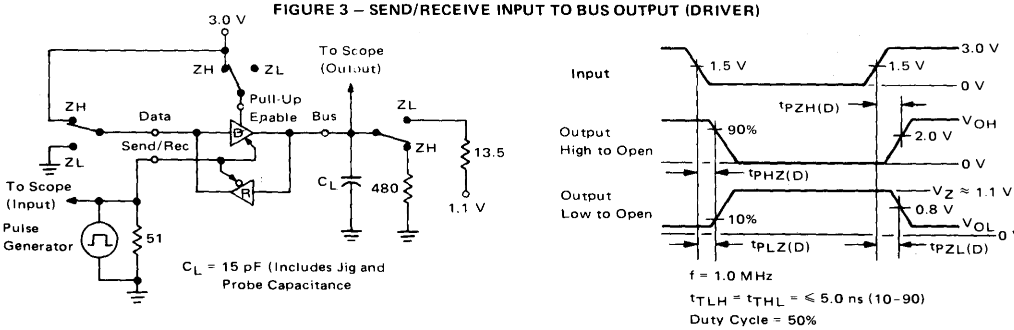


FIGURE 4 – SEND/RECEIVE INPUT TO DATA OUTPUT (RECEIVER)

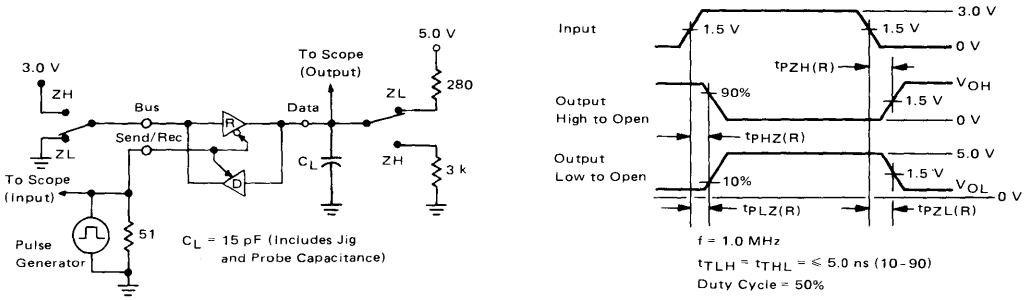


FIGURE 5 – ENABLE INPUT TO BUS OUTPUT (DRIVER)

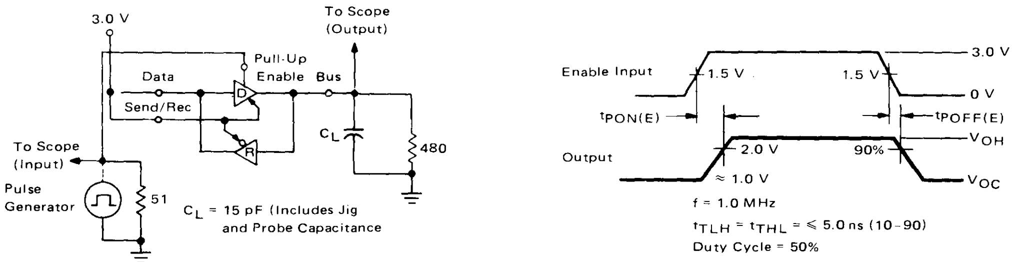


FIGURE 6 – TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

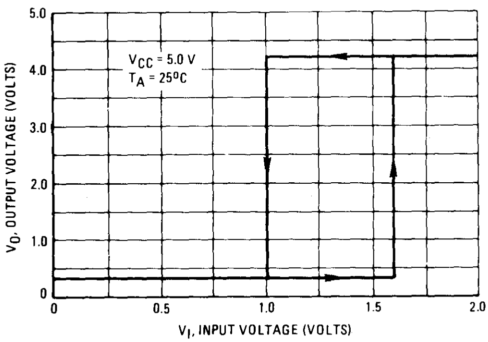


FIGURE 7 – TYPICAL BUS LOAD LINE

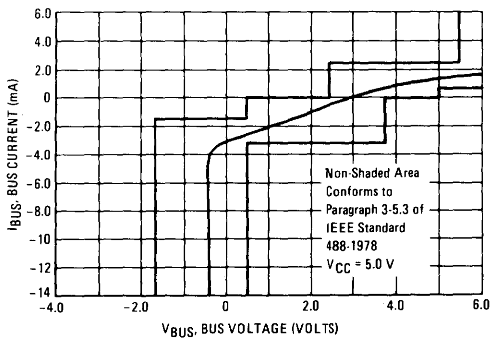
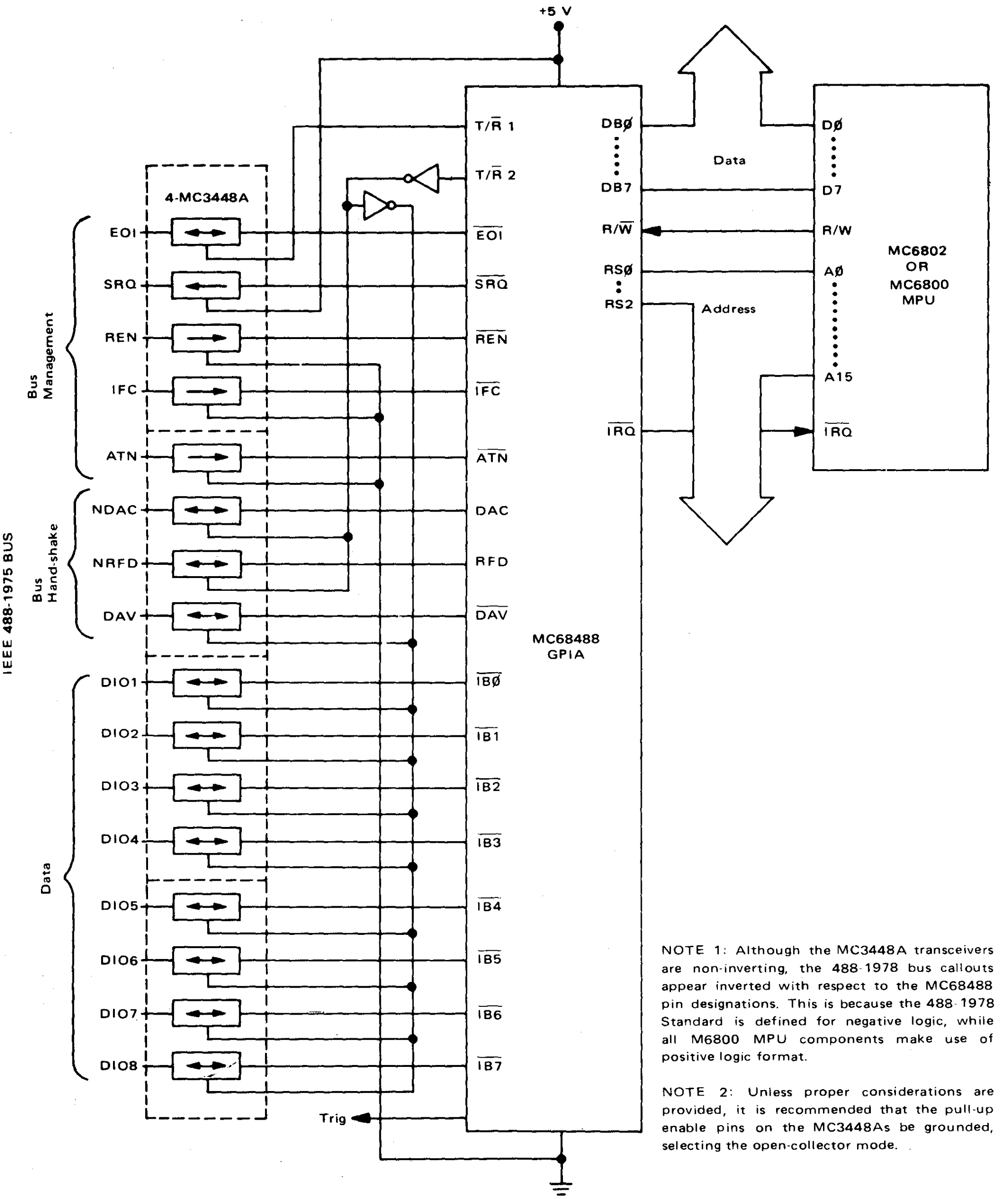


FIGURE 8 – SIMPLE SYSTEM CONFIGURATION



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

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

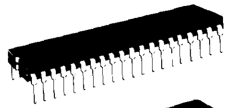
MOTOROLA

MC6800

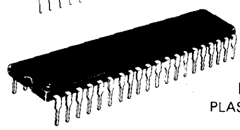
MOS

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

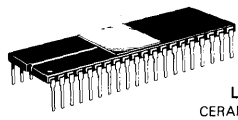
MICROPROCESSOR



S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

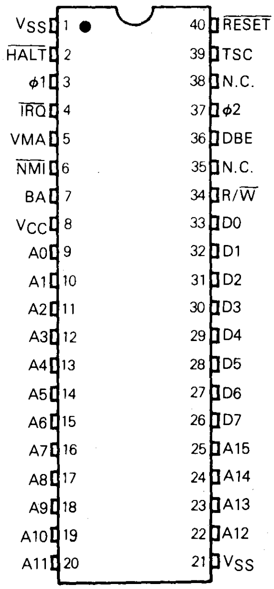
8-BIT MICROPROCESSING UNIT (MPU)

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

PIN ASSIGNMENT



ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6800L
	1.0	−40°C to 85°C	MC6800CL
	1.5	0°C to 70°C	MC68A00L
	1.5	−40°C to 85°C	MC68A00CL
	2.0	0°C to 70°C	MC68B00L
Cerdip S Suffix	1.0	0°C to 70°C	MC6800S
	1.0	−40°C to 85°C	MC6800CS
	1.5	0°C to 70°C	MC68A00S
	1.5	−40°C to 85°C	MC68A00CS
	2.0	0°C to 70°C	MC68B00S
Plastic P Suffix	1.0	0°C to 70°C	MC6800P
	1.0	−40°C to 85°C	MC6800CP
	1.5	0°C to 70°C	MC68A00P
	1.5	−40°C to 85°C	MC68A00CP
	2.0	0°C to 70°C	MC68B00P

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

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}).

THERMAL RESISTANCE

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

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

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

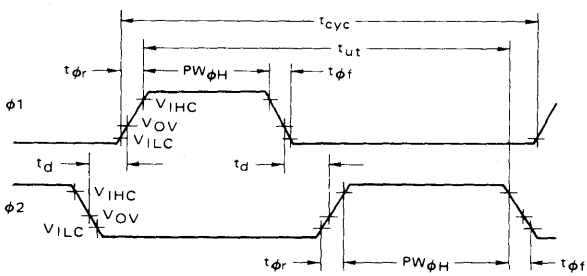
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
	$\phi1, \phi2$	V_{IHC}	$V_{CC}-0.6$	–	$V_{CC}+0.3$	
Input Low Voltage	Logic	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
	$\phi1, \phi2$	V_{ILC}	$V_{SS}-0.3$	–	$V_{SS}+0.4$	
Input Leakage Current (V_{in}=0 to 5.25 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	
Hi-Z Input Leakage Current (V_{in}=0.4 to 2.4 V, V_{CC}=Max)	D0-D7	I_{IZ}	–	2.0	10	μA
	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 (Measured at V_{CC} – 0.6 V)	φ1, φ2 – MC6800	$PW_{\phi H}$	400	–	9500	ns
	φ1, φ2 – MC68A00		230	–	9500	
	φ1, φ2 – MC68B00		180	–	9500	
Total φ1 and φ2 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	

3

FIGURE 1 – CLOCK TIMING WAVEFORM



NOTES:
1. Voltage levels shown are $V_L \leq 0.4$, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.

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

Characteristic	Symbol	MC6800 Min	MC6800 Typ	MC6800 Max	MC68A00 Min	MC68A00 Typ	MC68A00 Max	MC68B00 Min	MC68B00 Typ	MC68B00 Max	Unit
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	
Hi-Z Enable	t_{TSE}	0	–	40	0	–	40	0	–	40	
Hi-Z Delay	t_{TSD}	–	–	270	–	–	270	–	–	220	
Data Bus Enable Down Time During φ1 Up Time	t_{DBE}	150	–	–	120	–	–	75	–	–	
Data Bus Enable Rise and Fall Times	t_{DBEr}, t_{DBEf}	–	–	25	–	–	25	–	–	25	

3

FIGURE 2 — READ DATA FROM MEMORY OR PERIPHERALS

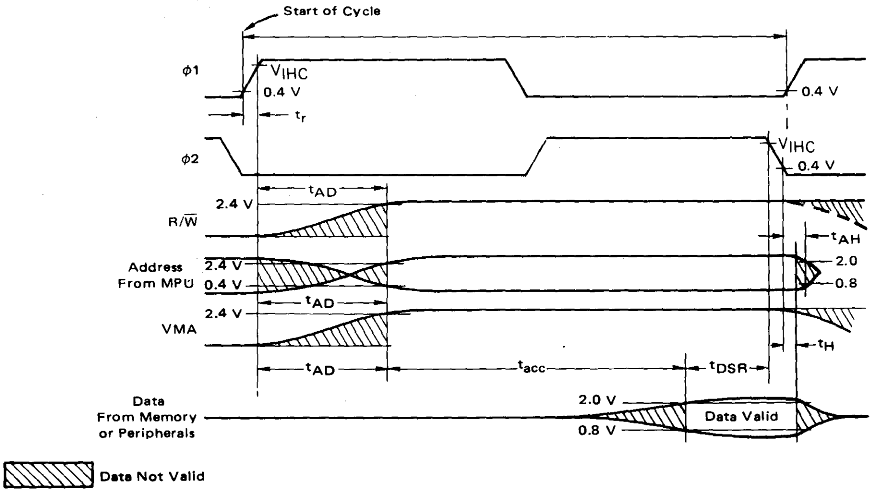
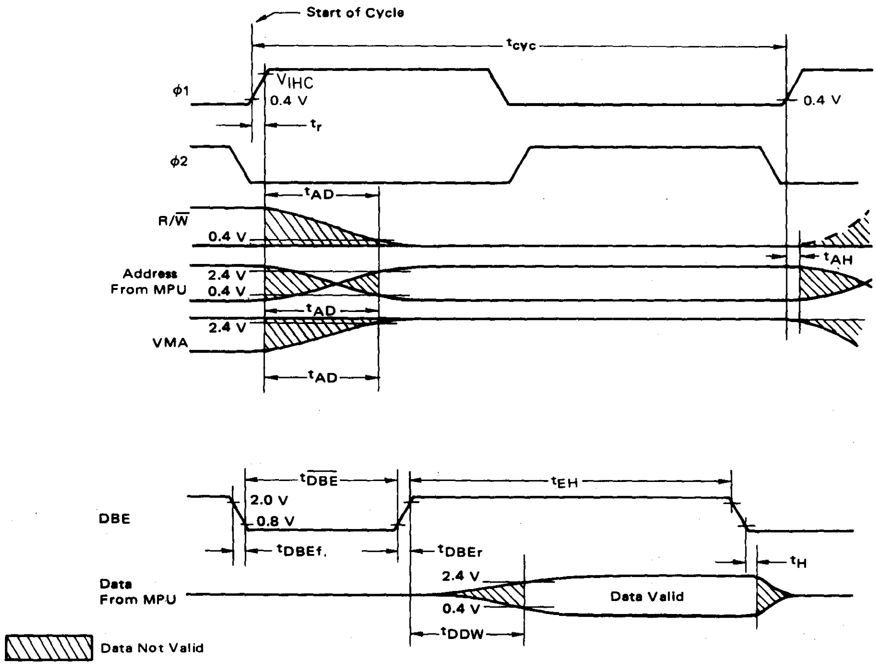


FIGURE 3 — WRITE IN MEMORY OR PERIPHERALS



NOTES:

1. Voltage levels shown are $V_L \leq 0.4$, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.

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

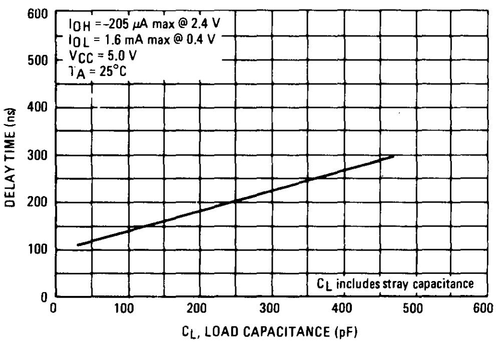


FIGURE 5 — TYPICAL READ/WRITE, VMA, AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING (T_{AD})

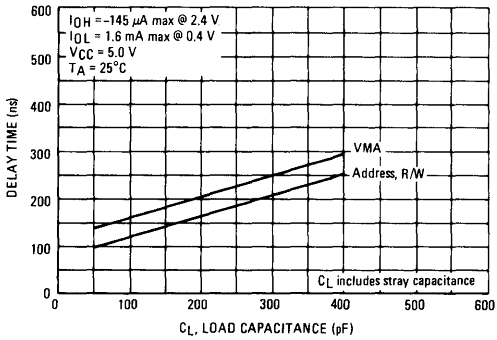
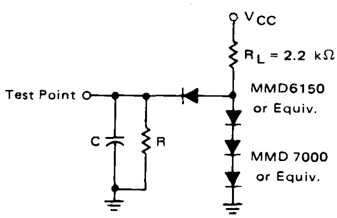


FIGURE 6 — BUS TIMING TEST LOADS



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

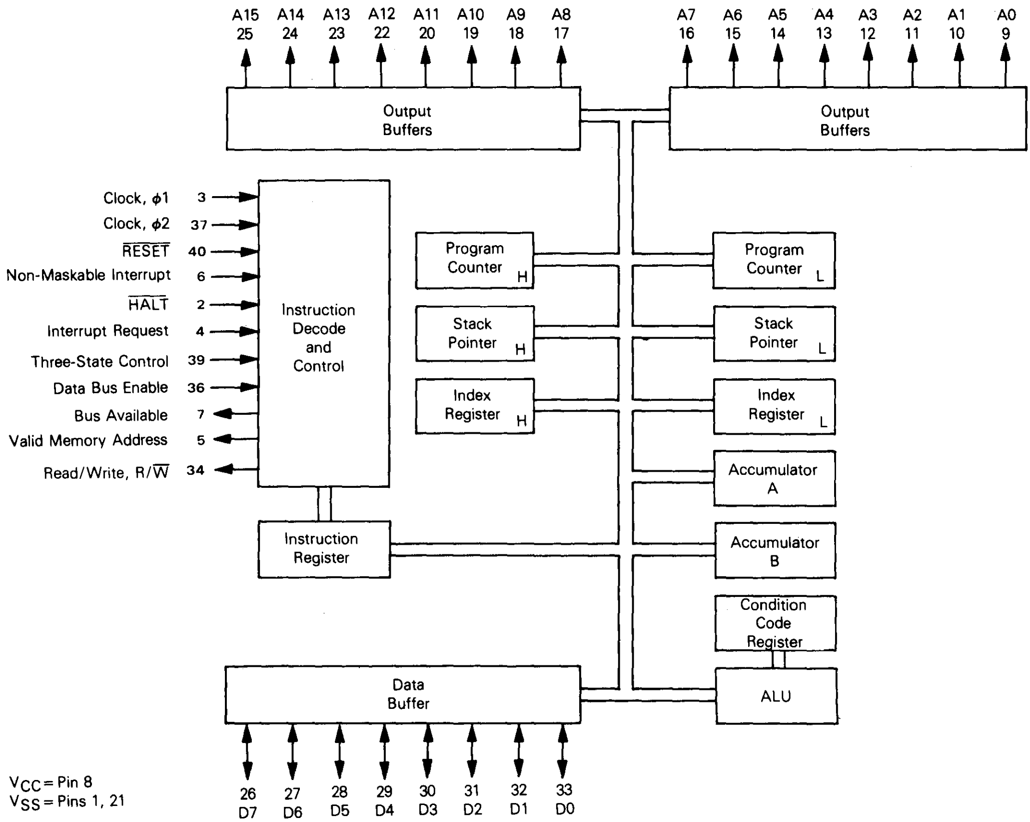
R = 11.7 kΩ for D0-D7
= 16.5 kΩ for A0-A15, R/$\overline{W}$, and VMA
= 24 kΩ for BA

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.

Notice 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 ($\phi 1$, $\phi 2$) — 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 $\phi 1$ and $\phi 2$ 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 $\phi 2$ cycles. The $\overline{\text{RESET}}$ pulse can be completely asynchronous with the MPU system clock and will be recognized during $\phi 2$ 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.

3

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

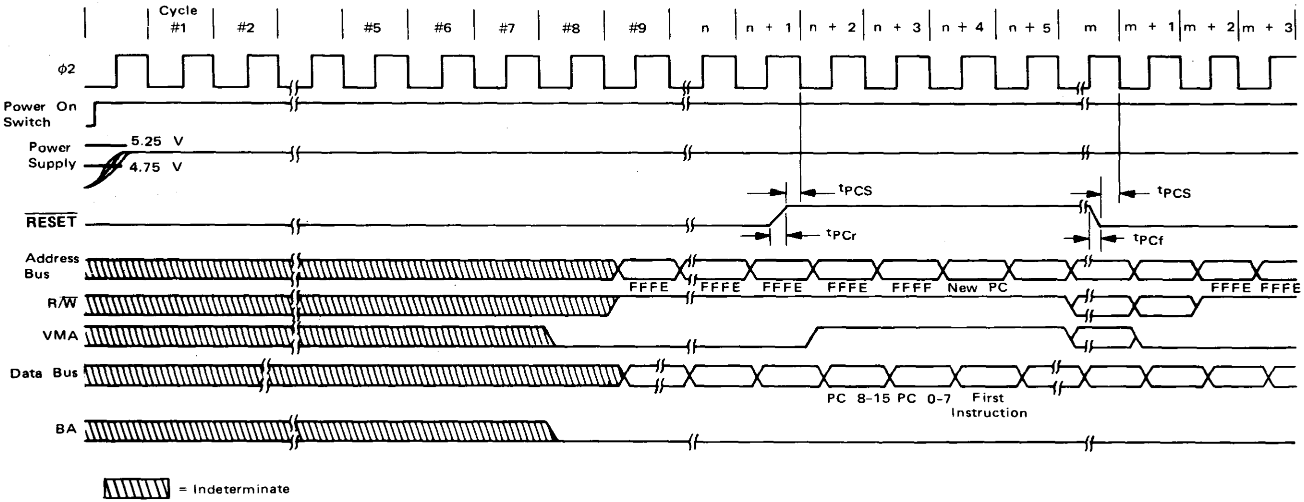
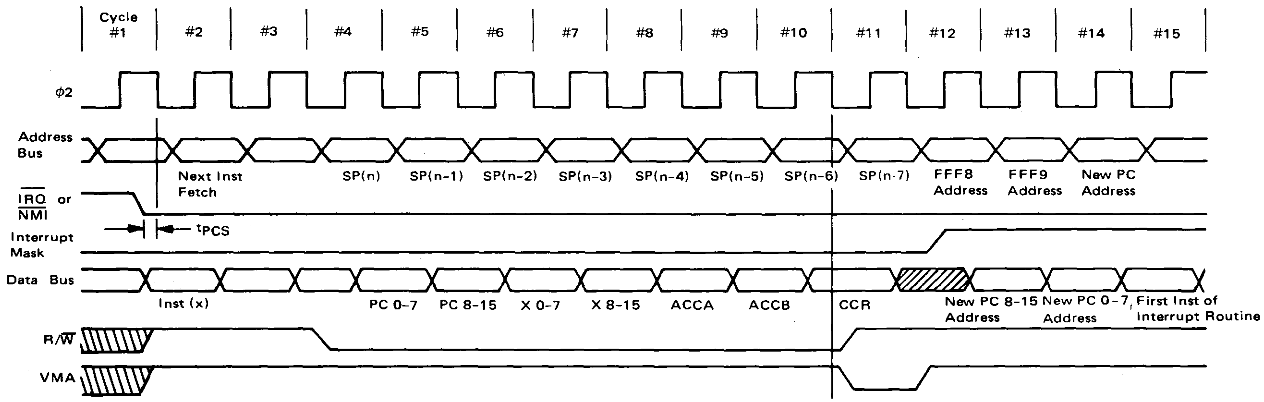


FIGURE 9 — INTERRUPT TIMING



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

The $\overline{IRQ}$ 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 ($\overline{NMI}$) and Wait for Interrupt (WAI) – The MC6800 is capable of handling two types of interrupts: maskable ($\overline{IRQ}$) as described earlier, and non-maskable ($\overline{NMI}$) which is an edge sensitive input. $\overline{IRQ}$ is maskable by the interrupt mask in the condition code register while $\overline{NMI}$ 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 $\overline{IRQ}$ or $\overline{NMI}$ and can be asynchronous with respect to $\phi 2$. 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 $\overline{NMI}$ interrupt and from FFF8, FFF9 for an $\overline{IRQ}$ 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/$\overline{W}$ 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 MS	Vector LS	Description
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/$\overline{W}$ 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 $\phi 1$ (or $\phi 2$) 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 $\phi 1$ 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 $\phi 1$ high and $\phi 2$ low as shown. The Address Bus and R/$\overline{W}$ 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 $\phi 2$ is being held low since DBE = $\phi 2$. 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/$\overline{W}$ 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.

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

The $\overline{HALT}$ line provides an input to the MPU to allow control of program execution by an outside source. If $\overline{HALT}$ 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/$\overline{W}$ 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.

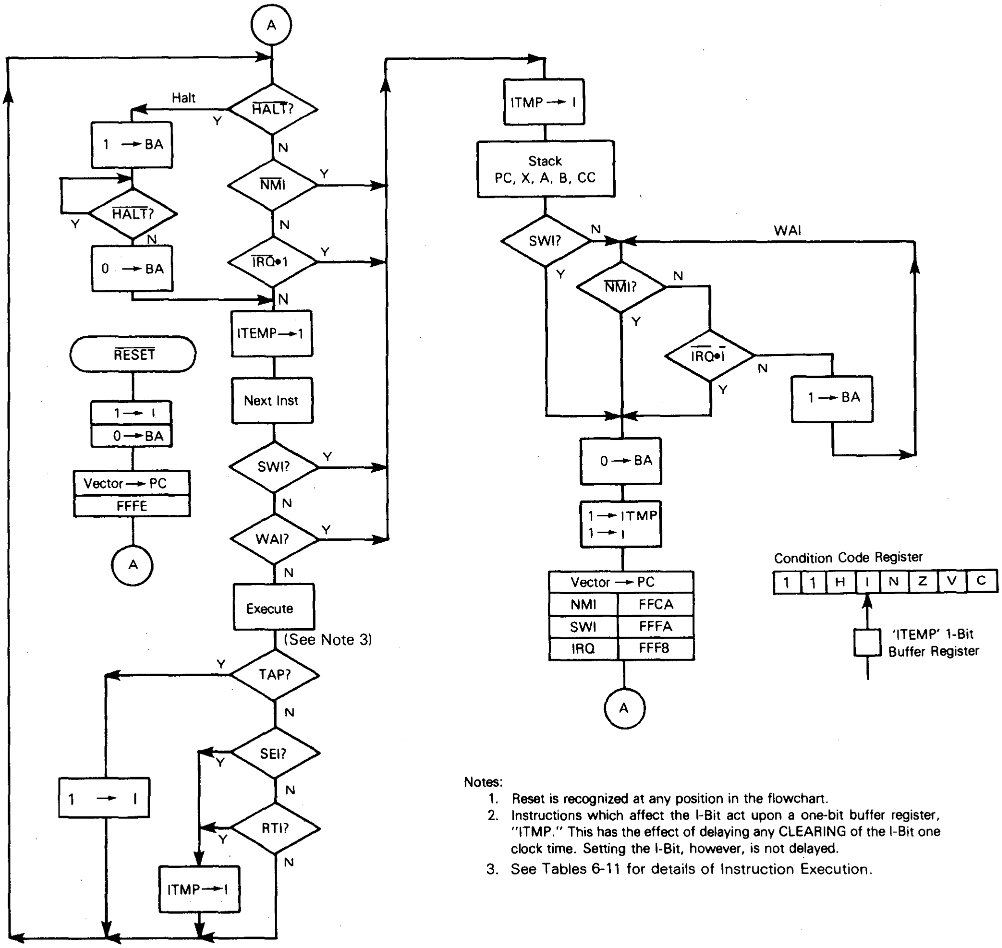
While the MPU is halted, all program activity is stopped, and if either an $\overline{NMI}$ or $\overline{IRQ}$ interrupt occurs, it will be latched into the MPU and acted on as soon as the MPU is taken out of the halted mode. If a $\overline{RESET}$ command occurs while the MPU is halted, the following states occur: VMA = low, BA = low, Data Bus = high impedance, R/$\overline{W}$ = high (read state), and the Address Bus will contain address FFFE as long as $\overline{RESET}$ is low. As soon as the $\overline{RESET}$ 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 $\overline{HALT}$ goes low, the MPU will halt after completing execution of the current instruction. The transition of $\overline{HALT}$ must occur t_{PCS} before the trailing edge of $\phi 1$ of the last cycle of an instruction (point A of Figure 13). $\overline{HALT}$ 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 $\overline{HALT}$ had not been low at Point A but went low during $\phi 2$ 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/$\overline{W}$, 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 $\overline{HALT}$ 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/$\overline{W}$ 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 FLOWCHART



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

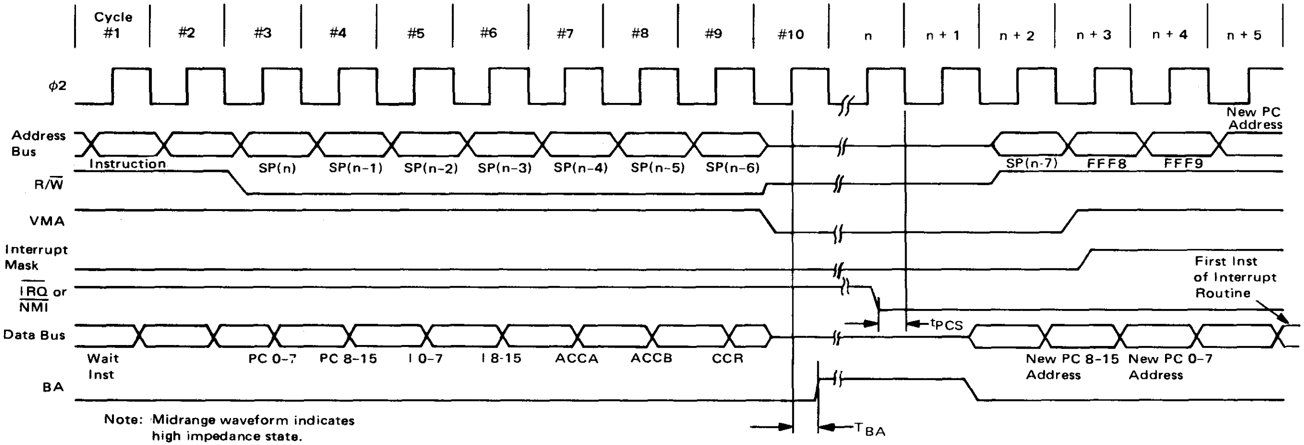


FIGURE 12 — THREE-STATE CONTROL TIMING

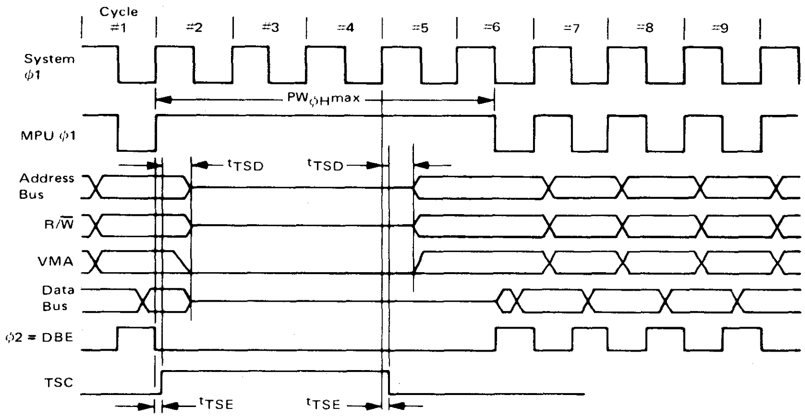
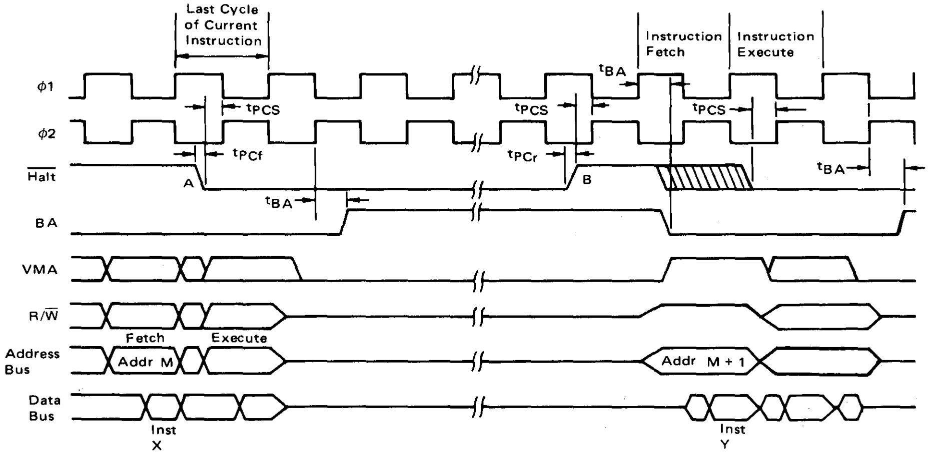


FIGURE 13 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



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.

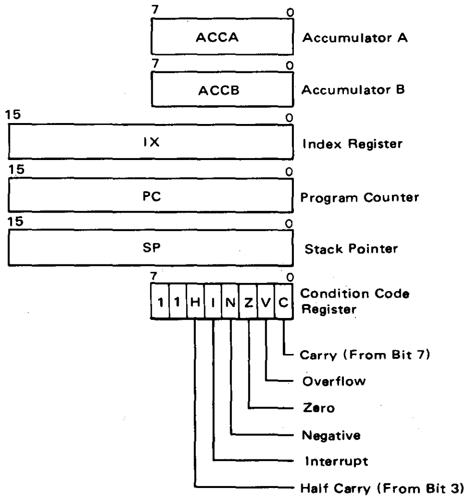
Stack Pointer — The stack ponter is a two byte register that contains the address of the next available location in an external push-down/pop-up stack. This stack is normally a random access Read/Write memory that may have any location (address) that is convenient. In those applications that require 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 "*".

3

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 C ← b7 ← b0	●	●	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M	●	●	↕	↕	⑥	↕
	RORA													46	2	1	A	●	●	↕	↕	⑥	↕
	RORB													56	2	1	B C → b7 → b0	●	●	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	●	●	↕	↕	⑥	↕
	ASLA													48	2	1	A	●	●	↕	↕	⑥	↕
	ASLB													58	2	1	B C ← b7 ← b0 ← 0	●	●	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	●	●	↕	↕	⑥	↕
	ASRA													47	2	1	A	●	●	↕	↕	⑥	↕
	ASRB													57	2	1	B b7 → b0 → C	●	●	↕	↕	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 0 → b7 → b0 → C	●	●	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;
- $\overline{M}$ 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 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 BSR or 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.

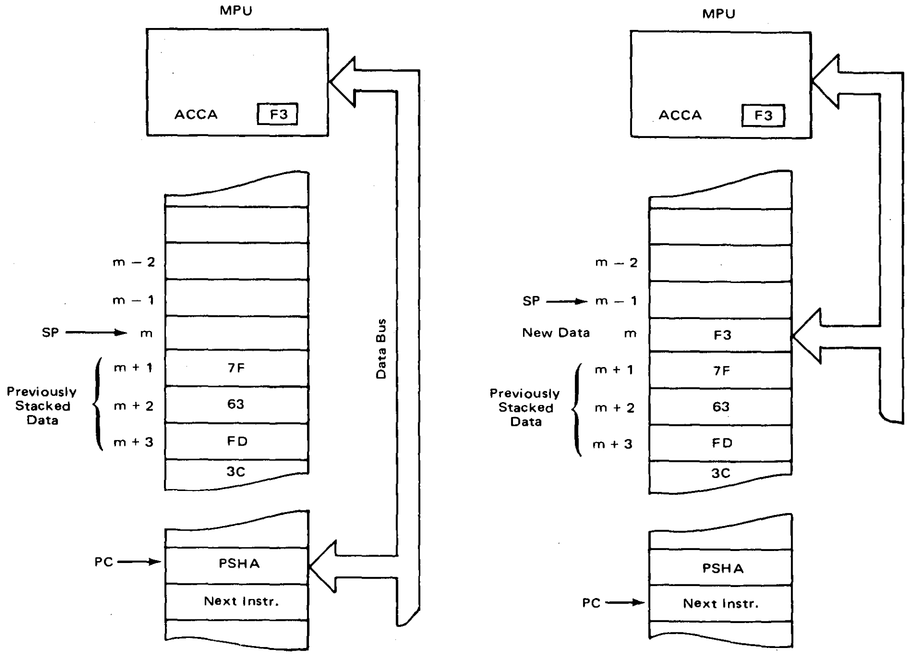
3

TABLE 3 — INDEX REGISTER AND STACK POINTER 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				$X_H - M, X_L - (M+1)$	●	●	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)

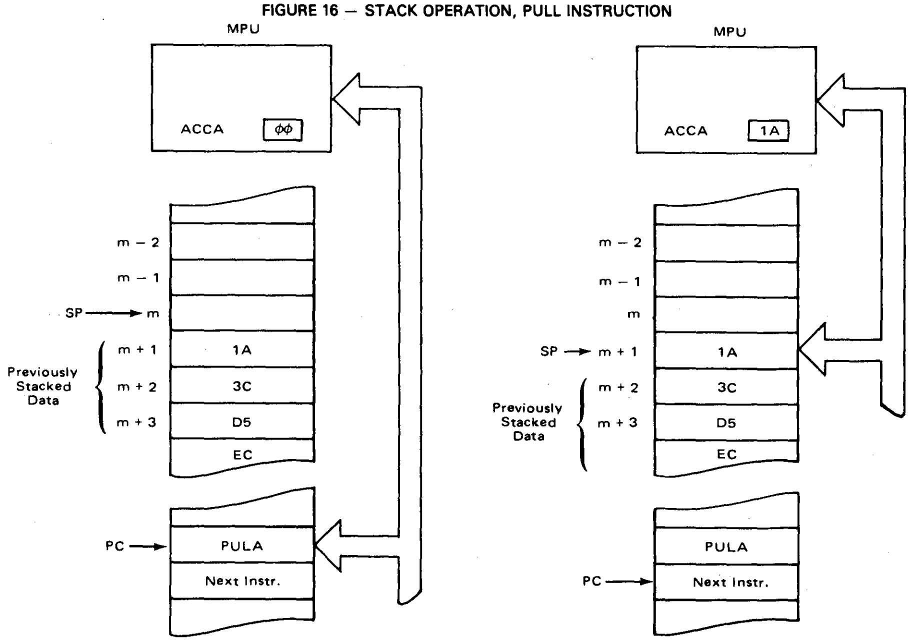
FIGURE 15 — STACK OPERATION, PUSH INSTRUCTION



(a) Before PSHA

(b) After PSHA

FIGURE 16 — STACK OPERATION, PULL INSTRUCTION



(a) Before PULA

(b) After PULA

TABLE 4 — JUMP AND BRANCH INSTRUCTIONS

OPERATIONS	MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE #	INDEX OP	INDEX ~	INDEX #	EXTND OP	EXTND ~	EXTND #	IMPLIED OP	IMPLIED ~	IMPLIED #	BRANCH TEST	COND. CODE REG. 5 H	4 I	3 N	2 Z	1 V	0 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										3B	10	1		①	①	①	①	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.

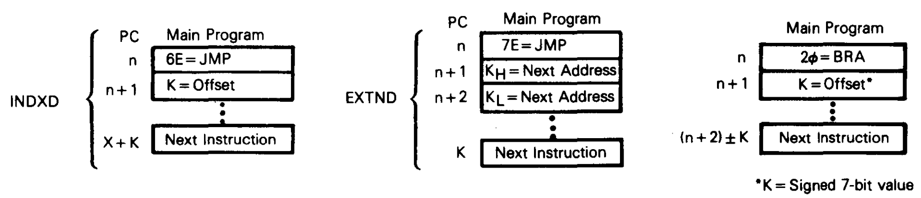
① (All) Load Condition Code Register from Stack. (See Special Operations)
② (Bit 1) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.

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 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



(a) Jump (b) Branch

FIGURE 18 — PROGRAM FLOW FOR BSR

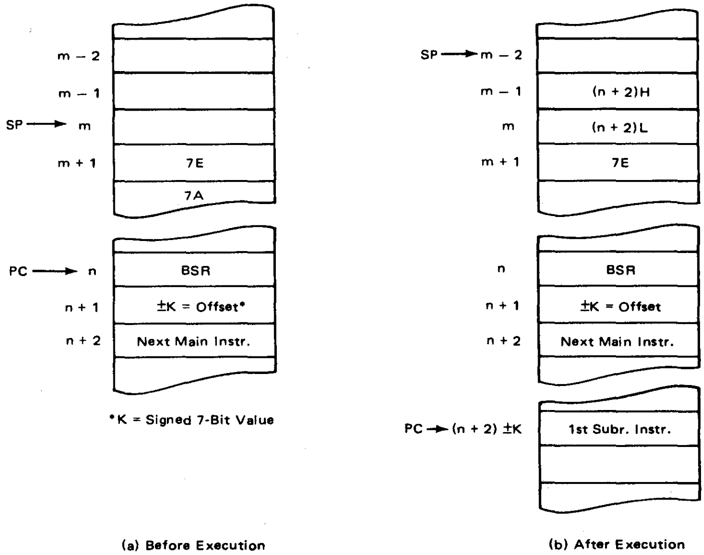


FIGURE 19 — PROGRAM FLOW FOR JSR (EXTENDED)

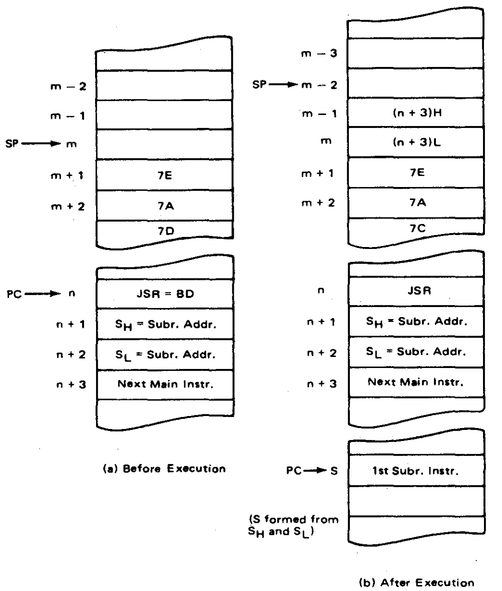


FIGURE 20 — PROGRAM FLOW FOR JSR (INDEXED)

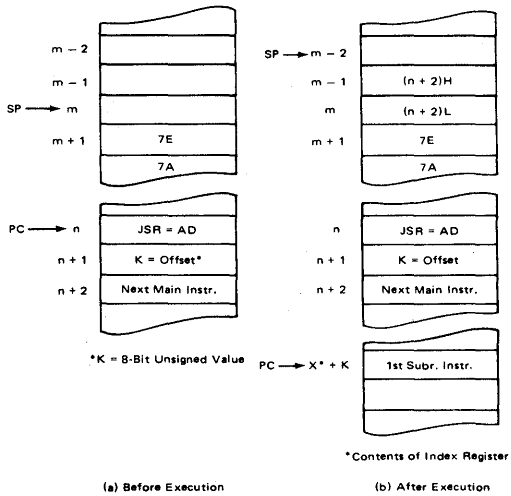


FIGURE 21 — PROGRAM FLOW FOR RTS

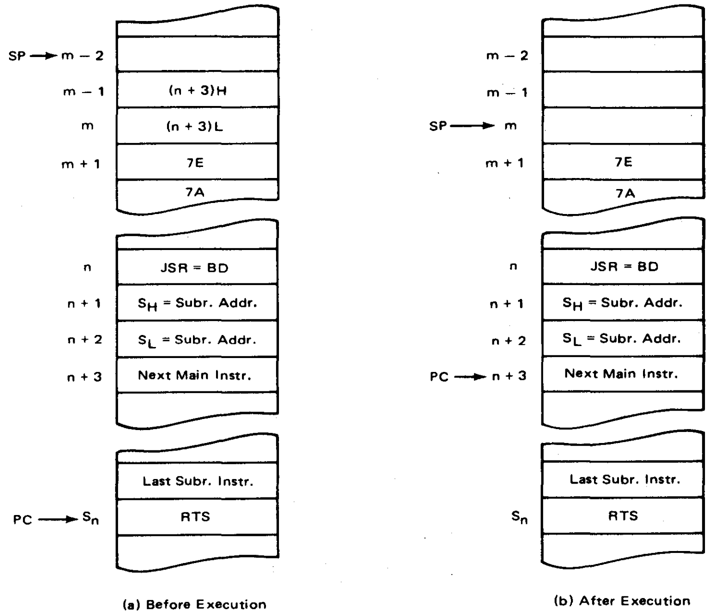


FIGURE 22 — PROGRAM FLOW FOR RTI

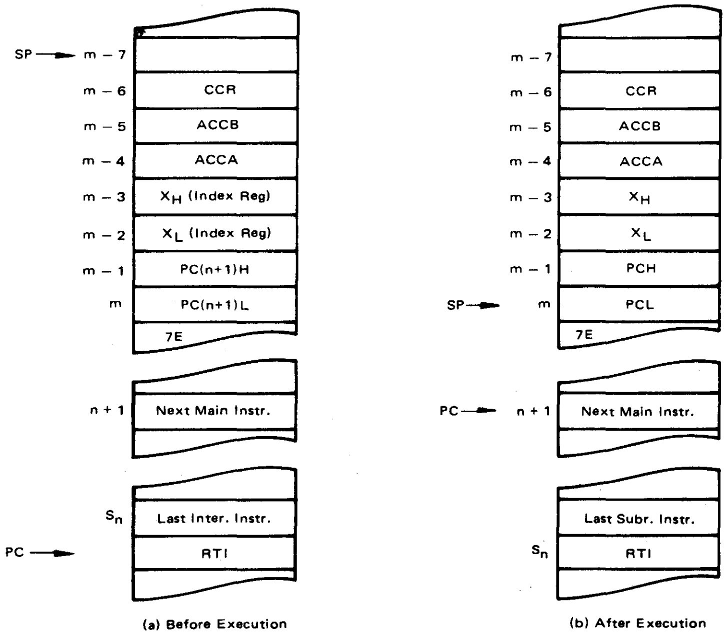


FIGURE 23 — PROGRAM FLOW FOR INTERRUPTS

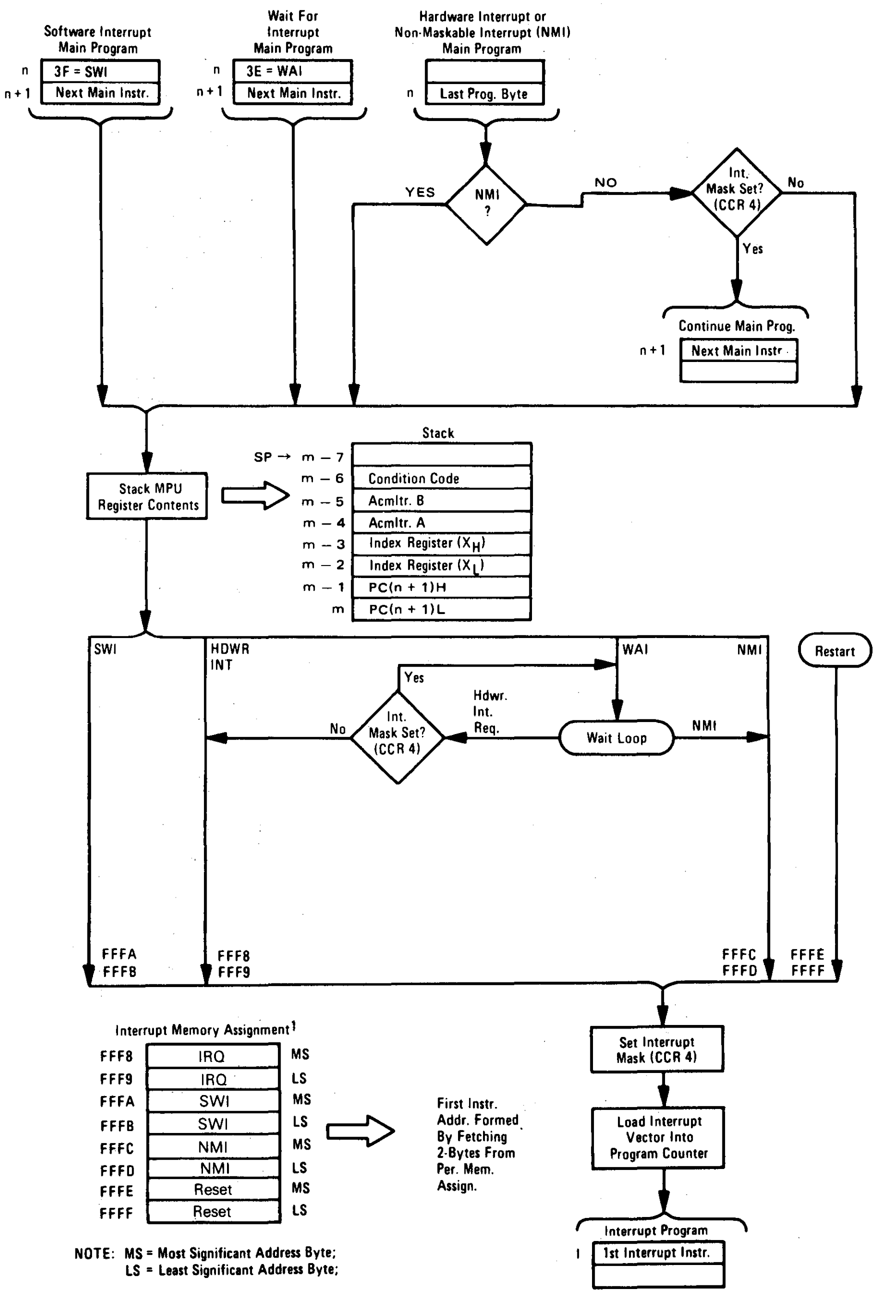


FIGURE 24 — CONDITIONAL BRANCH INSTRUCTIONS

BMI :	$N = 1$;	BEQ :	$Z = 1$;
BPL :	$N = \phi$;	BNE :	$Z = \phi$;
BVC :	$V = \phi$;	BCC :	$C = \phi$;
BVS :	$V = 1$;	BCS :	$C = 1$;
BHI :	$C + Z = \phi$;	BLT :	$N \oplus V = 1$;
BLS :	$C + Z = 1$;	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.

Systems 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

		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	(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)

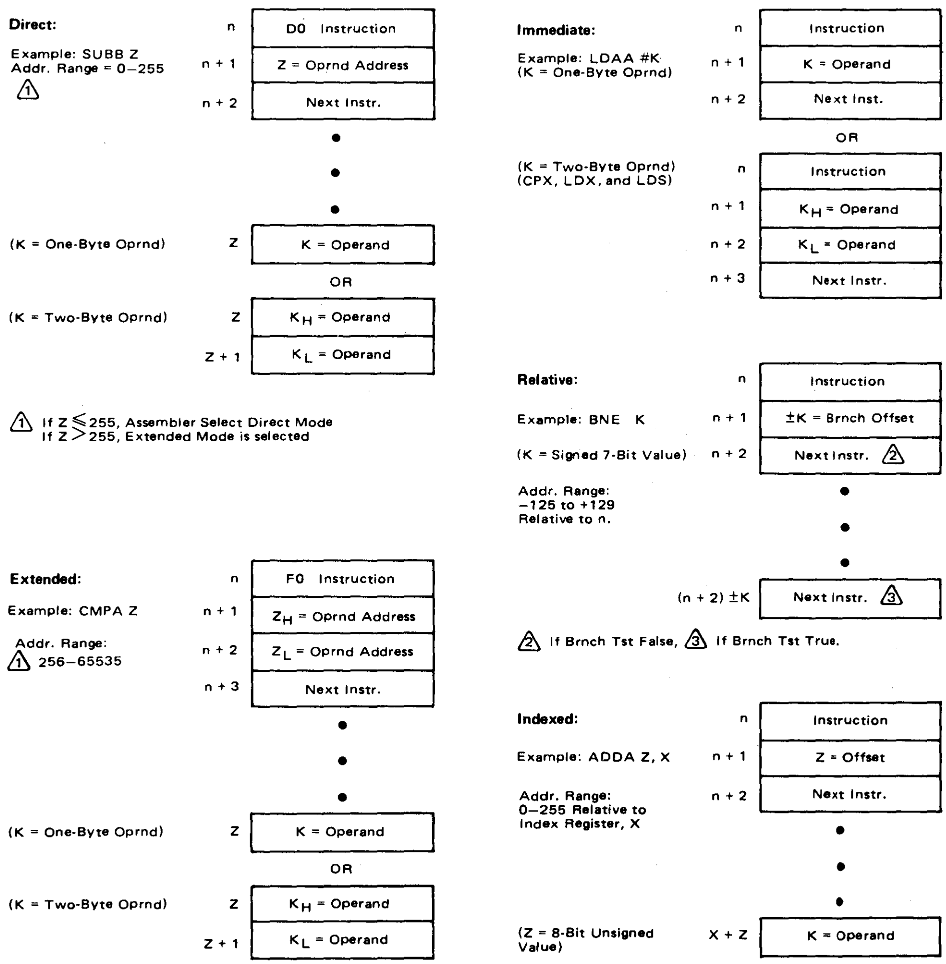
The successive fields in a statement are normally separated by one or more spaces. An exception to this rule occurs for instructions that use dual addressing in the operand field and for instructions that must distinguish between the two accumulators. In these cases, A and B are "operands" but the space 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

Operator	Comment
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

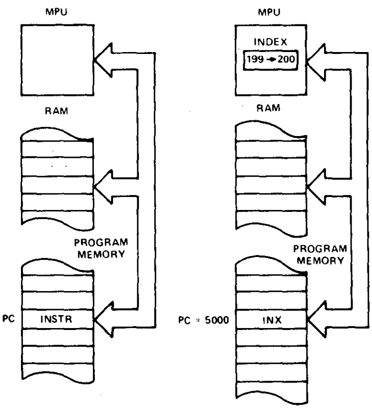
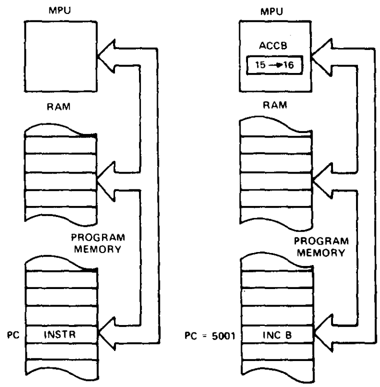


FIGURE 28 — ACCUMULATOR ADDRESSING



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

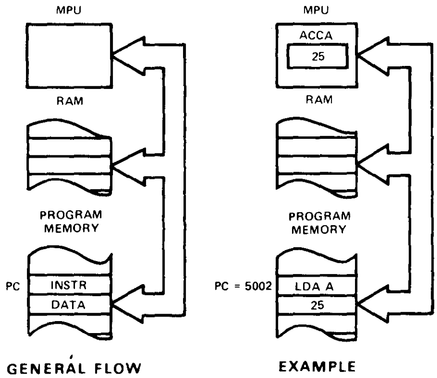


FIGURE 30 — DIRECT ADDRESSING MODE

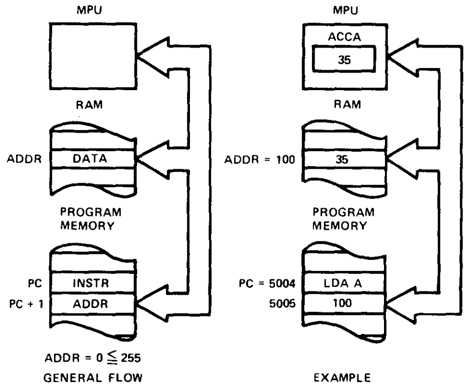


TABLE 7 — IMMEDIATE MODE CYCLE-BY-CYCLE OPERATION

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

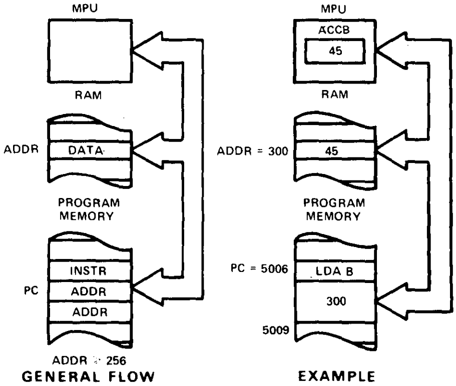


TABLE 9 — EXTENDED MODE CYCLE-BY-CYCLE

Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W̄ 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

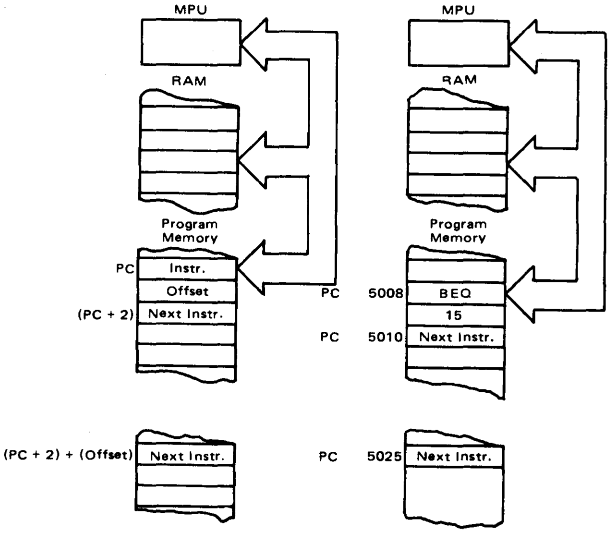


FIGURE 33 — INDEXED ADDRESSING MODE

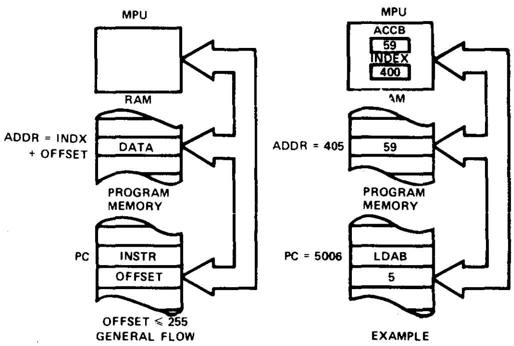


TABLE 10 — RELATIVE MODE CYCLE-BY-CYCLE OPERATION

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.

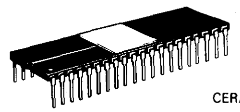
MOTOROLA

MC6802
MC6808
MC6802NS

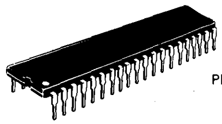
MOS

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

MICROPROCESSOR WITH CLOCK AND OPTIONAL RAM



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

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.

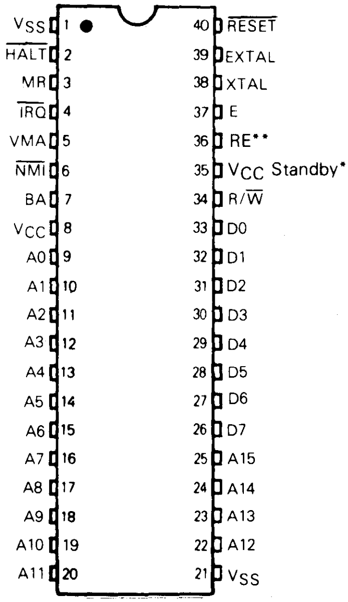
The MC6802NS is identical to the MC6802 without standby RAM feature. The MC6808 is identical to the MC6802 without on-board RAM.

- 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

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6802L
	1.0	−40°C to 85°C	MC6802CL
	1.0	0°C to 70°C	MC6802NSL
	1.0	0°C to 70°C	MC6808L
	1.5	0°C to 70°C	MC68A02L
	1.5	−40°C to 85°C	MC68A02CL
	1.5	0°C to 70°C	MC68A08L
	2.0	0°C to 70°C	MC68B02L
	2.0	0°C to 70°C	MC68B08L
Plastic P Suffix	1.0	0°C to 70°C	MC6802P
	1.0	−40°C to 85°C	MC6802CP
	1.0	0°C to 70°C	MC6802NSP
	1.0	0°C to 70°C	MC6808P
	1.5	0°C to 70°C	MC68A02P
	1.5	−40°C to 85°C	MC68A02CP
	1.5	0°C to 70°C	MC68A08P
	2.0	0°C to 70°C	MC68B02P
	2.0	0°C to 70°C	MC68B08P

PIN ASSIGNMENT

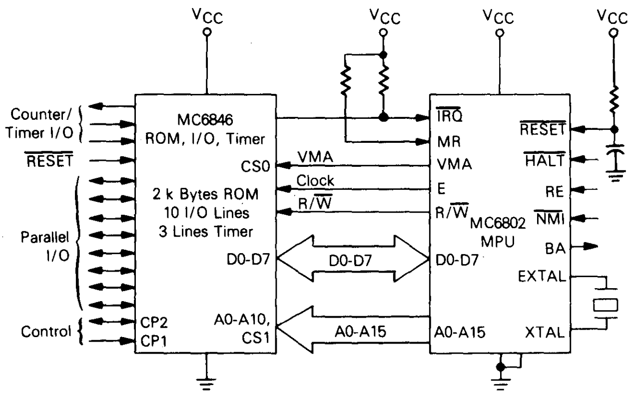


*Pin 35 must be tied to 5 V on the MC6802NS

**Pin 36 must be tied to ground for the MC6808

3

TYPICAL MICROCOMPUTER



This block diagram shows 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.

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 MC6802, MC680A02, MC680B02 MC6802C, MC680A02C MC6802NS MC6808, MC68A08, MC68B08	T_A	 0 to +70 −40 to +85 0 to +70 0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

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}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Average Thermal Resistance (Junction to Ambient) Plastic Ceramic	θ_{JA}	 100 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 = 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 $\pm5\%$, $V_{SS}=0$, $T_A=0$ to 70°C, unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXTAL	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
	$\overline{RESET}$		$V_{SS}+4.0$	–	V_{CC}	
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)	D0-D7	V_{OH}	$V_{SS}+2.4$	–	–	V
($I_{Load}=-145$ μA, $V_{CC}=$ min)	A0-A15, R/$\overline{W}$, VMA, E		$V_{SS}+2.4$	–	–	
($I_{Load}=-100$ 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 $T_A=0$°C)		P_{INT}	–	*0.750	1.0	W
V_{CC} Standby	Power Down	V_{SBB}	4.0	–	5.25	V
	Power Up	V_{SB}	4.75	–	5.25	
Standby Current		I_{SBB}	–	–	8.0	mA
Capacitance # ($V_{in}=0$, $T_A=25$°C, $f=1.0$ MHz)	D0-D7	C_{in}	–	10	12.5	pF
	Logic Inputs, EXTAL		–	6.5	10	
	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 $\pm5\%$, $V_{SS}=0$, $T_A=T_L$ to T_H, unless otherwise noted)

Characteristics	Symbol	MC6802 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

3

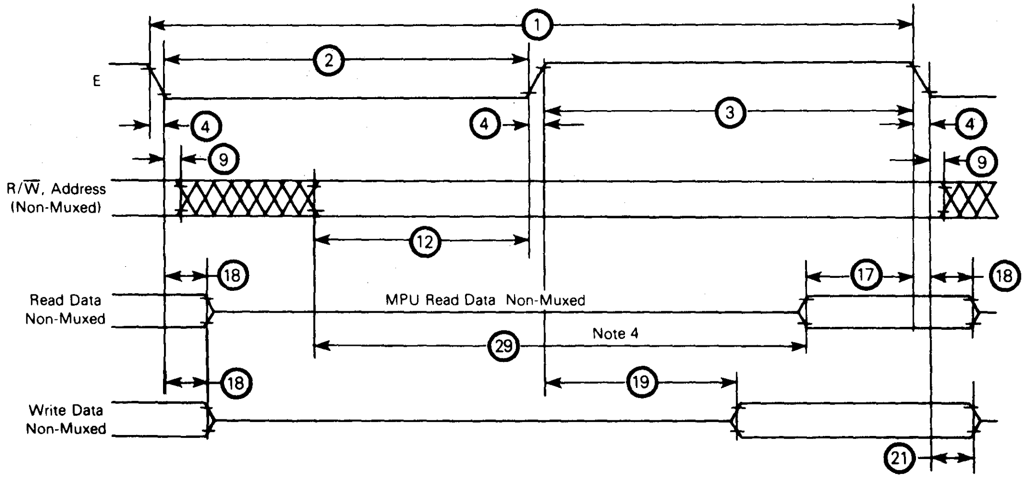
3

BUS TIMING CHARACTERISTICS

Ident. Number	Characteristic	Symbol	MC6802 MC6802NS 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	–	25	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}	535	–	335	–	235	–	ns

*Address and data hold times are periodically tested rather than 100% tested.

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.
6. All electrical and control characteristics are referenced from: $T_L = 0°C$ minimum and $T_H = 70°C$ maximum.

FIGURE 3 — BUS TIMING TEST LOAD

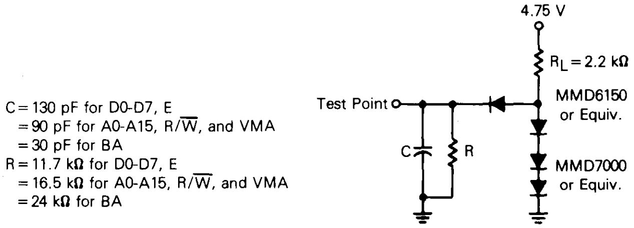


FIGURE 4 — TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

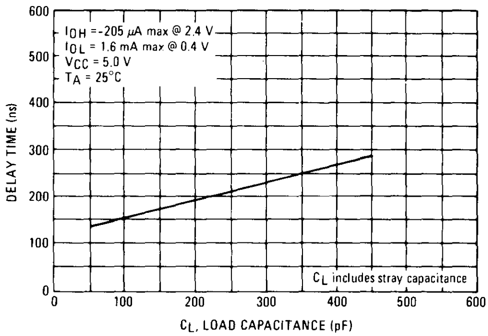


FIGURE 5 — TYPICAL READ/WRITE, VMA AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

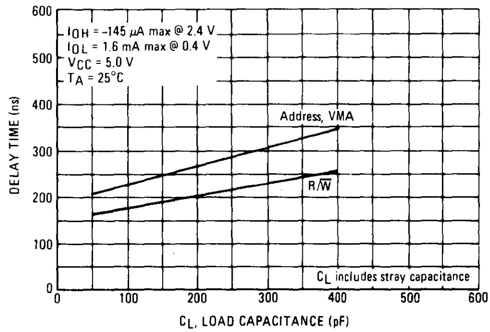
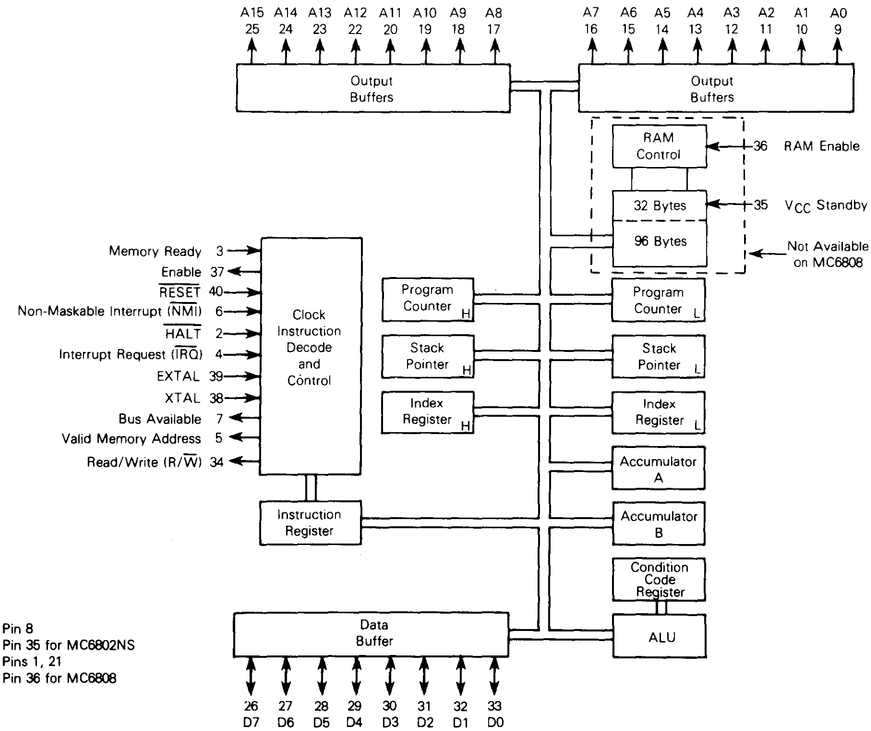


FIGURE 6 — EXPANDED BLOCK DIAGRAM



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 x 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

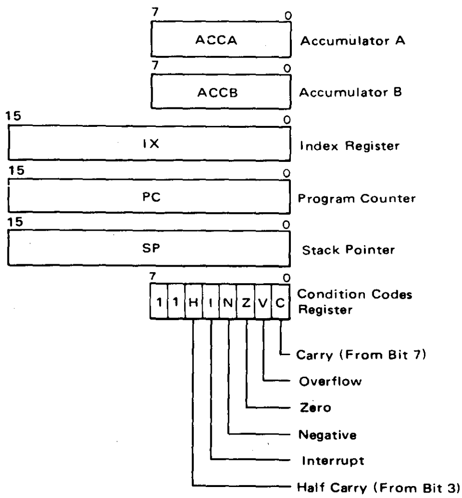
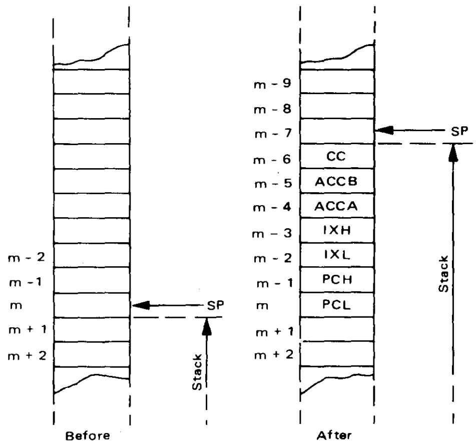


FIGURE 8 — SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK

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.

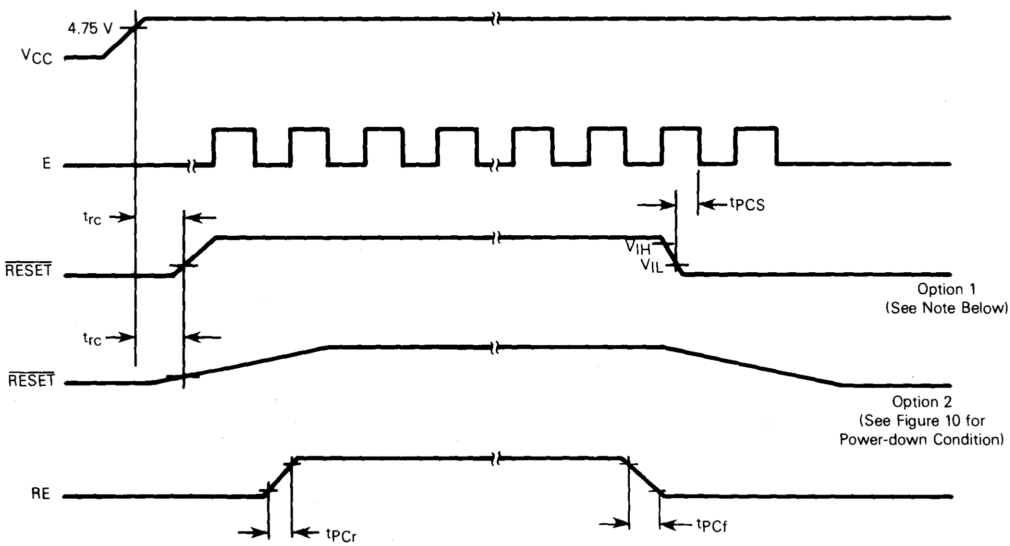
NON-MASKABLE INTERRUPT ($\overline{\text{NMI}}$)

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 $\overline{\text{IRQ}}$ and $\overline{\text{NMI}}$ 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 MS	Vector LS	Description
$FFFE	$FFFF	Restart
$FFFC	$FFFD	Non-Maskable Interrupt
$FFFA	$FFFB	Software Interrupt
$FFF8	$FFF9	Interrupt Request

FIGURE 10 — POWER-DOWN SEQUENCE

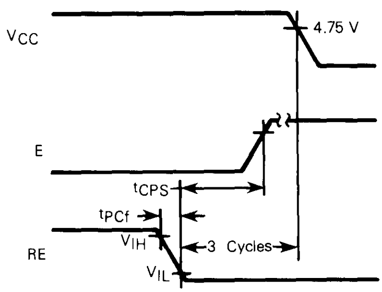


FIGURE 11 — MPU FLOWCHART

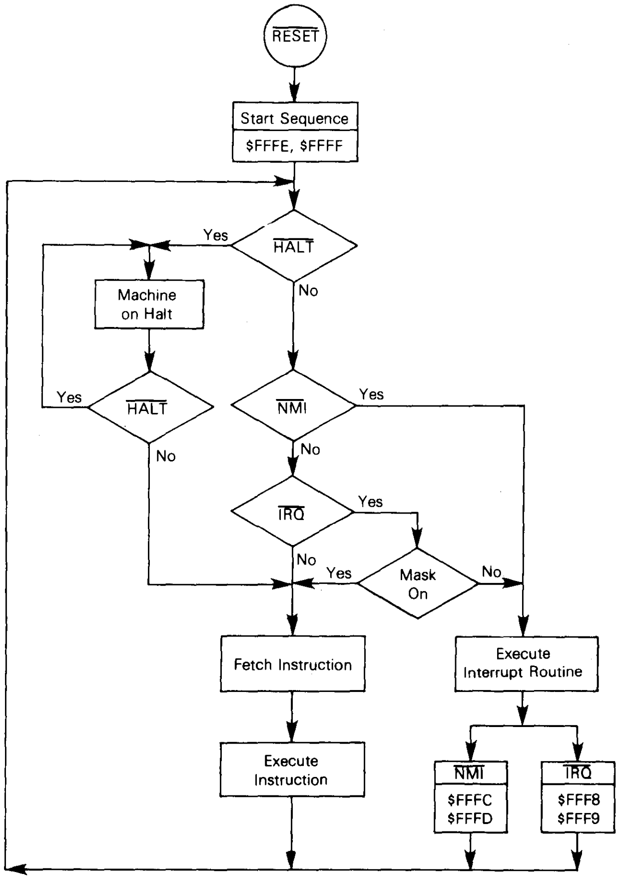
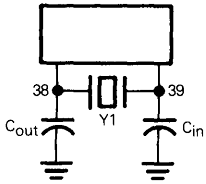
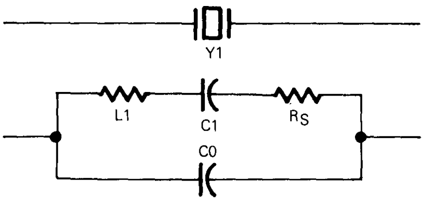


FIGURE 12 — CRYSTAL SPECIFICATIONS



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

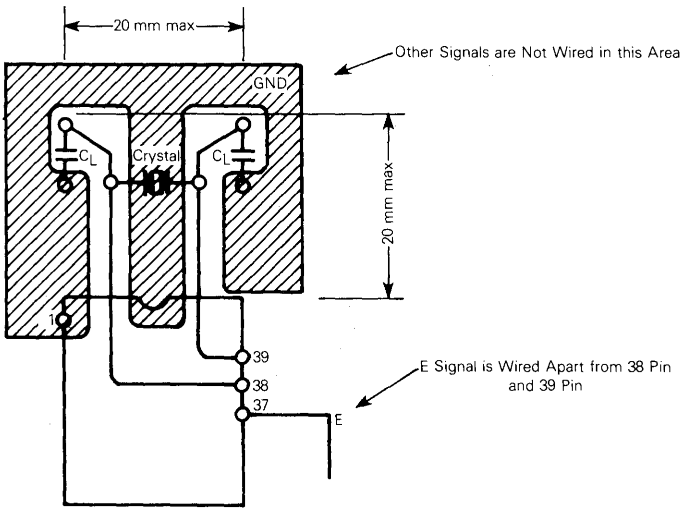


FIGURE 14 — MEMORY READY SYNCHRONIZATION

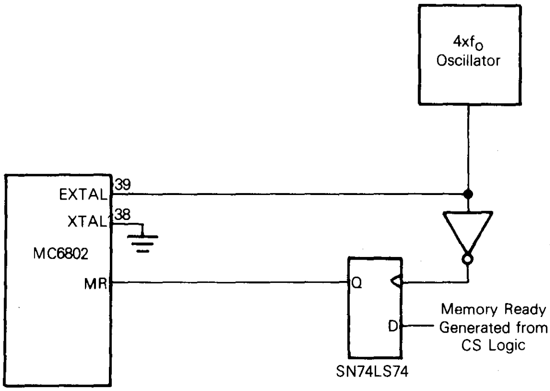
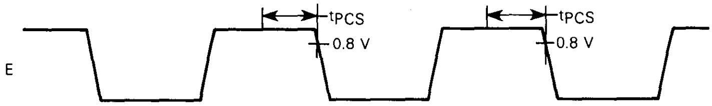


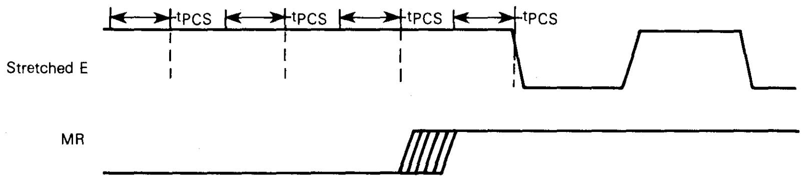
FIGURE 15 — MR NEGATIVE SETUP TIME REQUIREMENT

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 $4xf_0$ 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	CLR	Clear	PUL	Pull Data
ADC	Add with Carry	CLV	Clear Overflow	ROL	Rotate Left
ADD	Add	CMP	Compare	ROR	Rotate Right
AND	Logical And	COM	Complement	RTI	Return from Interrupt
ASL	Arithmetic Shift Left	CPX	Compare Index Register	RTS	Return from Subroutine
ASR	Arithmetic Shift Right	DAA	Decimal Adjust	SBA	Subtract Accumulators
BCC	Branch if Carry Clear	DEC	Decrement	SBC	Subtract with Carry
BCS	Branch if Carry Set	DES	Decrement Stack Pointer	SEC	Set Carry
BEQ	Branch if Equal to Zero	DEX	Decrement Index Register	SEI	Set Interrupt Mask
BGE	Branch if Greater or Equal Zero	EOR	Exclusive OR	SEV	Set Overflow
BGT	Branch if Greater than Zero	INC	Increment	STA	Store Accumulator
BHI	Branch if Higher	INS	Increment Stack Pointer	STS	Store Stack Register
BIT	Bit Test	INX	Increment Index Register	STX	Store Index Register
BLE	Branch if Less or Equal	JMP	Jump	SUB	Subtract
BLS	Branch if Lower or Same	JSR	Jump to Subroutine	SWI	Software Interrupt
BLT	Branch if Less than Zero	LDA	Load Accumulator	TAB	Transfer Accumulators
BMI	Branch if Minus	LDS	Load Stack Pointer	TAP	Transfer Accumulators to Condition Code Reg.
BNE	Branch if Not Equal to Zero	LDX	Load Index Register	TBA	Transfer Accumulators
BPL	Branch if Plus	LSR	Logical Shift Right	TPA	Transfer Condition Code Reg. to Accumulator
BRA	Branch Always	NEG	Negate	TST	Test
BSR	Branch to Subroutine	NOP	No Operation	TSX	Transfer Stack Pointer to Index Register
BVC	Branch if Overflow Clear	ORA	Inclusive OR Accumulator	TXS	Transfer Index Register to Stack Pointer
BVS	Branch if Overflow Set	PSH	Push Data	WAI	Wait for Interrupt
CBA	Compare Accumulators				
CLC	Clear Carry				
CLI	Clear Interrupt Mask				

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				$\overline{M}$ → 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 C b7 b0	•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	⑥	↕
	RORA													46	2	1	A	•	•	↕	↕	⑥	↕
	RORB													56	2	1	B C b7 b0	•	•	↕	↕	⑥	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A 0	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B C b7 b0	•	•	↕	↕	⑥	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B b7 b0 C	•	•	↕	↕	⑥	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A 0	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B b7 b0 C	•	•	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;
- $\overline{M}$ Complement of M;
- → Transfer Into;
- 0 Bit = Zero;
- 00 Byte = Zero;

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	$X - 1 \rightarrow X$	•	•	•	↕	•	•
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				$M \rightarrow X_H, (M+1) \rightarrow X_L$	•	•	9	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H, (M+1) \rightarrow SP_L$	•	•	9	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M, X_L \rightarrow (M+1)$	•	•	9	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M, SP_L \rightarrow (M+1)$	•	•	9	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \rightarrow SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \rightarrow X$	•	•	•	•	•	•

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										$N \oplus V = 0$	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										$Z + (N \oplus V) = 0$	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										$Z + (N \oplus V) = 1$	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										$N \oplus V = 1$	•	•	•	•	•	•
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

SPECIAL OPERATIONS

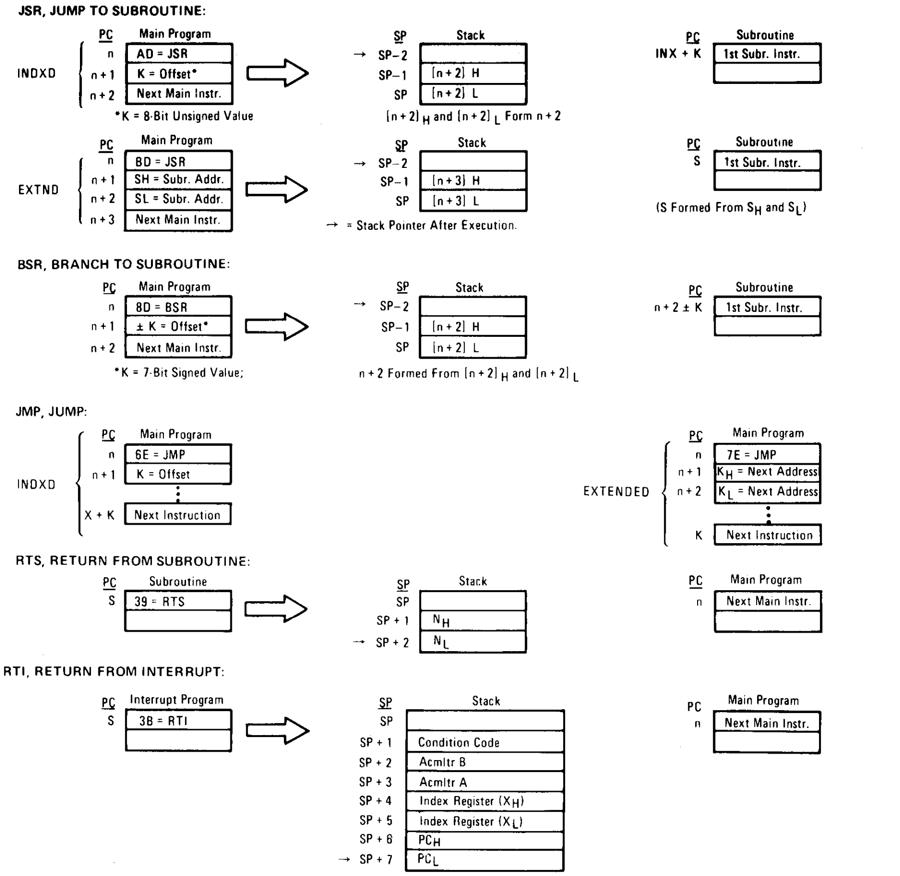


TABLE 6 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

		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.

3

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)

3

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

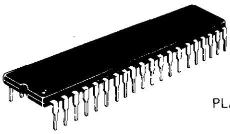
Advance Information

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

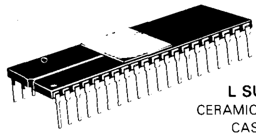
8-BIT MICROPROCESSOR

8-BIT MICROPROCESSOR

The MC6803E is an 8-bit microprocessing unit (MPU) designed for uses in which the internal clock needs to be synchronized with systems, peripherals, or other MPUs. The MC6803E also supports DMA and dynamic RAM refresh with its halt ($\overline{\text{HALT}}$) 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.



P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

- Enhanced MC6800 Instruction Set
- Upward Source and Object Code Compatible with the MC6800
- Bus Compatible with the M6800 Family
- Direct Source and Object Code Compatible with the MC6801
- 8 × 8 Multiply Instruction
- 64K Memory Map (Unused High Order Address Lines Can Be Used as Input Lines)
- External Clock Inputs (E and AS) Allow Synchronization
- DMA Capability (Clock Stretching) with $\overline{\text{HALT}}$ 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

PIN ASSIGNMENT

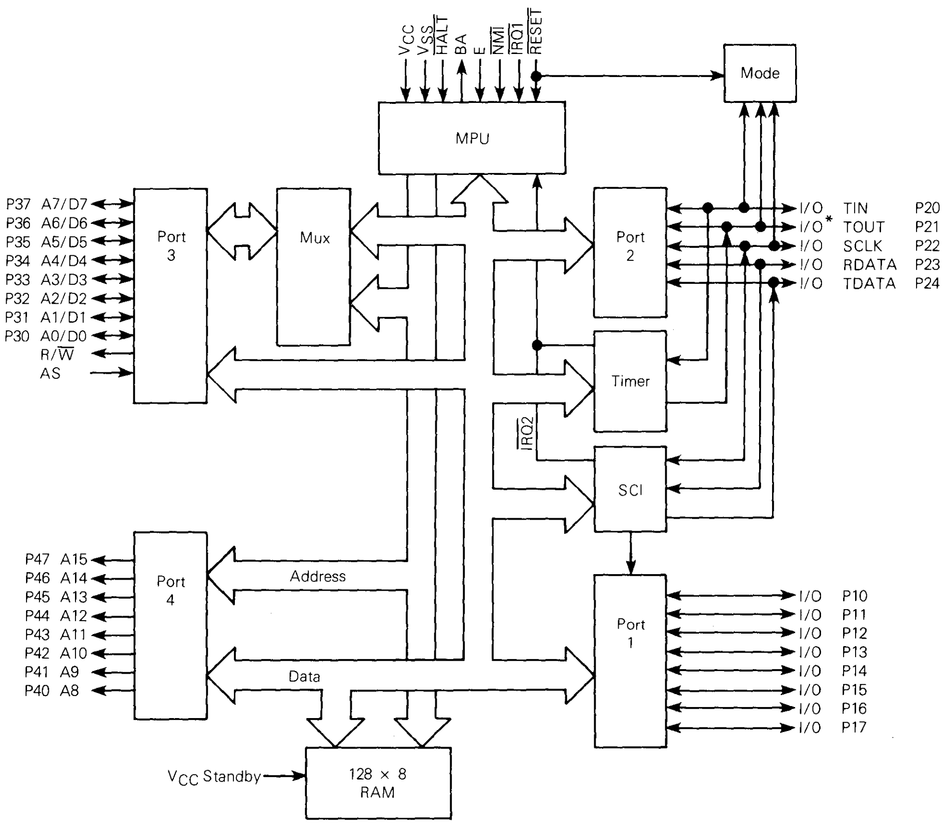
Signal	Pin	Pin	Signal
V_{SS}	1	40	E
$\overline{\text{HALT}}$	2	39	AS
BA	3	38	R/$\overline{\text{W}}$
$\overline{\text{NMI}}$	4	37	P30
$\overline{\text{IRQ1}}$	5	36	P31
$\overline{\text{RESET}}$	6	35	P32
V_{CC}	7	34	P33
P20	8	33	P34
P21	9	32	P35
P22	10	31	P36
P23	11	30	P37
P24	12	29	P40
P10	13	28	P41
P11	14	27	P42
P12	15	26	P43
P13	16	25	P44
P14	17	24	P45
P15	18	23	P46
P16	19	22	P47
P17	20	21	V_{CC} Standby

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

Package Type	Frequency	Order Number
Plastic P Suffix	1.0 MHz 1.25 MHz	MC6803EP MC6803EP-1
Ceramic L Suffix	1.0 MHz 1.25 MHz	MC6803EL MC6803EL-1

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

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:

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

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 = 0°C to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	E	V_{EIH}	$V_{CC}-0.75$	–	V_{CC}	V
Input Low Voltage	E	V_{EIL}	$V_{SS}-0.3$	–	$V_{SS}+0.6$	V
Input High Voltage	$\overline{RESET}$	V_{IH}	$V_{SS}+4.0$	–	V_{CC}	V
	Other Inputs*		$V_{SS}+2.0$	–	V_{CC}	
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	$\overline{HALT}$, AS, $\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	1.5	2.5	μA
Hi-Z Input Current (V_{in} = 0.5 to 2.4 V)	P10-P17, P20-P24, P30-P37	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	C_{in}	–	–	12.5	pF
	Other Inputs (Except E)		–	–	10.0	
V_{CC} Standby	Power Down	V_{SBB}	4.0	–	5.25	V
	Power Up	V_{SB}	4.75	–	5.25	
Standby Current	Power Down	I_{SBB}	–	–	6.0	mA

*Except mode programming levels; see Figure 8.

PERIPHERAL PORT TIMING (Refer to Figures 1 and 2)

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

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

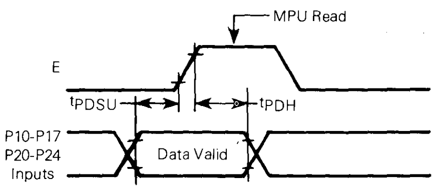
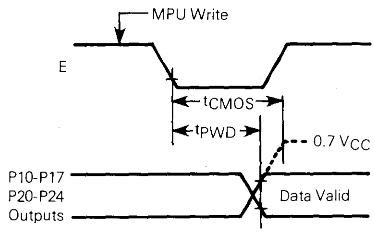


FIGURE 2 — DATA SETUP AND HOLD TIMES (MPU WRITE)



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

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

FIGURE 3 — CMOS LOAD

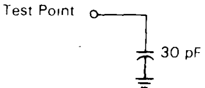
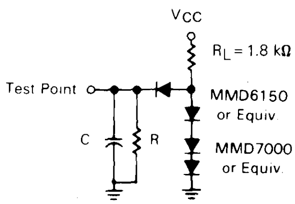


FIGURE 4 — TIMING TEST LOAD PORTS 1, 2, 3, 4

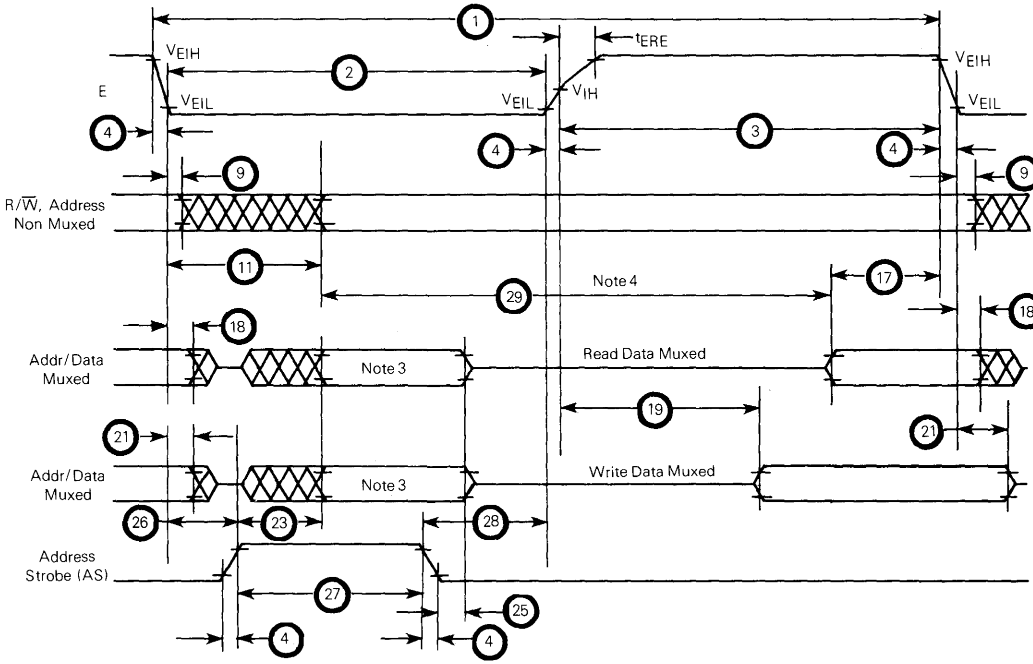


C = 90 pF for P30-P37, P40-P47, R/$\overline{W}$
= 30 pF for P10-P17, P20-P24, BA
R = 24 kΩ for P10-P17, P20-P24, P30-P37, P40-P47, R/$\overline{W}$, BA

BUS TIMING (See Notes 1 and 2)

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

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

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

FIGURE 6 — PROGRAMMING MODEL

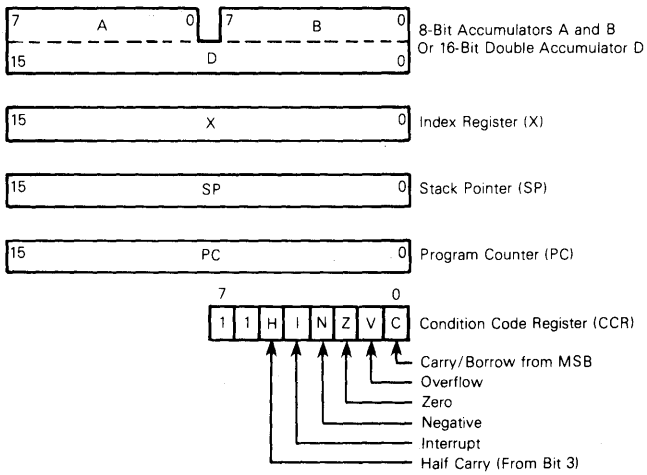


TABLE 1 — NEW INSTRUCTIONS

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

OPERATING MODES

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

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

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

Figure 8 depicts a typical operating configuration.

PROGRAMMING THE MODE

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

PORT 2 DATA REGISTER

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

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

FIGURE 7 — TYPICAL LATCH ARRANGEMENT

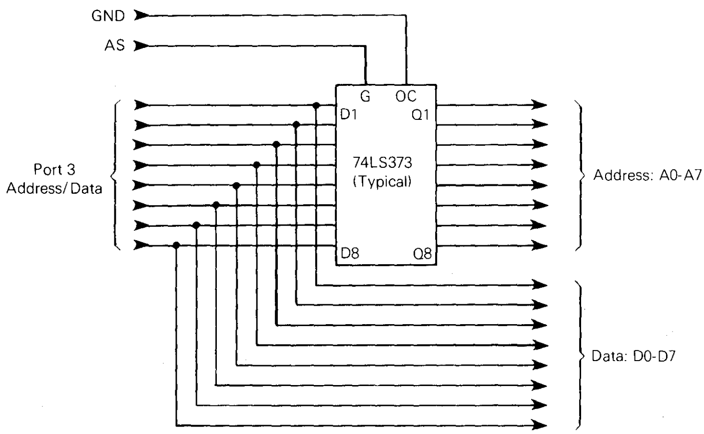
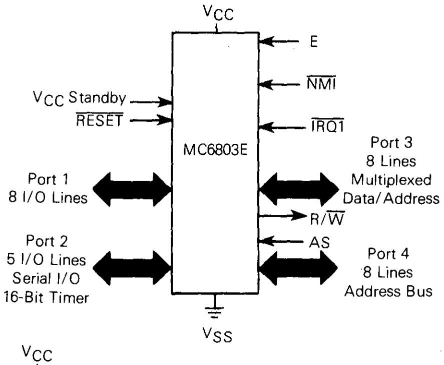
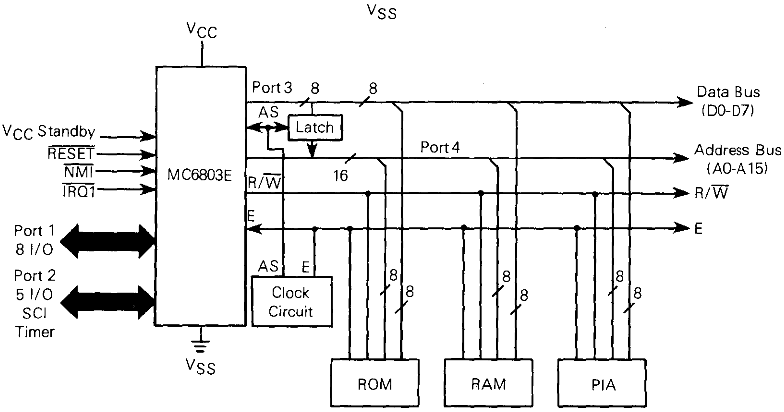


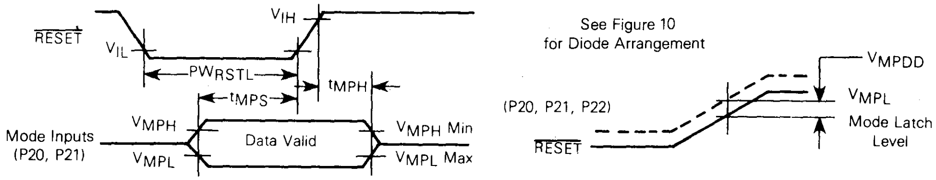
FIGURE 8 — EXPANDED MULTIPLEXED CONFIGURATION





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

FIGURE 9 — MODE PROGRAMMING TIMING



MODE PROGRAMMING (Refer to Figure 9)

Characteristic	Symbol	Min	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	V
Mode Programming Diode Differential (If Diodes are Used)	V_{MPDD}	0.6	–	V
$\overline{\text{RESET}}$ 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 $\geq 1\ \mu s$ $\overline{\text{RESET}}$ Rise Time $< 1\ \mu s$	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

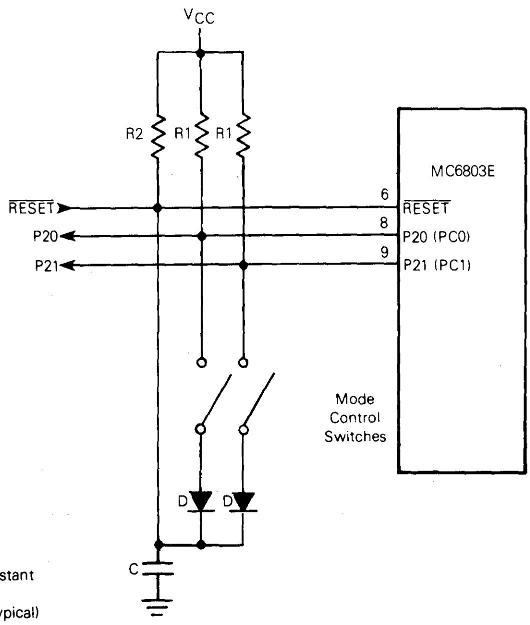
TABLE 3 — MODE SELECTION SUMMARY

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

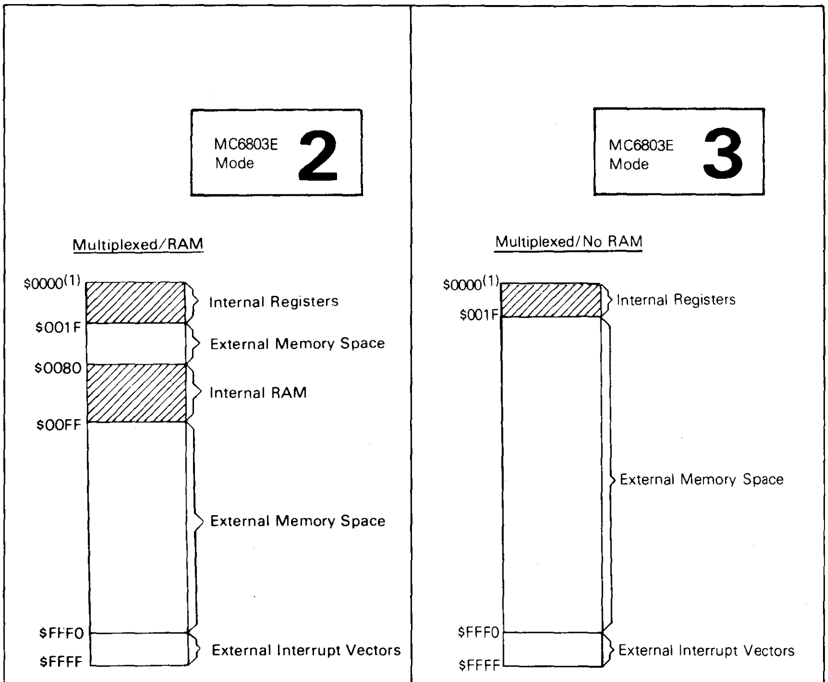


NOTES:
1. Mode 3 as shown
2. R2•C = reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)
5. Diode V_f should not exceed V_{MPDD} min.

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

3

MC6803E INTERRUPTS

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

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

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

TABLE 5 — MCU INTERRUPT VECTOR LOCATIONS

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

*$\overline{IRQ2}$ Interrupt

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

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

V_{CC} STANDBY

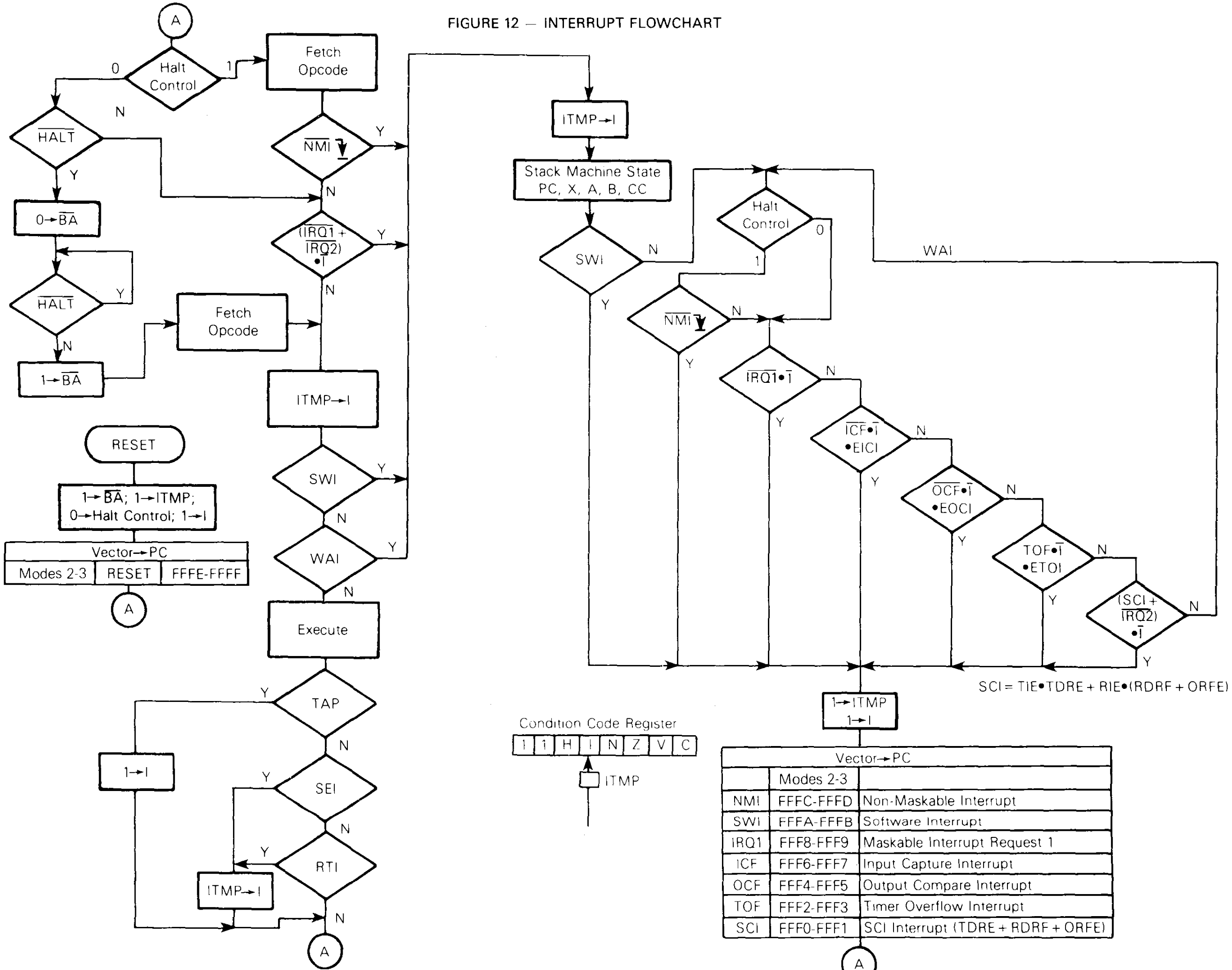
V_{CC} standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM control register. Voltage requirements depend on whether the device is in a power-up or power-down state. In the power-up state, the power supply should provide +5 volts (± 5%)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

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

AS (ADDRESS STROBE)

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

FIGURE 12 — INTERRUPT FLOWCHART



Vector→PC		
Modes 2-3	RESET	FFFE-FFFF

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

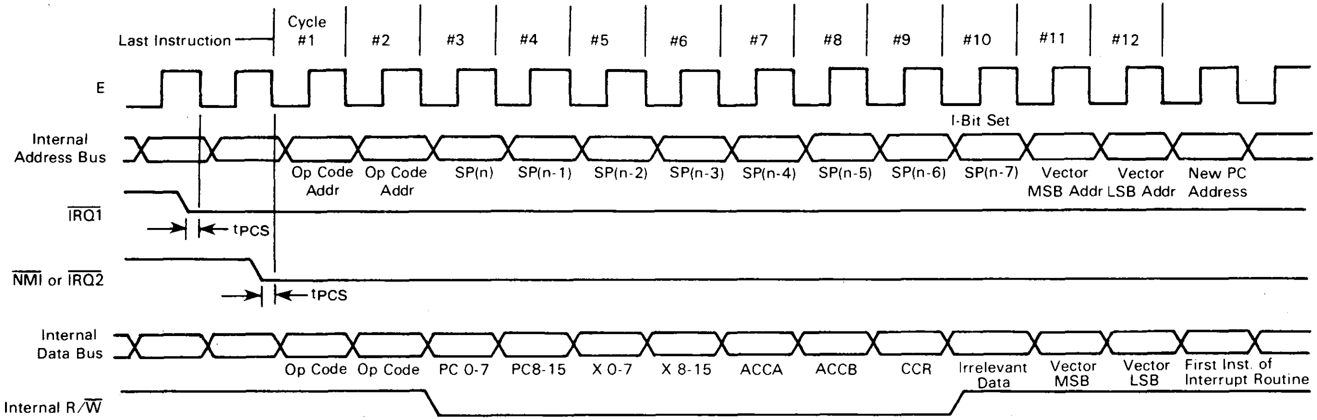
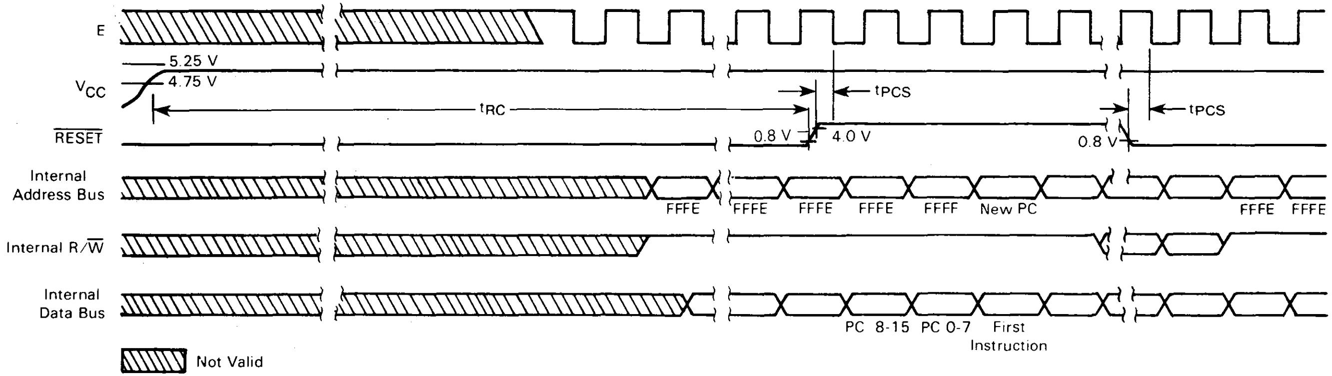


FIGURE 14 — RESET TIMING



$\overline{\text{HALT}}$

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.

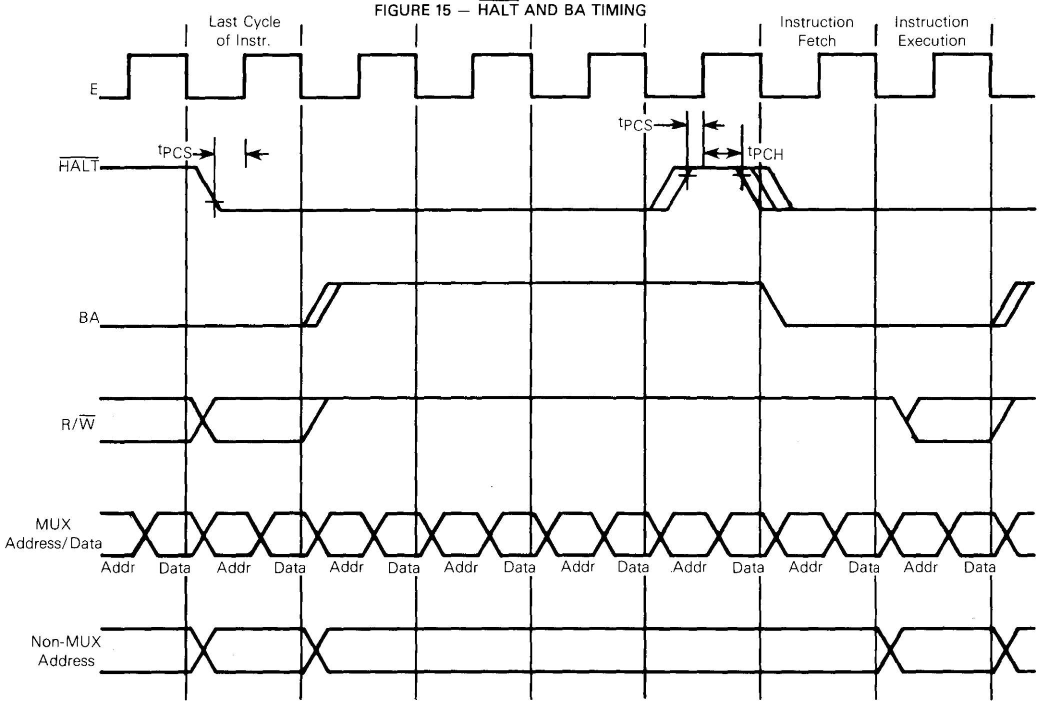
BA (BUS AVAILABLE)

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. 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. Note that the BA line will also go high when a wait instruction is executed.

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.

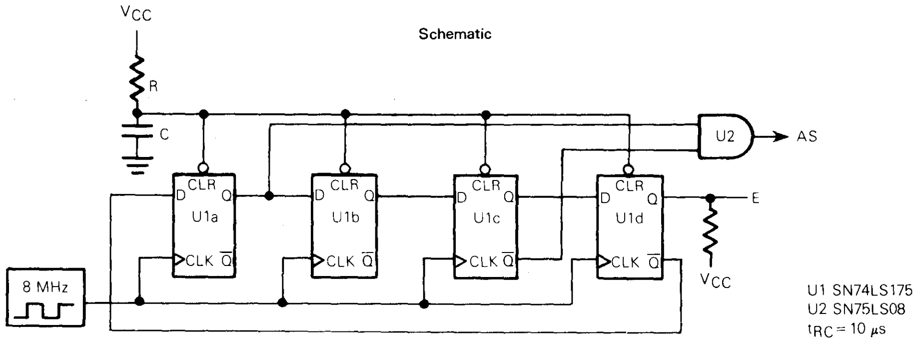
E (ENABLE)

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.

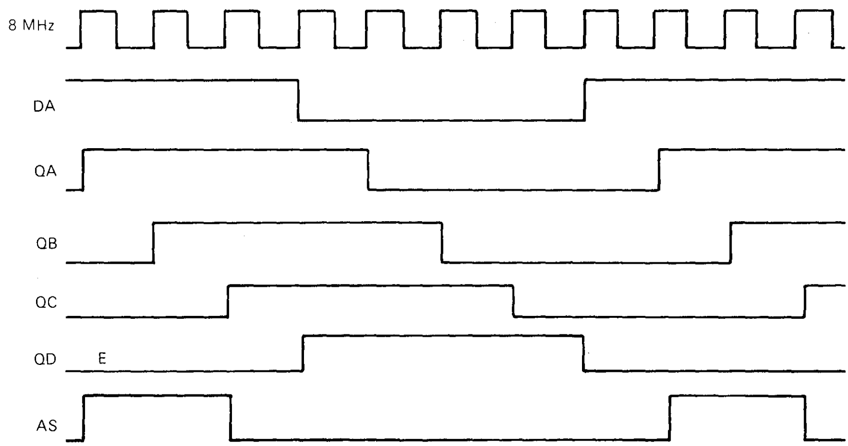
Enable is the primary MC6803E system timing signal and all timing data specified as cycles is assumed to be referenced to this clock unless otherwise noted.

FIGURE 16 — CLOCK CIRCUIT EXAMPLE 1

Schematic



Timing



3

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

An $\overline{NMI}$ 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. $\overline{NMI}$ typically requires a 3.3 kilohm (nominal) resistor to V_{CC}. There is no internal $\overline{NMI}$ pullup resistor. $\overline{NMI}$ must be held low for at least one E cycle to be recognized under all conditions.

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

$\overline{IRQ1}$ is a level-sensitive input which can be used to request an interrupt sequence. The MPU will complete the current instruction before it responds to the request. If the 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.

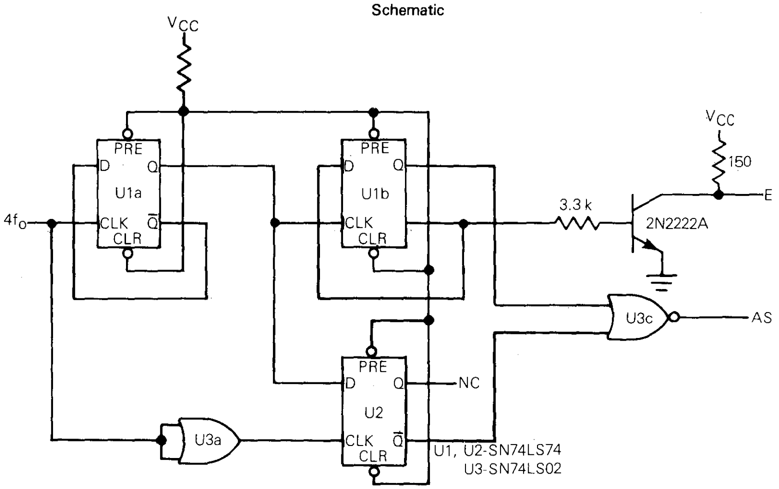
$\overline{IRQ1}$ typically requires an external 3.3 kilohm (nominal) resistor to V_{CC} for wire-OR applications. $\overline{IRQ1}$ has no internal pullup resistors.

P10-P17 (PORT 1)

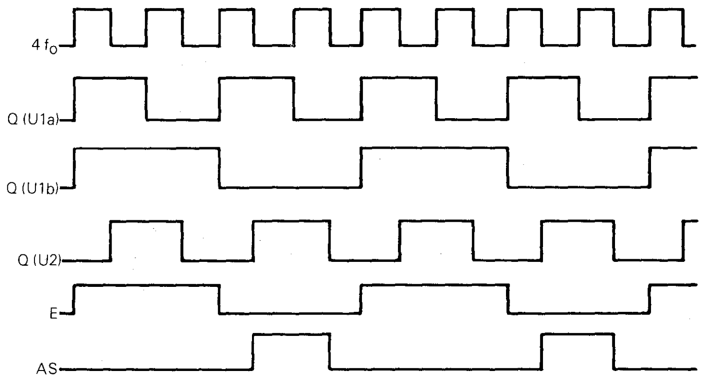
Port 1 is a mode independent 8-bit I/O port with each line an input or output as defined by the port 1 data direction register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 picofarads, Darlington transistors, or CMOS devices using external pullup resistors. It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

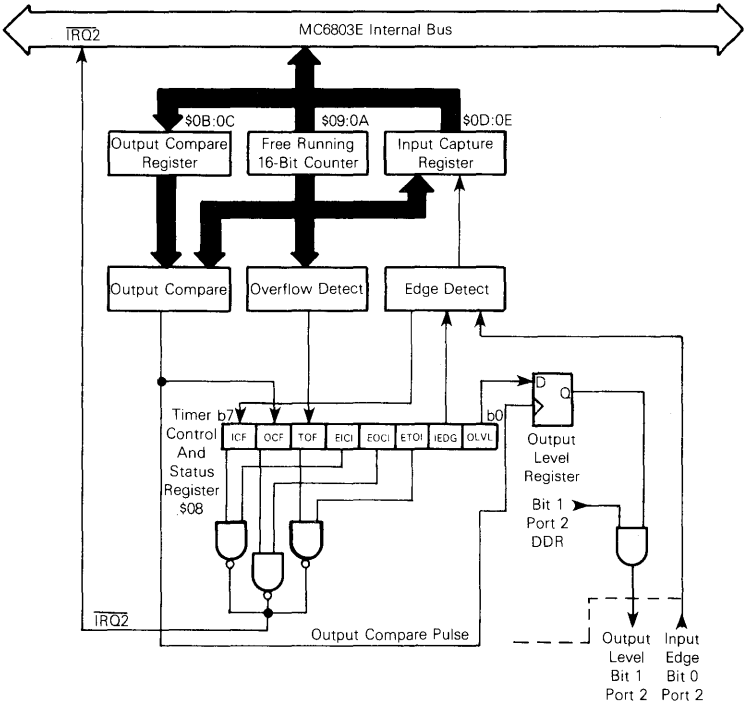
OUTPUT COMPARE REGISTER (\$0B:0C)

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 $\overline{RESET}$.

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 $\overline{\text{IRQ2}}$ 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	$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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ2}}$ 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 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, 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:

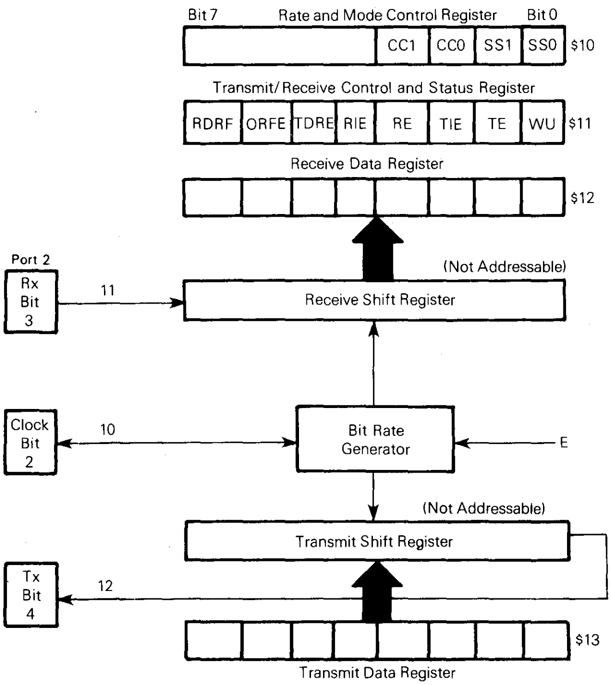
- format : standard mark/space (NRZ) or bi-phase
- clock: external or internal bit rate clock
- baud: one of 4 per E clock frequency, or external clock (8x desired baud)
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal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

3

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

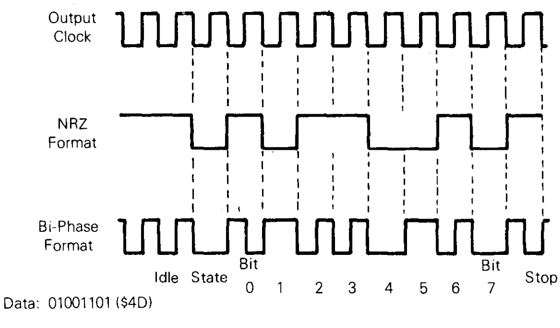


TABLE 8 – CPU INSTRUCTION MAP

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	•					*UNDEFINED OP CODE			
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	•								

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.

3

TABLE 9 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

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

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

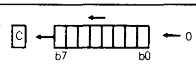
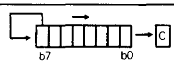
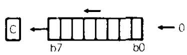
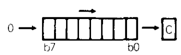
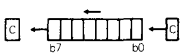
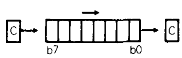
Accumulator and		Immed			Direct			Index			Extend			Inher			Boolean	Condition Codes					
																		5	4	3	2	1	0
Memory Operations	MNEM	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Expression	H	I	N	Z	V	C
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B +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

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

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

The condition code register notes are listed after Table 12.

TABLE 11 — JUMP AND BRANCH INSTRUCTIONS

Operations	MNEM	Direct Op	Direct ~	Direct #	Relative Op	Relative ~	Relative #	Index Op	Index ~	Index #	Extend Op	Extend ~	Extend #	Inherent Op	Inherent ~	Inherent #	Branch Test	5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C=0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C=1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z=1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										$N \oplus V=0$	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										$Z+(N \oplus V)=0$	•	•	•	•	•	•
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										$Z+(N \oplus V)=1$	•	•	•	•	•	•
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										$N \oplus V=1$	•	•	•	•	•	•
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	See Special Operations - Figure 21	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

TABLE 12 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

LEGEND

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

CONDITION CODE SYMBOLS

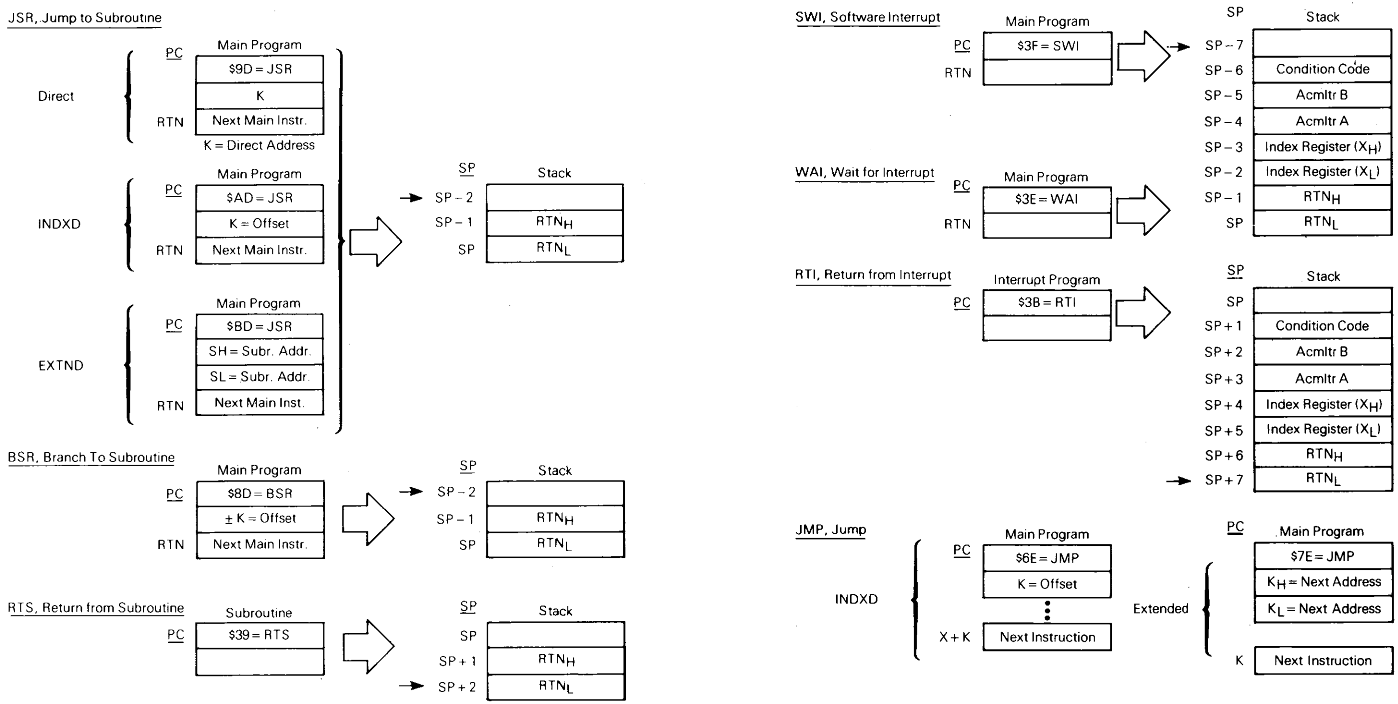
- H Half-carry from bit 3
- I Interrupt mask
- N Negative (sign bit)
- Z Zero (byte)
- V Overflow, 2's complement
- C Carry/Borrow from MSB
- R Reset Always
- S Set Always
- ↕ Affected
- • Not Affected

TABLE 13 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

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

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

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

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

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

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

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

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

3

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

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

3

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

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

MOTOROLA

MC6809

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 = 4 × E)
- $\overline{\text{DMA/BREQ}}$ 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

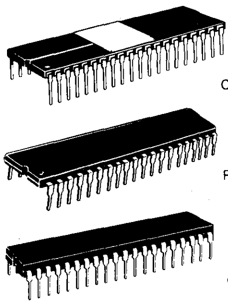
SOFTWARE FEATURES

- 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

HMOS

(HIGH DENSITY N-CHANNEL, SILICON-GATE)

8-BIT MICROPROCESSING UNIT



L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	$\overline{\text{HALT}}$
$\overline{\text{NMI}}$	2	39	XTAL
$\overline{\text{IRQ}}$	3	38	EXTAL
$\overline{\text{FIRQ}}$	4	37	$\overline{\text{RESET}}$
BS	5	36	MRDY
BA	6	35	Q
V_{CC}	7	34	E
A0	8	33	$\overline{\text{DMA/BREQ}}$
A1	9	32	R/$\overline{\text{W}}$
A2	10	31	D0
A3	11	30	D1
A4	12	29	D2
A5	13	28	D3
A6	14	27	D4
A7	15	26	D5
A8	16	25	D6
A9	17	24	D7
A10	18	23	A15
A11	19	22	A14
A12	20	21	A13

3

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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Cerdip Plastic	θ_{JA}	 50 60 100	°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 levels (e.g., either V_{SS} or V_{CC}).

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

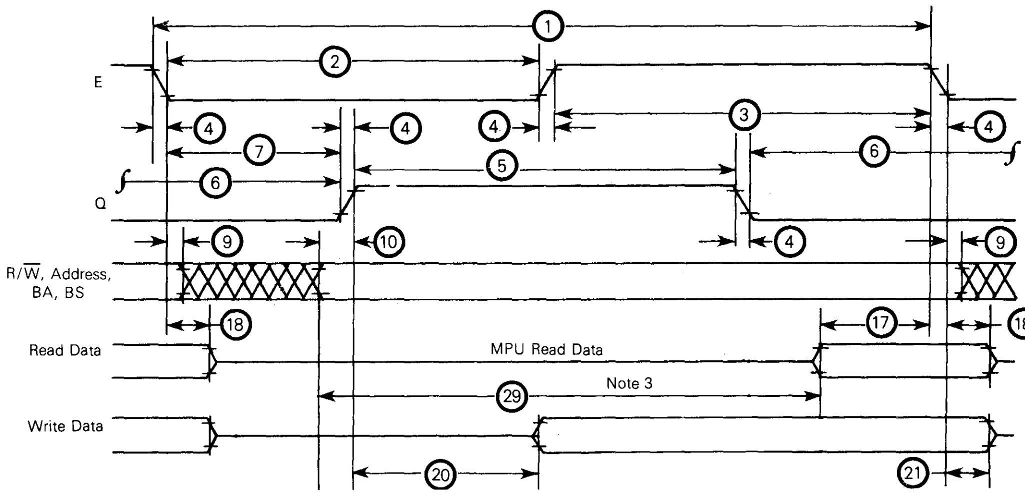
Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

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, 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/$\overline{W}$, 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
Hi-Z (Off State) Input Current (V_{in}=0.4 to 2.4 V, V_{CC}=max)	D0-D7 A0-A15, R/$\overline{W}$	I_{TSI}	– –	2.0 –	10 100	μA

*Capacitances are periodically tested rather than 100% tested.

FIGURE 1 — BUS TIMING



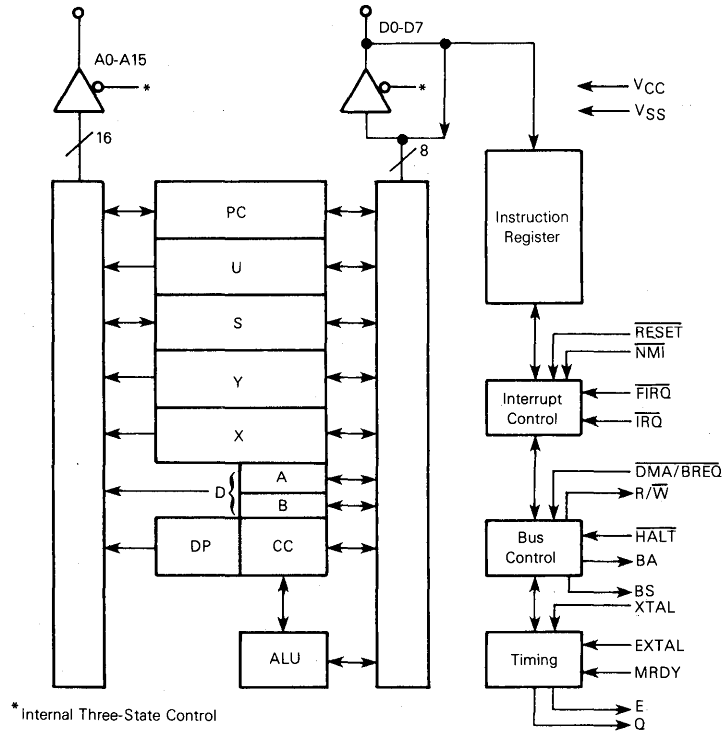
BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6809 Min	MC6809 Max	MC68A09 Min	MC68A09 Max	MC68B09 Min	MC68B09 Max	Unit
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, $\overline{DMA}/\overline{BREQ}$, $\overline{HALT}$, $\overline{RESET}$) (Figures 6, 8, 9, 10, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	Crystal Oscillator Start Time (Figures 6 and 7)	t_{RC}	–	100	–	100	–	100	ms
	Processor Control Rise and Fall Time (Figures 6 and 8)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

*Address and data hold times are periodically tested rather than 100% tested.

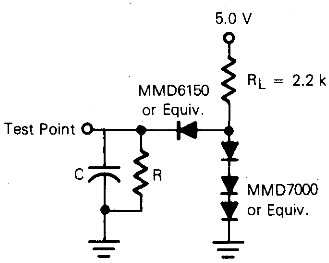
NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Usable access time is computed by: 1 – 4 – 7 max + 10 – 17.
4. Hold time (9) for BA and BS is not specified.
5. Maximum t_{cyc} during MRDY or $\overline{DMA}/\overline{BREQ}$ is 16 μs.

FIGURE 2 — MC6809 EXPANDED BLOCK DIAGRAM



*Internal Three-State Control

FIGURE 3 — 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 4, 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)

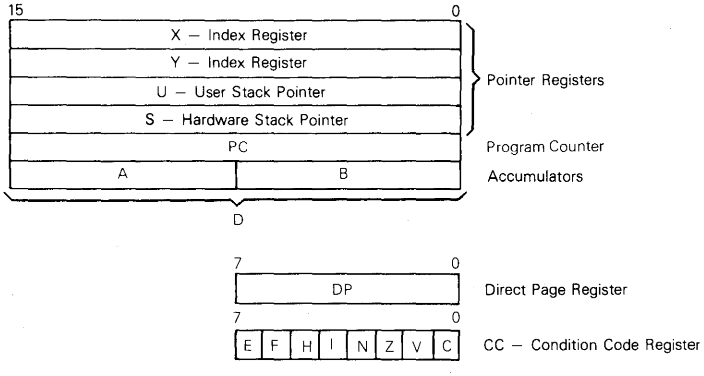
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

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 M6800 compatibility, all bits of this register are cleared during processor reset.

FIGURE 4 — 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 modified 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. This allows 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.

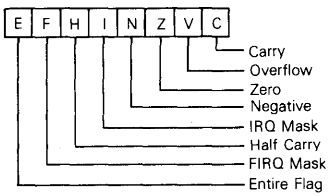
PROGRAM COUNTER

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 5.

FIGURE 5 — 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 twos 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.

3

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 twos-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 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 $\overline{FIRQ}$ 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. 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 LSTTL loads, and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus. Each pin will drive one Schottky TTL load or four LSTTL 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 6. The reset vectors are fetched from locations $FFFE_{16}$ and $FFFF_{16}$ (Table 1) when interrupt acknowledge is true, ($\overline{BA} \bullet BS = 1$). During initial power on, the $\overline{RESET}$ line should be held low until the clock oscillator is fully operational. See Figure 7.

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 $\overline{DMA}/\overline{BREQ}$ 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 ($\overline{RESET}$, $\overline{DMA}/\overline{BREQ}$), a halted state ($BA \bullet BS = 1$) can be achieved by pulling $\overline{HALT}$ low while $\overline{RESET}$ is still low. If $\overline{DMA}/\overline{BREQ}$ 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 the become halted. See Figure 8.

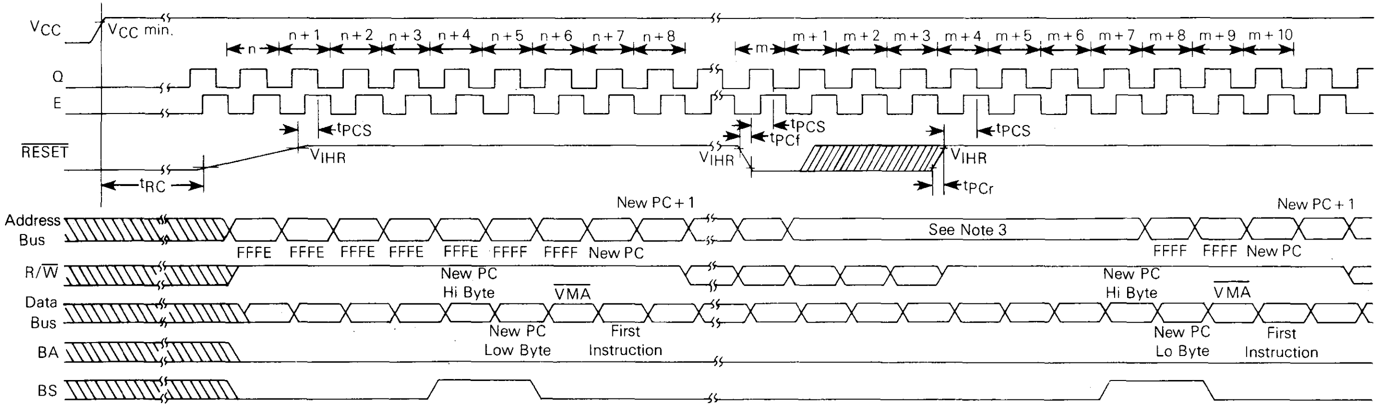
BUS AVAILABLE, BUS STATUS (BA, BS)

The bus available output is an indication of an internal control signal which makes the MOS buses of the MPU high impedance. 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).

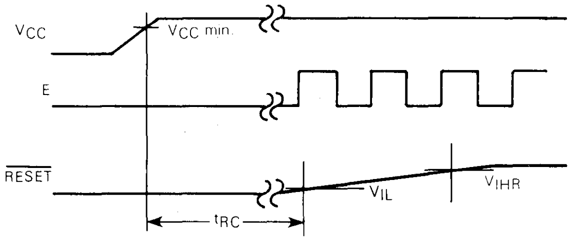
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt or Bus Grant Acknowledge

FIGURE 6 — RESET TIMING



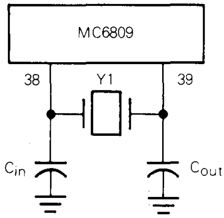
NOTES: 1. Parts with date codes prefixed by 7F or 5A will come out of RESET one cycle sooner than shown.
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
3. FFFE appears on the bus during RESET low time. Following the active transition of the RESET line, three more FFFE cycles will appear followed by the vector fetch.

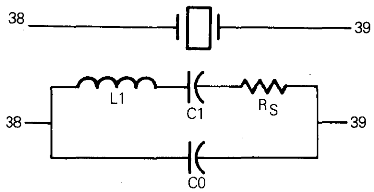
FIGURE 7 — CRYSTAL CONNECTIONS AND OSCILLATOR START UP



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.

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
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	>40 k	>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.

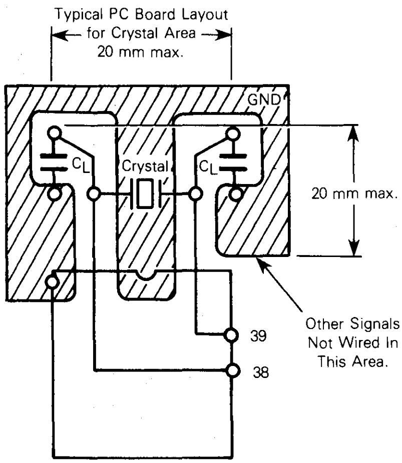
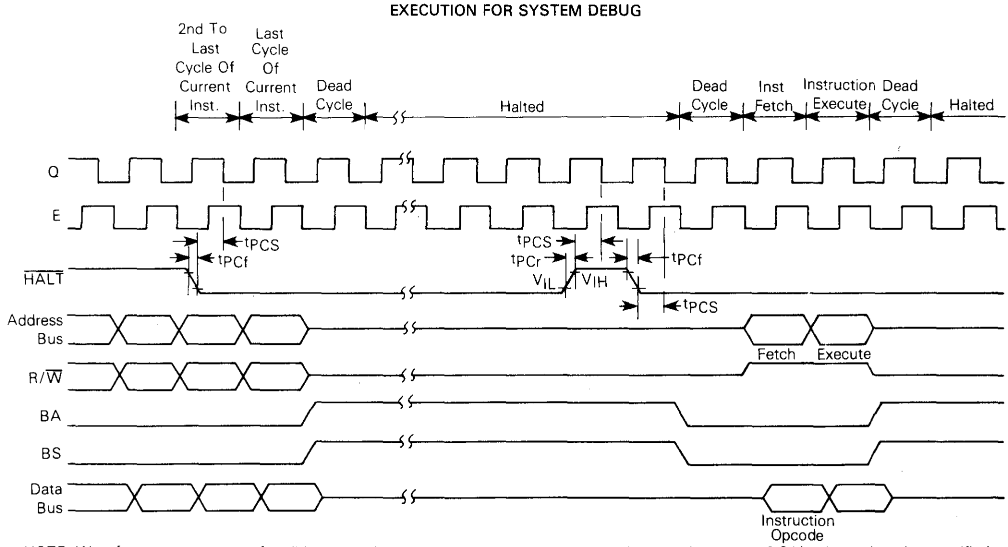


FIGURE 8 — $\overline{\text{HALT}}$ AND SINGLE INSTRUCTION EXECUTION FOR SYSTEM DEBUG



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 9.

FAST-INTERRUPT REQUEST ($\overline{\text{FIRQ}}$)*

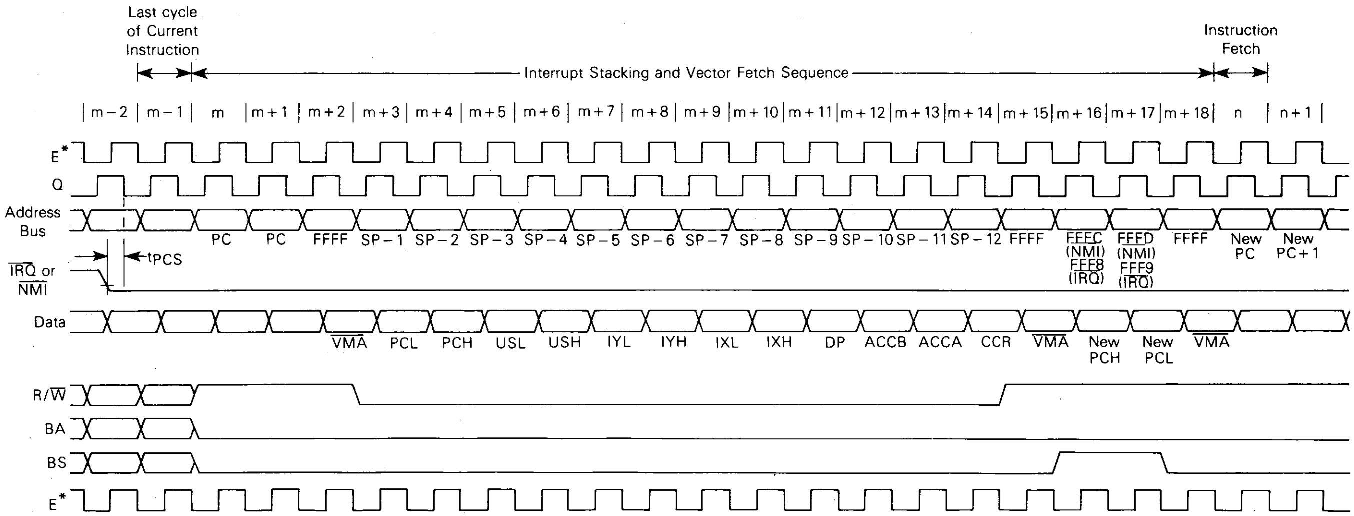
A low level on this input pin will initiate a fast interrupt sequence, provided its mask bit (F) in the CC is clear. This sequence has priority over the standard interrupt request ($\overline{\text{IRQ}}$), and is fast in the sense that it stacks only the contents of the condition code register and the 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.

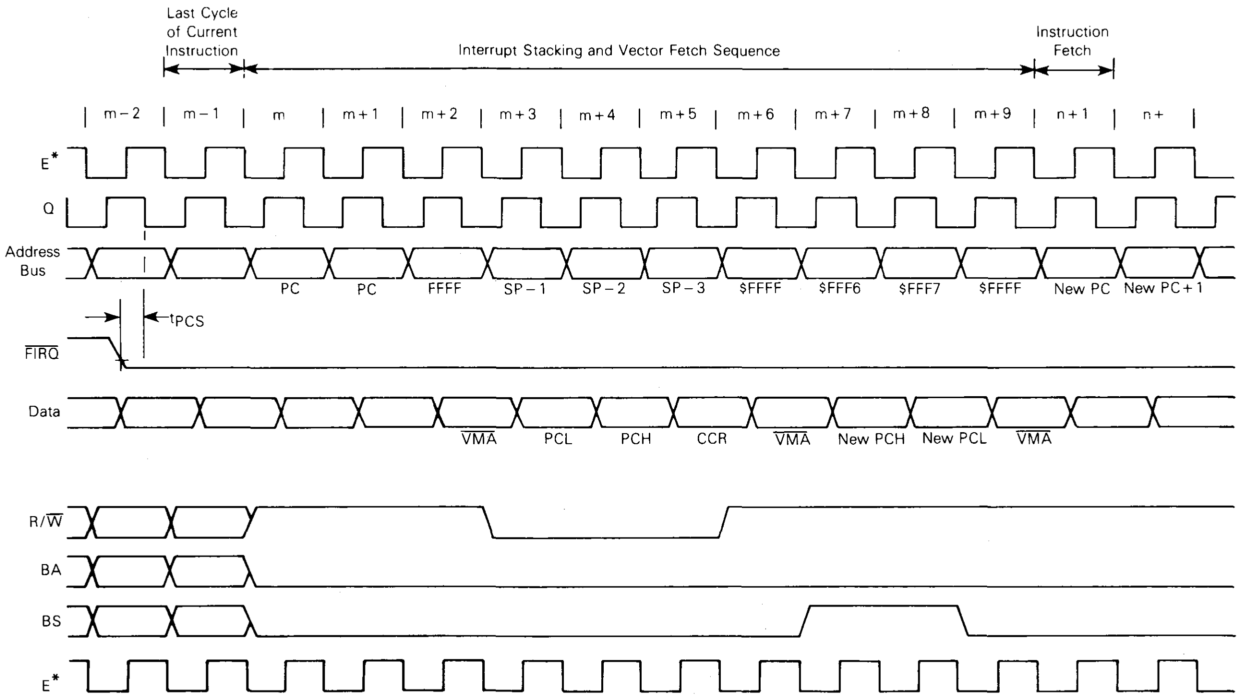
*$\overline{\text{NMI}}$, $\overline{\text{FIRQ}}$, and $\overline{\text{IRQ}}$ requests are sampled on the falling edge of Q. One cycle is required for synchronization before these interrupts are recognized. The pending interrupt(s) will not be serviced until completion of the current instruction unless 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.

FIGURE 9 — $\overline{IRQ}$ AND $\overline{NMI}$ 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 10 — $\overline{\text{FIRQ}}$ 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 7. 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 phase 2; Q is a quadrature clock signal which leads E. Q has no parrallel 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 11.

MRDY*

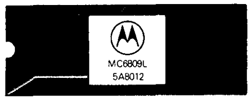
This input control signal allows stretching of E and Q to extend data-access time. E and Q operate normally while MRDY is high. When MRDY is low, E and Q may be stretched in integral multiples of quarter (¼) bus cycles, thus allowing interface to slow memories, as shown in Figure 12(a). During non-valid memory access ($\overline{VMA}$ 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 $\overline{HALT}$ and $\overline{DMA}/\overline{BREQ}$).

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 12(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 $\overline{HALT}$ 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 twelfth week of 1980.

$\overline{DMA}/\overline{BREQ}$*

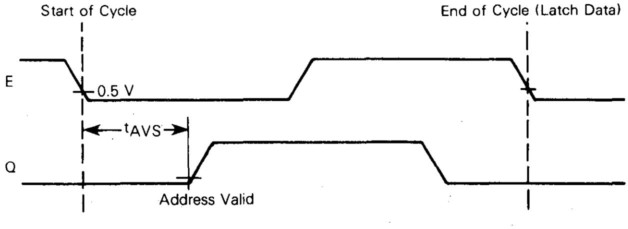
The $\overline{DMA}/\overline{BREQ}$ input provides a method of suspending execution and acquiring the MPU bus for another use, as shown in Figure 13. Typical uses include DMA and dynamic memory refresh.

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 $\overline{DMA}/\overline{BREQ}$ 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 14. The self-refresh counter is only cleared if $\overline{DMA}/\overline{BREQ}$ is inactive for two or more MPU cycles.

Typically, the DMA controller will request to use the bus by asserting $\overline{DMA}/\overline{BREQ}$ 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 $\overline{DMAVMA}$ signal which is LOW in any cycle when BA has changed.

FIGURE 11 — 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 $\overline{DMA}/\overline{BREQ}$ 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 $\overline{DMA}/\overline{BREQ}$ is released or after 16 bus cycles (14 DMA, two dead), whichever occurs first. While $\overline{DMA}/\overline{BREQ}$ is asserted it is sometimes necessary to pull MRDY low to allow DMA to/from slow memory/peripherals. As both MRDY and $\overline{DMA}/\overline{BREQ}$ control the internal MPU clocks, care must be exercised not to violate the maximum t_{cyc} specification for MRDY or $\overline{DMA}/\overline{BREQ}$. (Maximum t_{cyc} during MRDY or $\overline{DMA}/\overline{BREQ}$ is 16 μs.)

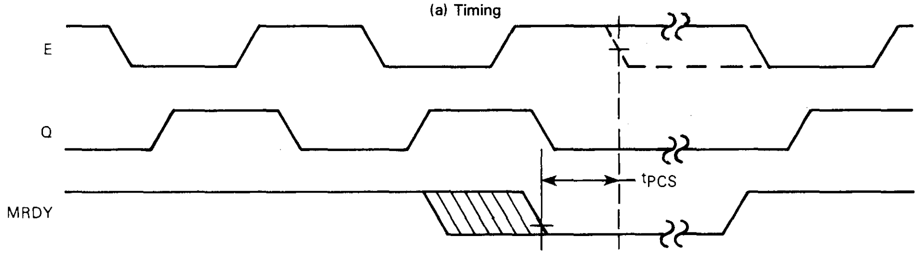
When BA goes low (either as a result of $\overline{DMA/BREQ}$ = HIGH or MPU self-refresh), the DMA device should be taken off the bus. Another dead cycle will elapse before the MPU accesses memory to allow transfer of bus mastership without contention.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{RESET}$ 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, $\overline{HALT}$, or $\overline{DMA/BREQ}$ can also alter the normal execution of instructions. Figure 15 illustrates the flowchart for the MC6809.

FIGURE 12 — MRDY TIMING AND SYNCHRONIZATION

(a) Timing



(b) Synchronization

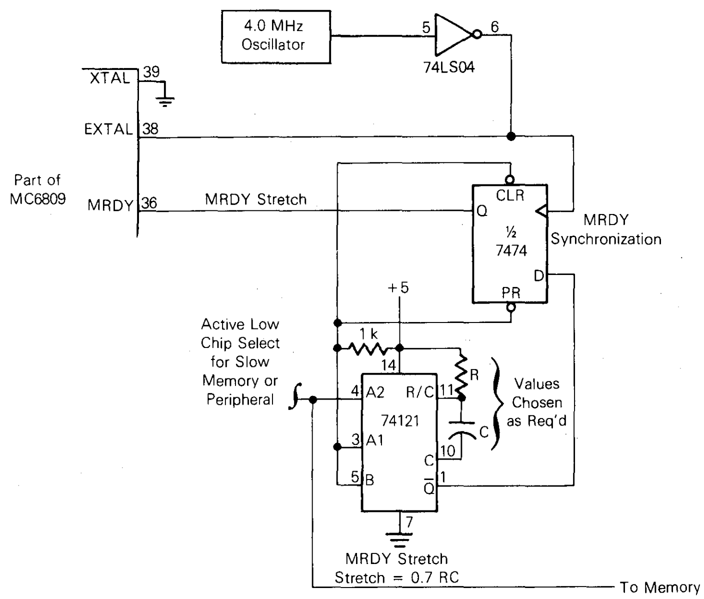


FIGURE 13 — TYPICAL DMA TIMING (<14 CYCLES)

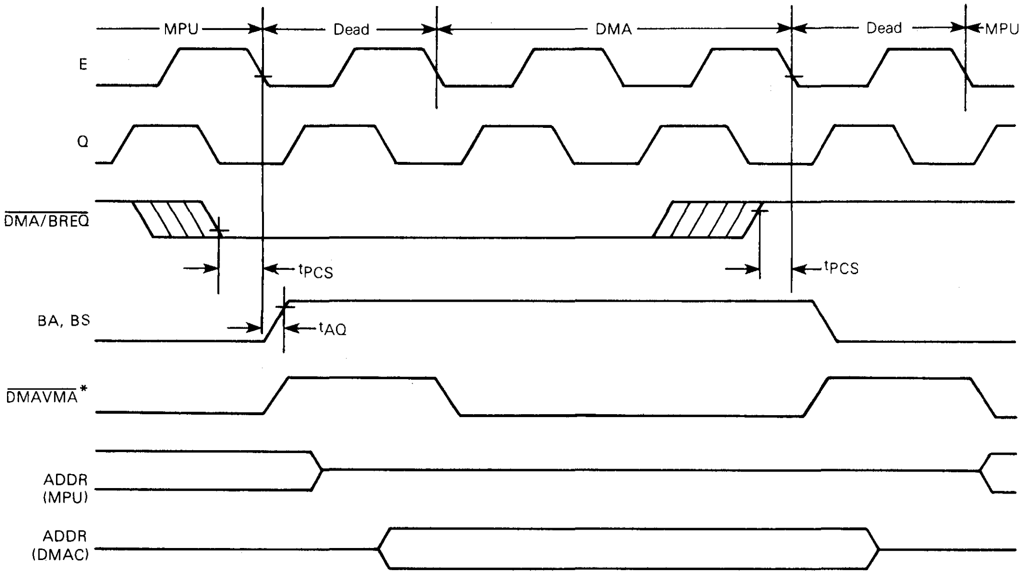
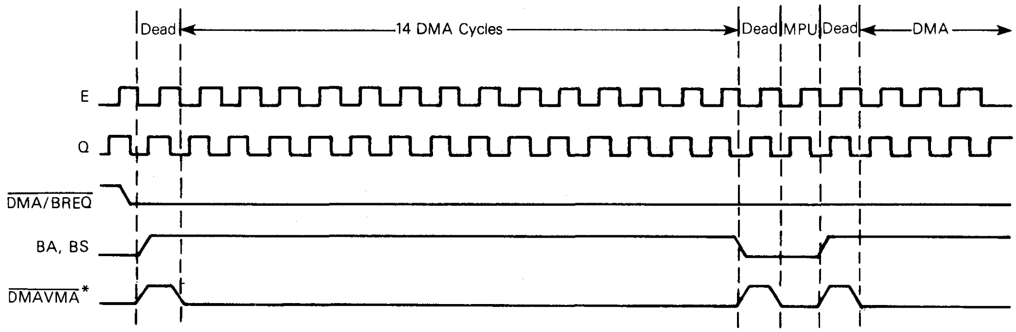


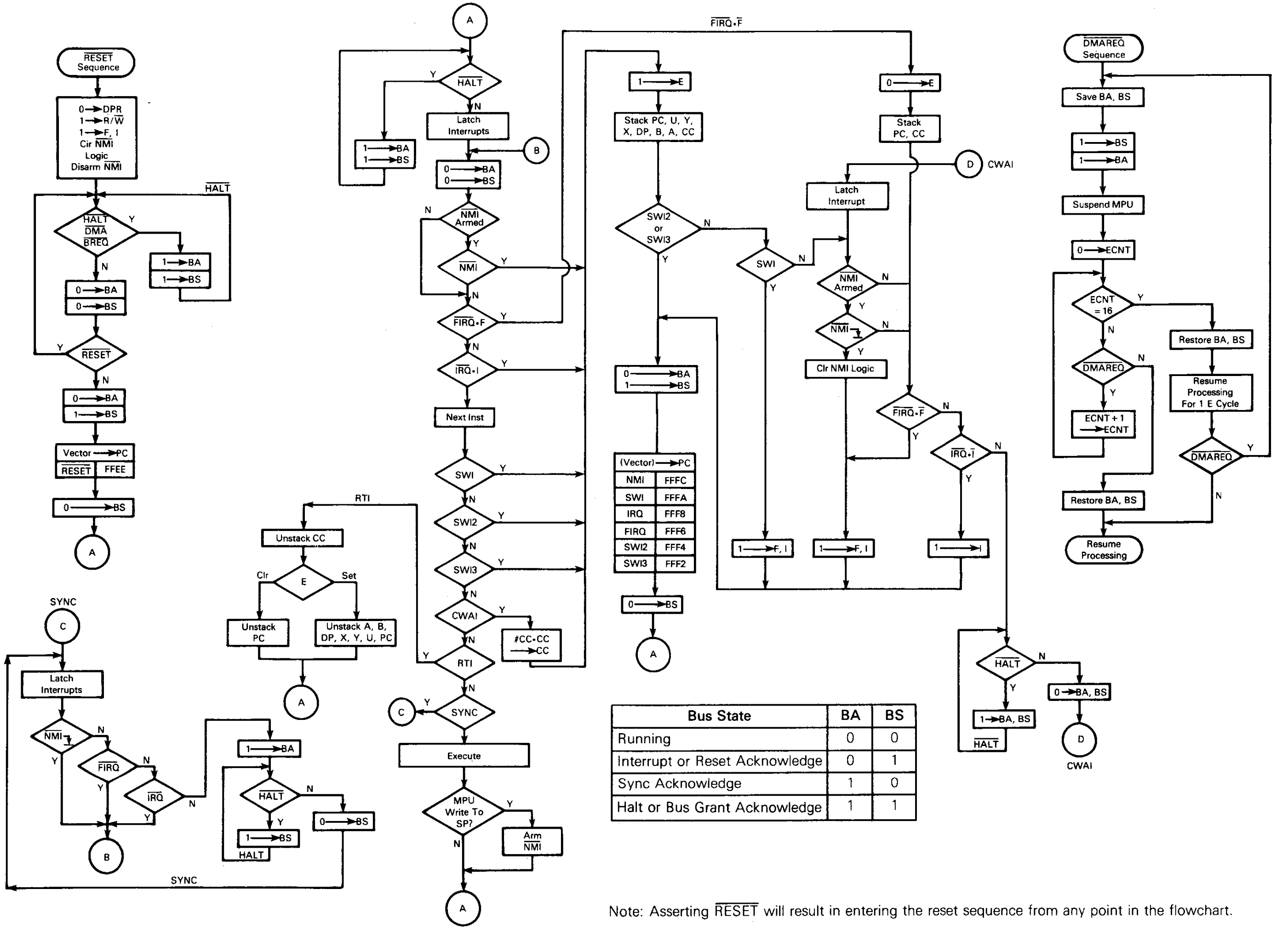
FIGURE 14 — AUTO-REFRESH DMA TIMING (>14 CYCLES)
(REVERSE CYCLE STEALING)



* $\overline{\text{DMAVMA}}$ is a signal which is developed externally, but is a system requirement for DMA. Refer to Application Note AN-820.

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 15 — FLOWCHART FOR MC6809 INSTRUCTIONS



Bus State	BA	BS
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 flowchart.

3

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 following 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. Examples 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 in the 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 eight bits of the address to be used. The upper eight 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 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

Postbyte Register Bit 7	6	5	4	3	2	1	0	Indexed Addressing Mode
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   O,X
LDA   S
```

CONSTANT OFFSET INDEXED — In this mode, a twos-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 twos complement 5-bit offset is included in the post-byte and, therefore, is most efficient in use of bytes and cycles. The twos complement 8-bit offset is contained in a single byte following the postbyte. The twos 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 Opcode	$\overset{+}{\sim}$	$\overset{+}{\#}$	Indirect Assembler Form	Indirect Postbyte Opcode	$\overset{+}{\sim}$	$\overset{+}{\#}$
Constant Offset From R	No Offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
(2s 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 (2s 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	8-Bit Offset	n, PCR	1xx01100	1	1	[n, PCR]	1xx11100	4	1
(2s 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 twos-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 allows 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 zero 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 auto increment
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 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 one 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 (one byte offset) and long (two 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]
```

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.

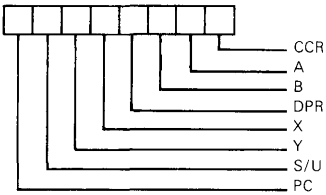
PSHU/PSHS

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.

Push/Pull Postbyte



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
↓

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 postbye define the source register, while bits 0-3 represent the destination register. These are denoted as follows:

Transfer/Exchange Postbyte

Source	Destination

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 auto increment and auto decrement 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	Substracts 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

Auto increment-by-two and auto decrement-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. The unsigned multiply also allows multiple-precision multiplications.

LONG AND SHORT RELATIVE BRANCHES

The MC6809 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 in-line 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 (Figure 18) 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. $\overline{VMA}$ 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.

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	$\overline{VMA}$ Cycle
5	FFFF	*	1	$\overline{VMA}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{VMA}$ 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	$\overline{VMA}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{VMA}$ Cycle
7	A000	7F	0	Store the Decremented Data

*The data bus has the data at that particular address.

INSTRUCTION SET TABLES

The instructions of the MC6809 have been broken down into five different categories. They are as follows:

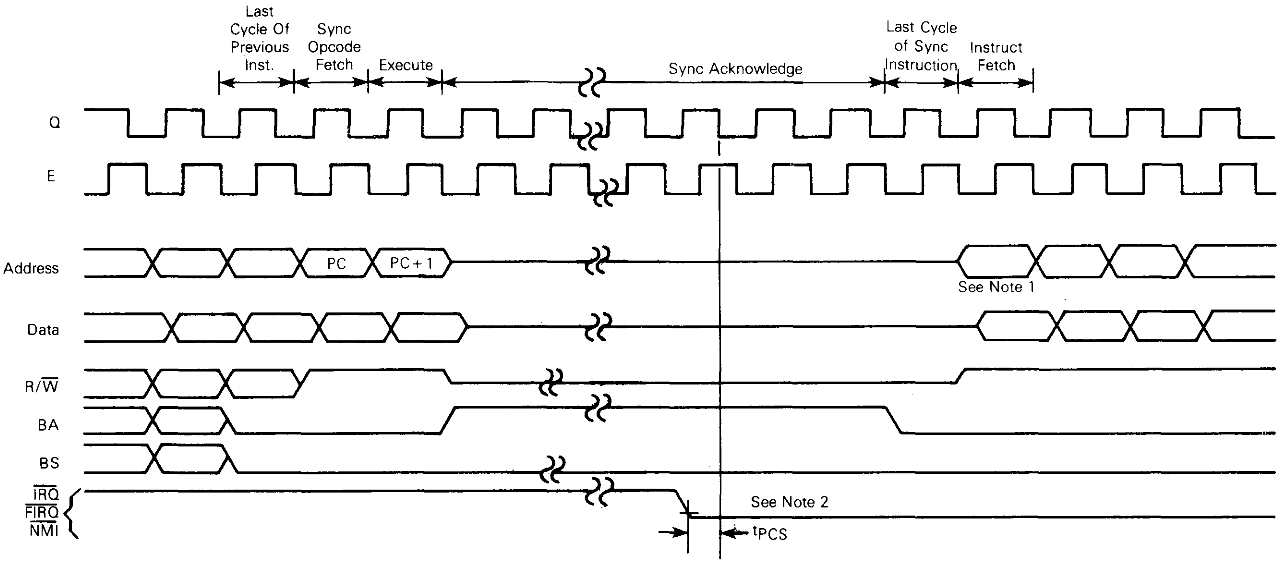
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

Hexadecimal values for the instructions are given in Table 9.

PROGRAMMING AID

Figure 19 contains a compilation of data that will assist in programming the MC6809.

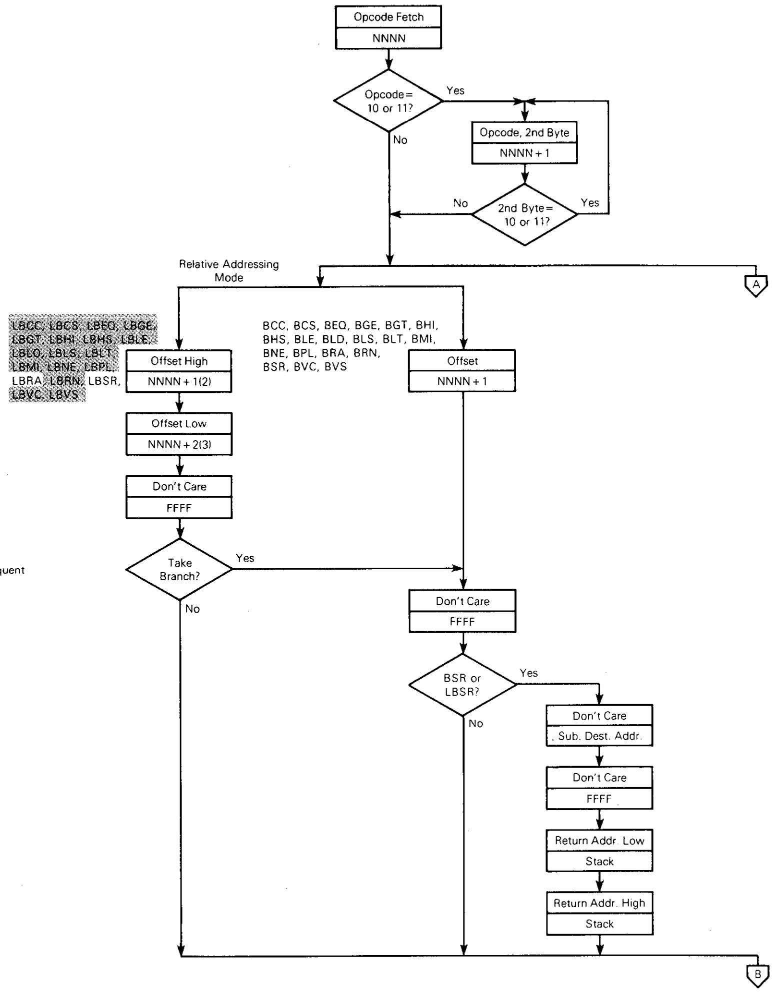
FIGURE 17 — SYNC TIMING



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 ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) interrupt processing continues 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 interrupt to be taken, although only one cycle is necessary to bring the processor out of SYNC.
3. 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 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 5)



NOTES:

1. Each state shows:

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2. Address NNNN is location of opcode.
3. If opcode is a two byte opcode subsequent addresses are in parenthesis (—).
4. Two-byte opcodes are highlighted.

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 5)

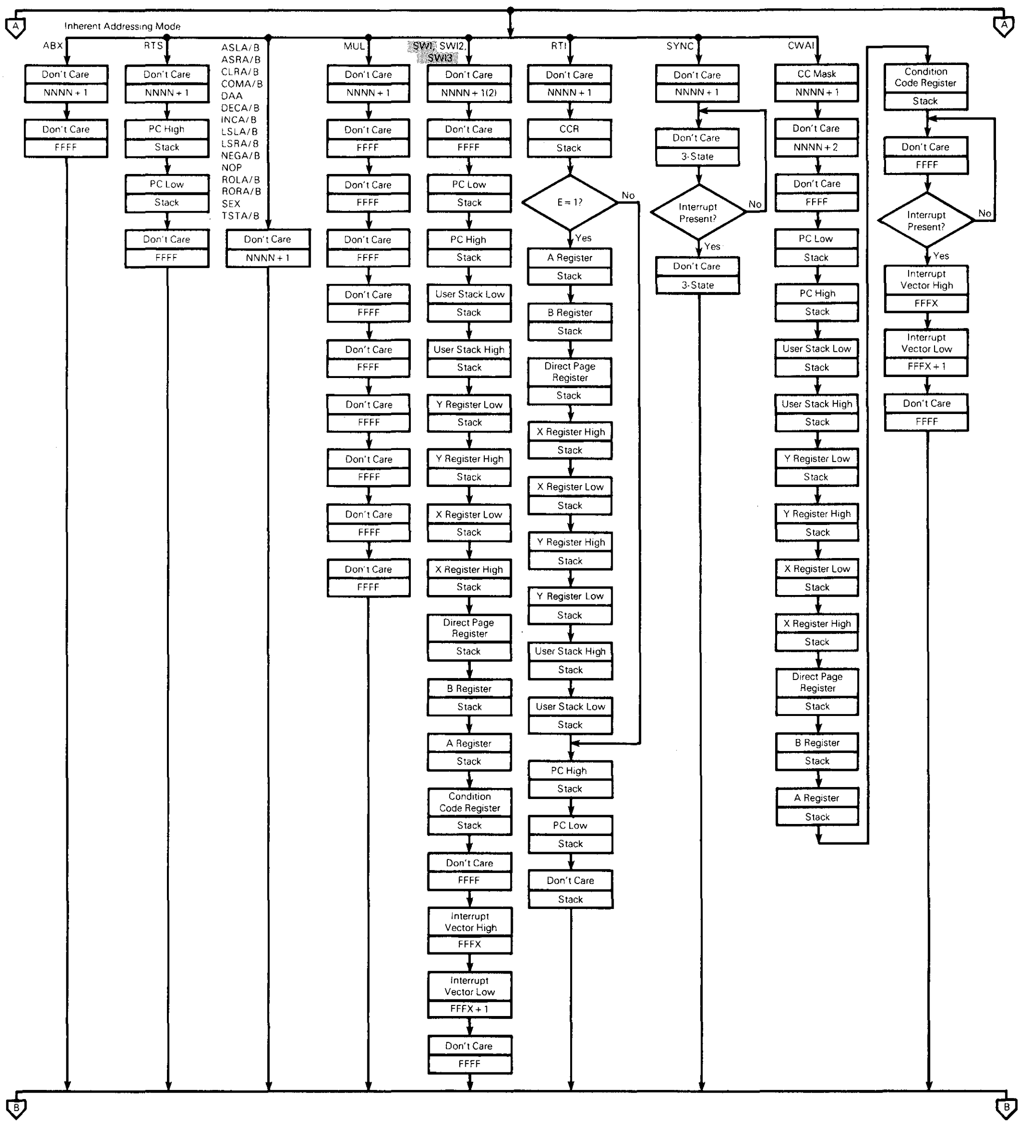


FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 3 of 5)

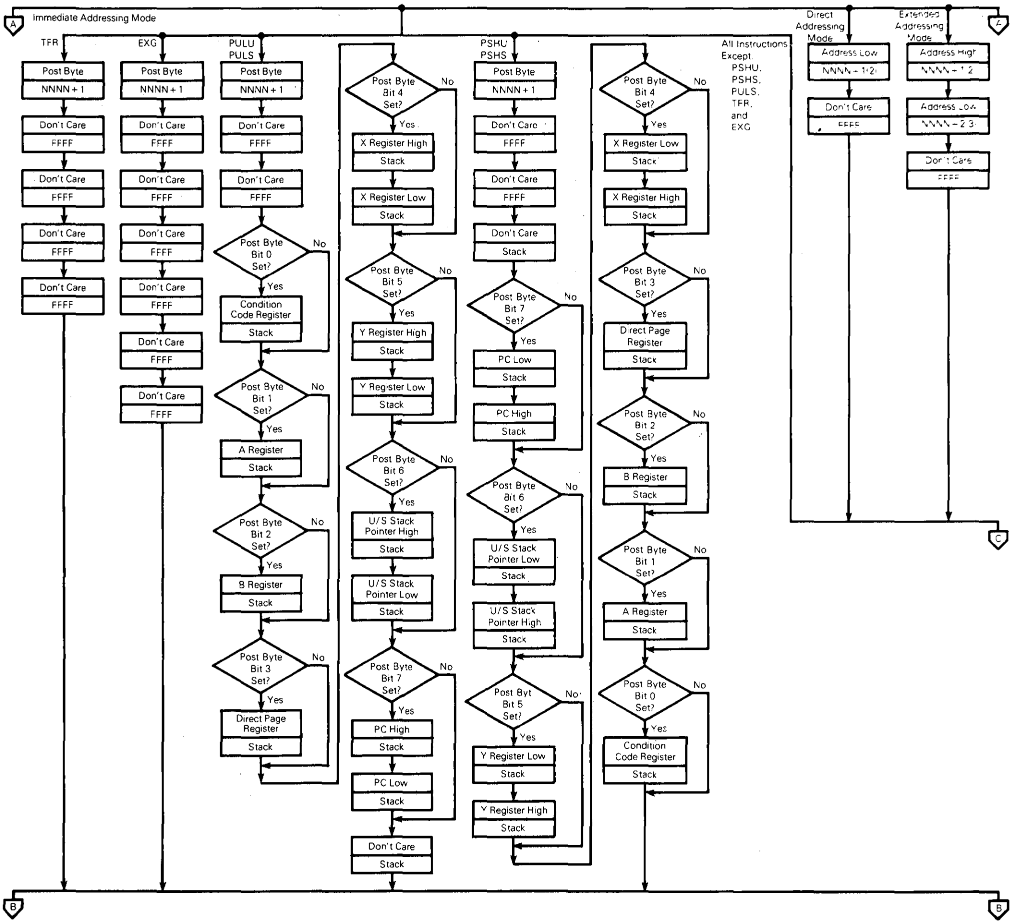
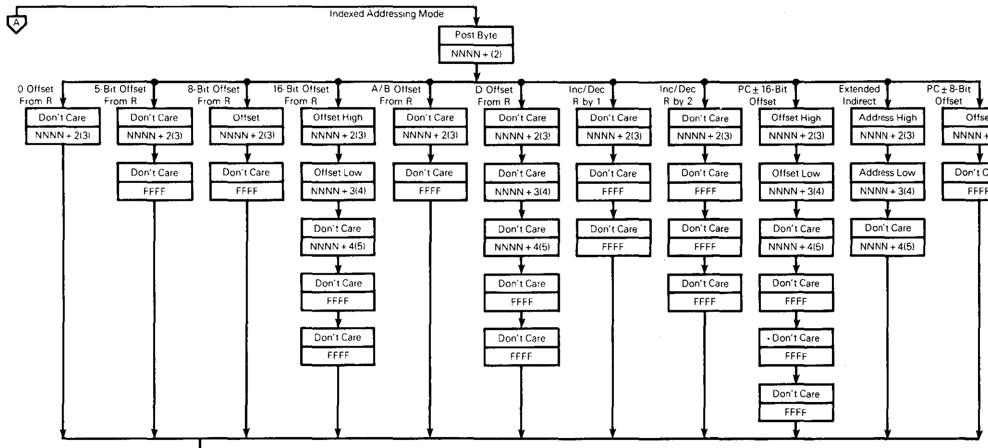
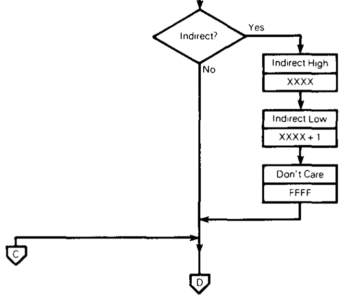


FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 5)



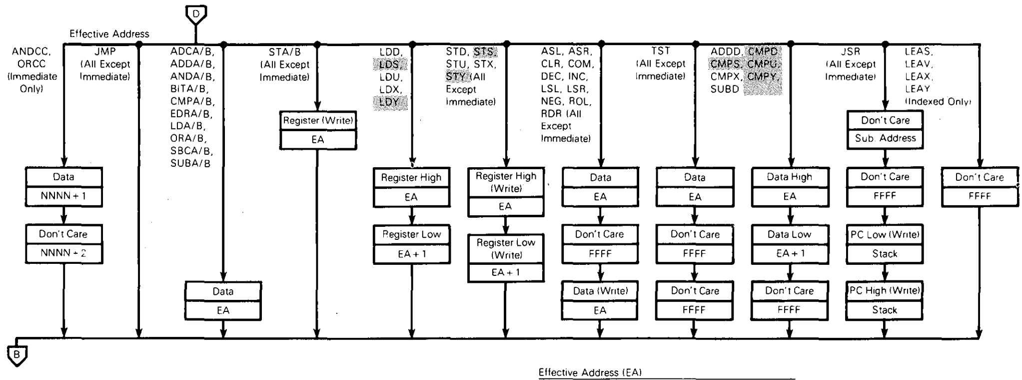


	XXXX
Constant Offset from R	
No Offset	Index Register
8-Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte: Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register – 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-bit Offset	Program Counter + Offset High Byte: Offset Low Byte
Extended Indirect	
16-Bit Address	Address High Byte: Addres Low Byte

*The index register is incremented following the indexed access.

3

FIGURE 18 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 5)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High: Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register − 1
Decrement by 2	Index Register − 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte: Offset Low Byte
Direct	Direct Page Register: Address Low
Extended	Address High: Address Low
Immediate	NNNN + 1

*The index register is incremented following the indexed access.

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) stack with either 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 stack with either 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, X, 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 2s 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 2s 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	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*	–			68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	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	Inherent	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	Immed	8	2	4E	*				7E	JMP		4	3
1F	TFR	Immed	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

3

TABLE 9 — HEXADECIMAL VALUES OF MACHINE CODES (CONTINUED)

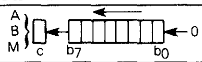
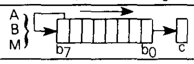
OP	Mnem	Mode	~	#
90	SUBA	Direct	4	2
91	CMPA		4	2
92	SBCA		4	2
93	SUBD		6	2
94	ANDA		4	2
95	BITA		4	2
96	LDA		4	2
97	STA		4	2
98	EORA		4	2
99	ADCA		4	2
9A	ORA		4	2
9B	ADDA		4	2
9C	CMPX		6	2
9D	JSR		7	2
9E	LDX		5	2
9F	STX	Direct	5	2
A0	SUBA	Indexed	4+	2+
A1	CMPA		4+	2+
A2	SBCA		4+	2+
A3	SUBD		6+	2+
A4	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

OP	Mnem	Mode	~	#
Page 2 and 3 Machine Codes				
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 19 — 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
- ∨ Logical or
- Λ Logical and
- ⊻ Logical Exclusive or

3

FIGURE 19 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Immediate Op	~	#	Direct Op	~	#	Indexed[1] Op	~	#	Extended Op	~	#	Inherent Op	~	#	Description	5 H	3 N	2 Z	1 V	0 C
LSL	LSLA													48	2	1	A B M C b7 b0	•	↕	↕	↕	↕
	LSLB													58	2	1		•	↕	↕	↕	↕
	LSL				08	6	2	68	6+	2+	78	7	3					•	↕	↕	↕	↕
LSR	LSRA													44	2	1	A B M 0 b7 b0 C	•	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 ∨ M → A	•	↕	↕	0	•
	ORB	CA	2	2	DA	4	2	EA	4+	2+	FA	5	3				B ∨ M → B	•	↕	↕	0	•
	ORCC	1A	3	2													CC ∨ 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	A B M C b7 b0	•	↕	↕	↕	↕
	ROLB													59	2	1		•	↕	↕	↕	↕
	ROL				09	6	2	69	6+	2+	79	7	3					•	↕	↕	↕	↕
ROR	RORA													46	2	1	A B M C b7 b0	•	↕	↕	•	↕
	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	•	•	•	•	•
	SWI2[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI3[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 19 — PROGRAMMING AID (CONTINUED)
Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~5	#	Description	5 H	3 N	2 Z	1 V	0 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=1	•	•	•	•	•
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 OP	~5	#	Description	5 H	3 N	2 Z	1 V	0 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

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

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

NOTES:
1. All conditional branches have both short and long variations.
2. All short branches are two bytes and require three 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 four bytes and six cycles if the branch is taken or five cycles if the branch is not taken.

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Ceramic L Suffix	1.0 MHz	0°C to 70°C	MC6809L
	1.0 MHz	−40°C to 85°C	MC6809CL
	1.5 MHz	0°C to 70 °C	MC68A09L
	1.5 MHz	−40°C to 85°C	MC68A09CL
	2.0 MHz	0°C to 70 °C	MC68B09L
	2.0 MHz	−40°C to 85°C	MC68B09CL
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6809P
	1.0 MHz	−40°C to 85°C	MC6809CP
	1.5 MHz	0°C to 70°C	MC68A09P
	1.5 MHz	−40°C to 85°C	MC68A09CP
	2.0 MHz	0°C to 70°C	MC68B09P
	2.0 MHz	−40°C to 85°C	MC68B09CP

Package Type	Frequency	Temperature Range	Order Number
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6809S
	1.0 MHz	−40°C to 85°C	MC6809CS
	1.5 MHz	0°C to 70°C	MC68A09S
	1.5 MHz	−40°C to 85°C	MC68A09CS
	2.0 MHz	0°C to 70°C	MC68B09S
	2.0 MHz	−40°C to 85°C	MC68B09CS

3

MOTOROLA

MC6809E

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

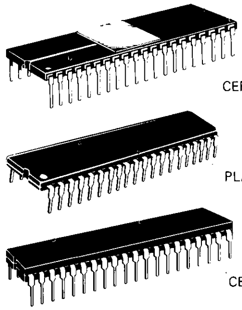
ARCHITECTURAL FEATURES
- Two 16-Bit Index Registers
- Two 16-Bit Indexable Stack Pointers
- Two 8-Bit Accumulators can be Concatenated to Form One 16-Bit Accumulator
- Direct Page Register Allows Direct Addressing Throughout Memory

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 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

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

8-BIT MICROPROCESSING UNIT



L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	$\overline{\text{HALT}}$
$\overline{\text{NMI}}$	2	39	TSC
$\overline{\text{IRQ}}$	3	38	LIC
$\overline{\text{FIRQ}}$	4	37	$\overline{\text{RESET}}$
BS	5	36	AVMA
BA	6	35	Q
V_{CC}	7	34	E
A0	8	33	BUSY
A1	9	32	R/$\overline{\text{W}}$
A2	10	31	D0
A3	11	30	D1
A4	12	29	D2
A5	13	28	D3
A6	14	27	D4
A7	15	26	D5
A8	16	25	D6
A9	17	24	D7
A10	18	23	A15
A11	19	22	A14
A12	20	21	A13

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 MC6809EC, MC68A09EC, MC68B09EC	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 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.

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

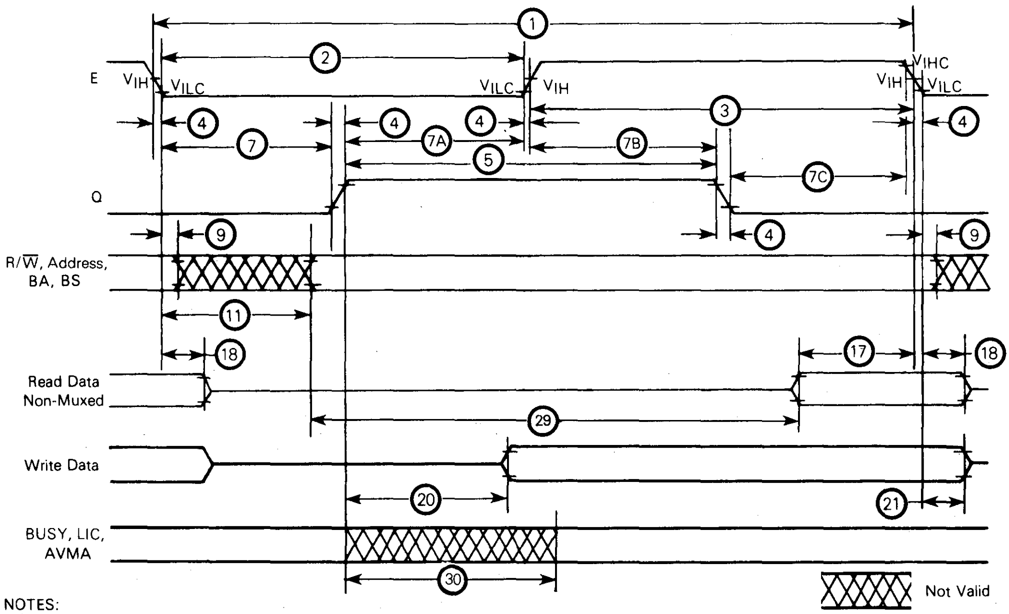
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, Q, $\overline{\text{RESET}}$ E	V_{IH} V_{IHR} V_{IHC}	V_{SS} + 2.0 V_{SS} + 4.0 V_{CC} − 0.75	 — —	V_{CC} V_{CC} V_{CC} + 0.3	V
Input Low Voltage	Logic, $\overline{\text{RESET}}$ E Q	V_{IL} V_{ILC} V_{ILQ}	V_{SS} − 0.3 V_{SS} − 0.3 V_{SS} − 0.3	— — —	V_{SS} + 0.8 V_{SS} + 0.4 V_{SS} + 0.6	V V V
Input Leakage Current (V_{in} = 0 to 5.25 V, V_{CC} = max)	Logic, Q, $\overline{\text{RESET}}$ E	I_{in}	— —	— —	2.5 100	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/$\overline{\text{W}}$ BA, BS, LIC, AVMA, BUSY	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, Logic Inputs, Q, $\overline{\text{RESET}}$ E	C_{in}	 — —	 10 30	 15 50	pF
	A0-A15, R/$\overline{\text{W}}$, BA, BS, LIC, AVMA, BUSY	C_{out}	—	10	15	pF
Frequency of Operation (E and Q Inputs)	MC6809E MC68A09E MC68B09E	f	0.1 0.1 0.1	— — —	1.0 1.5 2.0	MHz
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V, V_{CC} = max)	D0-D7 A0-A15, R/$\overline{\text{W}}$	I_{TSI}	— —	2.0 —	10 100	μA

*Capacitances are periodically tested rather than 100% tested.

BUS TIMING CHARACTERISTICS (See Notes 1, 2, 3, and 4)

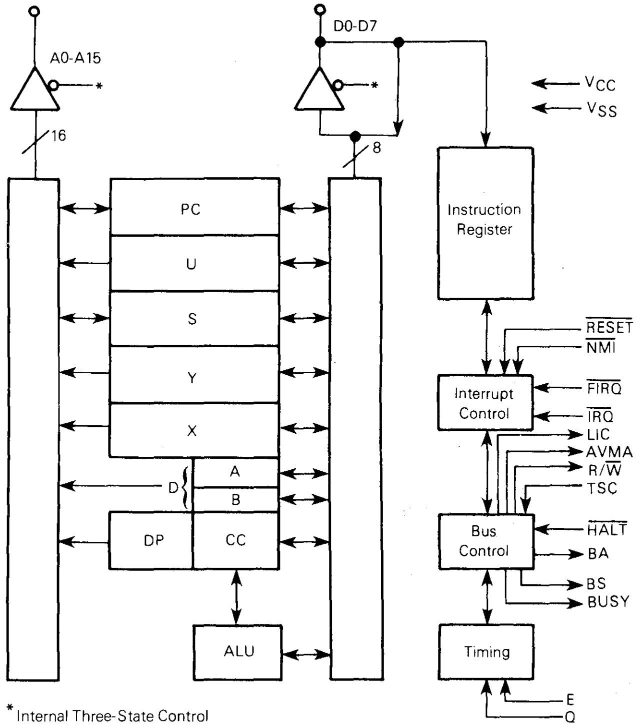
Ident. Number	Characteristics	Symbol	MC6809E Min	MC6809E Max	MC68A09E Min	MC68A09E Max	MC68B09E Min	MC68B09E Max	Unit
1	Cycle Time	t_{cyc}	1.0	10	0.667	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	450	9500	295	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and 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
30	Control Delay Time	t_{CD}	–	300	–	250	–	200	ns
	Interrupts, $\overline{HALT}$, $\overline{RESET}$, and TSC Setup Time (Figures 6, 7, 8, 9, 12, and 13)	t_{PCS}	200	–	140	–	110	–	ns
	TSC Drive to Valid Logic Level (Figure 13)	t_{TSV}	–	210	–	150	–	120	ns
	TSC Release MOS Buffers to High Impedance (Figure 13)	t_{TSR}	–	200	–	140	–	110	ns
	TSC Hi-Z Delay Time (Figure 13)	t_{TSD}	–	120	–	85	–	80	ns
	Processor Control Rise and Fall Time (Figure 7)	t_{PCr}, t_{PCf}	–	100	–	100	–	100	ns

FIGURE 1 — READ/WRITE DATA TO MEMORY OR PERIPHERALS TIMING DIAGRAM



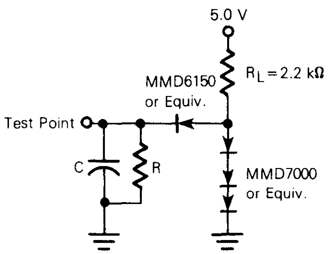
NOTES:
1. Voltage levels shown are $V_L \le 0.4$ V, $V_H \ge 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. Hold time (9) for BA and BS is not specified.
4. Usable access time is computed by: 1 – 4 – 11 max – 17.

FIGURE 2 — EXPANDED BLOCK DIAGRAM



* Internal Three-State Control

FIGURE 3 — 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 4, 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)

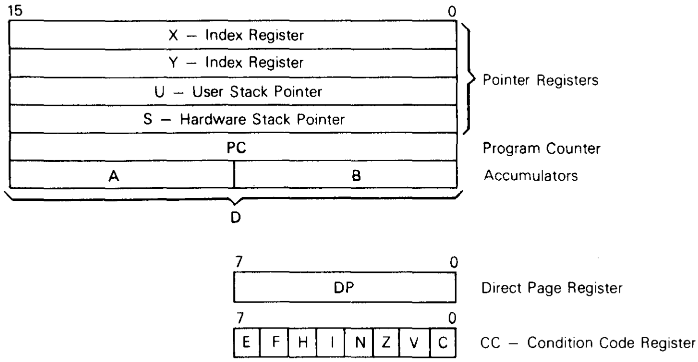
The A and B registers are general purpose accumulators which are used for arithmetic calculations and manipulation of data.

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 4 — 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 modified by an optional constant or register offset. During some indexed modes, the contents of the index register are incremented and 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 user stack pointer (U) is controlled exclusively by the programmer. This allows 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.

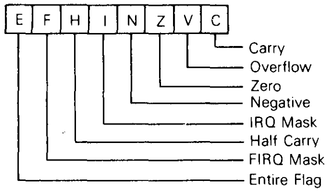
PROGRAM COUNTER

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 4.

FIGURE 5 — 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 twos 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 twos 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 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 $\overline{FIRQ}$ 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. 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 LSTTL loads and 90 pF.

DATA BUS (D0-D7)

These eight pins provide communication with the system bidirectional data bus. Each pin will drive one Schottky TTL load or four LSTTL 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 or when TSC is asserted.

$\overline{RESET}$

A low level on this Schmitt-trigger input for greater than one bus cycle will reset the MPU, as shown in Figure 6. 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 reset line should be held low until the clock input signals are fully operational.

Because the MC6809E $\overline{RESET}$ 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 state. While halted, the MPU will not respond to external real-time requests ($\overline{FIRQ}$, $\overline{IRQ}$) although $\overline{NMI}$ or $\overline{RESET}$ 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 $\overline{HALT}$ low while $\overline{RESET}$ is still low. See Figure 7.

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).

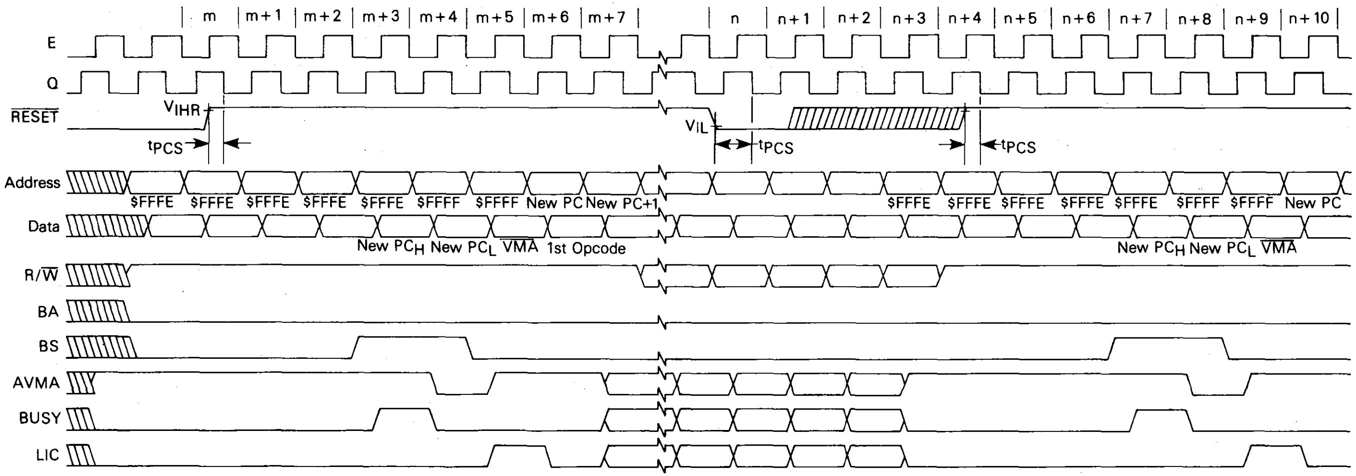
MPU State		MPU State Definition
BA	BS	
0	0	Normal (Running)
0	1	Interrupt or Reset Acknowledge
1	0	Sync Acknowledge
1	1	Halt Acknowledge

Interrupt Acknowledge is indicated during both cycles of a hardware vector fetch ($\overline{RESET}$, $\overline{NMI}$, $\overline{FIRQ}$, $\overline{IRQ}$, 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

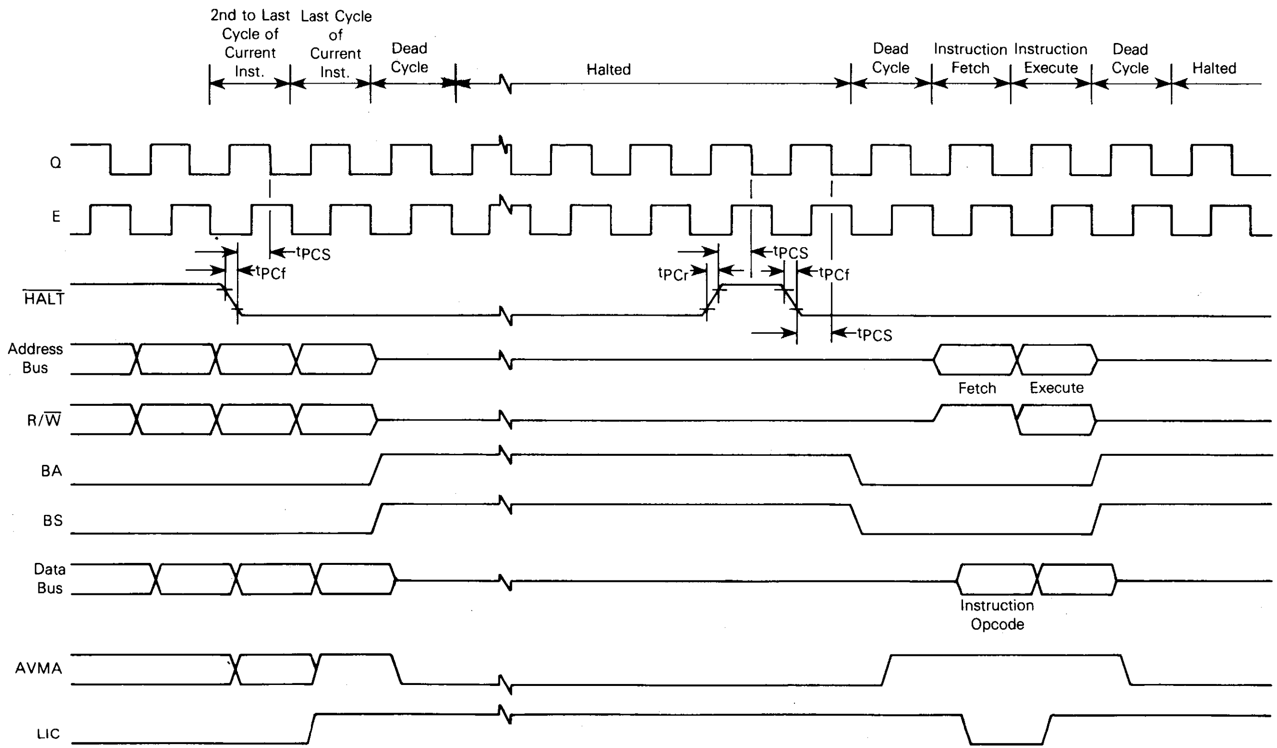
Memory Map For Vector Locations		Interrupt Vector Description
MS	LS	
FFFE	FFFF	$\overline{RESET}$
FFFC	FFFD	$\overline{NMI}$
FFFA	FFFB	SWI
FFF8	FFF9	$\overline{IRQ}$
FFF6	FFF7	$\overline{FIRQ}$
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reserved

FIGURE 6 — $\overline{\text{RESET}}$ 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 7 — $\overline{HALT}$ 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 ($\overline{NMI}$)*

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 $\overline{FIRQ}$, $\overline{IRQ}$, or software interrupts. During recognition of an $\overline{NMI}$, the entire machine state is saved on the hardware stack. After reset, an $\overline{NMI}$ will not be recognized until the first program load of the hardware stack pointer (S). The pulse width of $\overline{NMI}$ low must be at least one E cycle. If the $\overline{NMI}$ input does not meet the minimum set up with respect to Q, the interrupt will not be recognized until the next cycle. See Figure 8.

FAST-INTERRUPT REQUEST ($\overline{FIRQ}$)*

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 ($\overline{IRQ}$) 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 9.

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

A low level input on this pin will initiate an interrupt request sequence provided the mask bit (I) in the CC is clear. Since $\overline{IRQ}$ stacks the entire machine state, it provides a slower response to interrupts than $\overline{FIRQ}$. $\overline{IRQ}$ also has a lower priority than $\overline{FIRQ}$. Again, the interrupt service routine should clear the source of the interrupt before doing an RTI. See Figure 8.

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. Refer to **BUS TIMING CHARACTERISTICS** for E and Q and to Figure 10 which 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 multiprocessor 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 11. Timing information is given in Figure 12. 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 $\overline{HALT}$ 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 $\overline{RESET}$), 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/$\overline{W}$ 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).

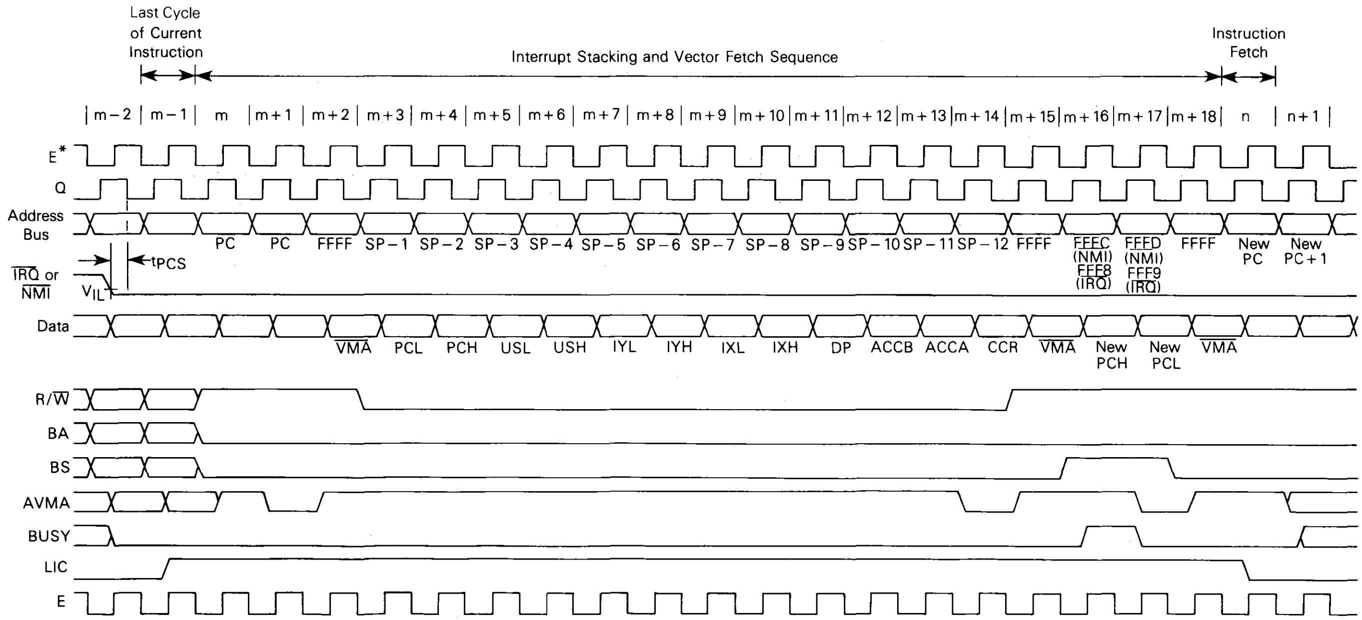
While E is low, TSC controls the address buffers and R/$\overline{W}$ directly. The data bus buffers during a write operation are in a high-impedance state until Q rises at which time, if TSC is true, they will remain in a high-impedance state. If TSC is held beyond the rising edge of E, then it will be internally latched, keeping the bus drivers in a high-impedance state for the remainder of the bus cycle. See Figure 13.

MPU OPERATION

During normal operation, the MPU fetches an instruction from memory and then executes the requested function. This sequence begins after $\overline{RESET}$ 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 $\overline{HALT}$ input can also alter the normal execution of instructions. Figure 14 is the flowchart for the MC6809E.

*$\overline{NMI}$, $\overline{FIRQ}$, and $\overline{IRQ}$ 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 $\overline{IRQ}$ and $\overline{FIRQ}$ do not remain low until completion of the current instruction, they may not be recognized. However, $\overline{NMI}$ is latched and need only remain low for one cycle. No interrupts are recognized or latched between the falling edge of $\overline{RESET}$ and the rising edge of BS indicating $\overline{RESET}$ acknowledge. See $\overline{RESET}$ sequence in the MPU flowchart in Figure 14.

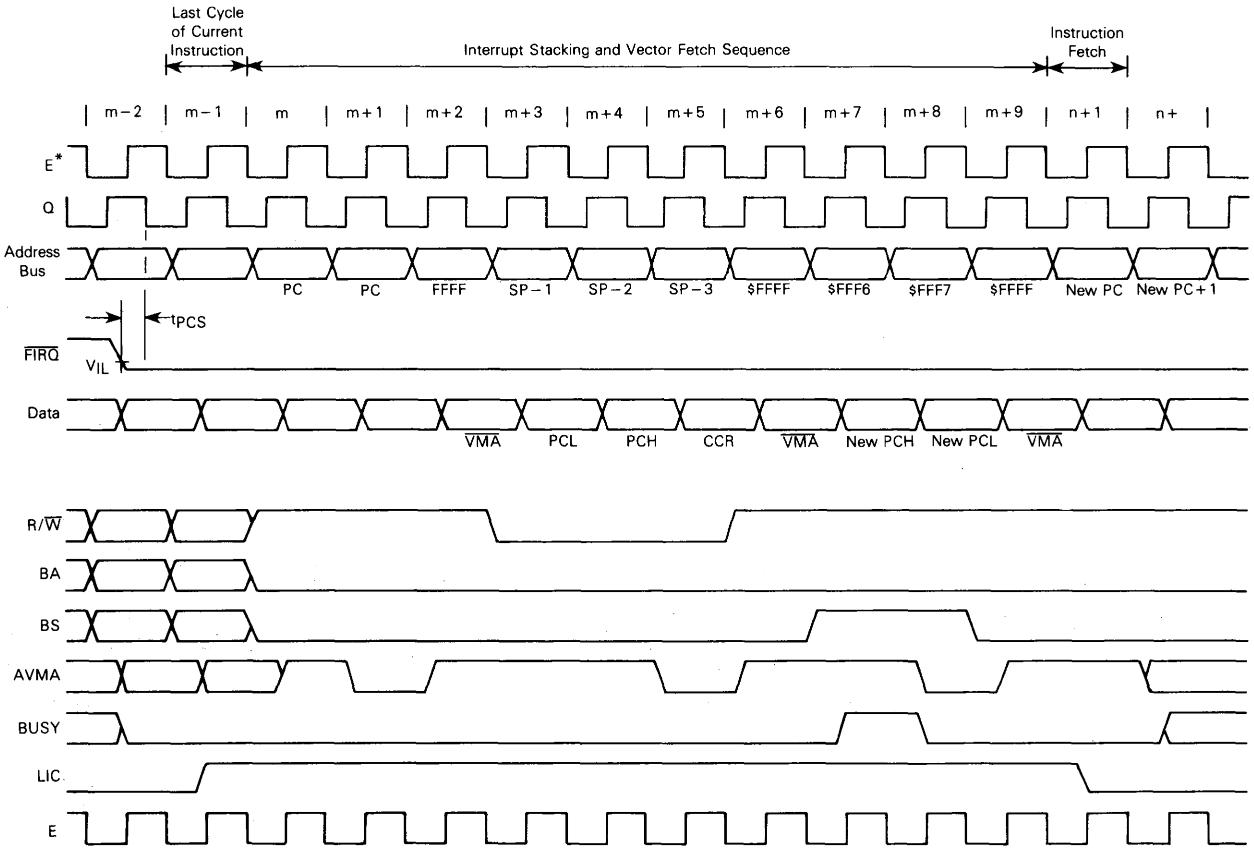
FIGURE 8 — $\overline{IRQ}$ AND $\overline{NMI}$ INTERRUPT TIMING



*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.

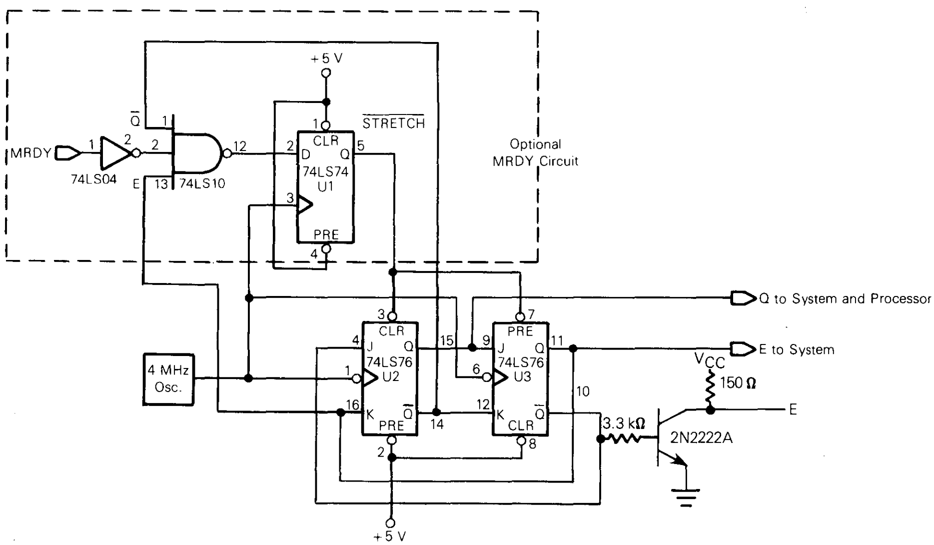
FIGURE 9 — $\overline{\text{FIRQ}}$ INTERRUPT TIMING

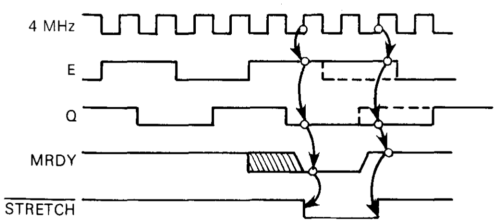


* E clock shown for reference only.

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 10 — CLOCK GENERATOR





NOTE: If optional circuit is not included the CLR and PRE inputs of U2 and U3 must be tied high.

FIGURE 11 — 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 12 — BUSY TIMING

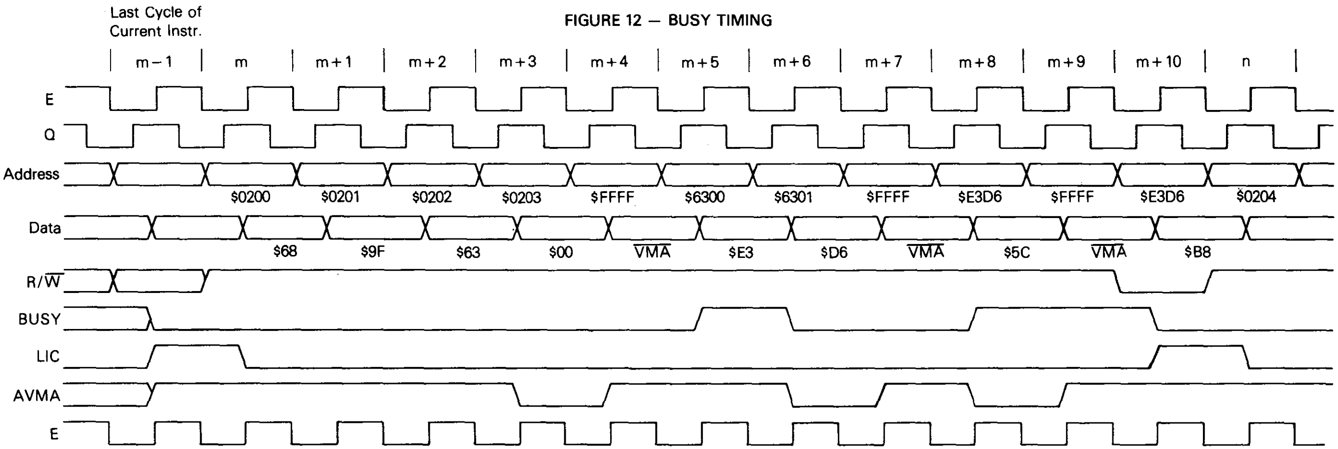
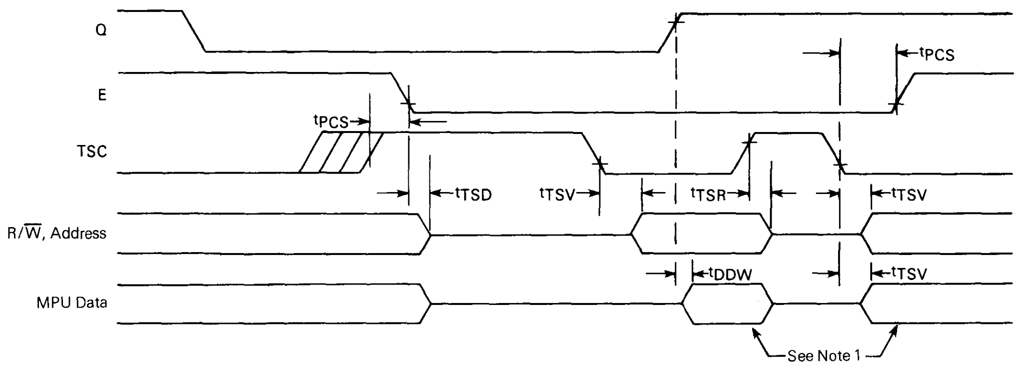


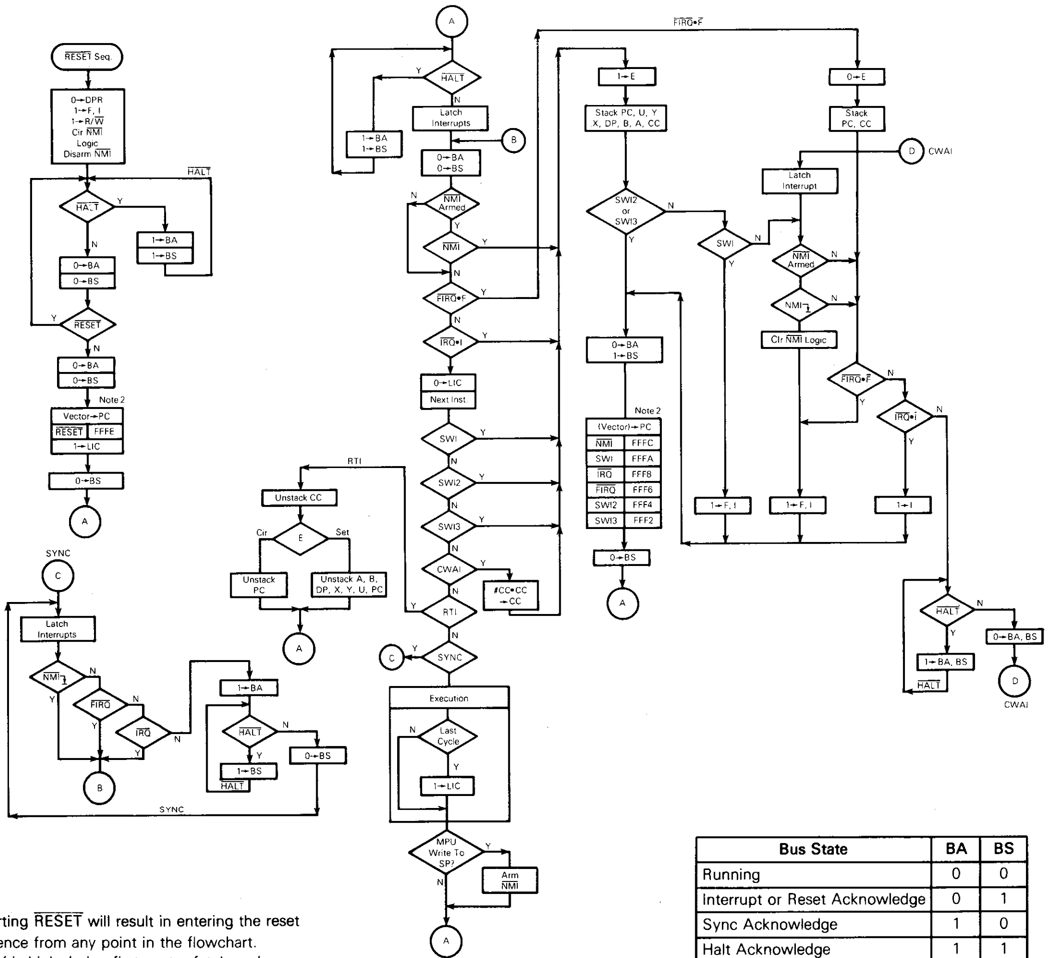
FIGURE 13 — TSC TIMING



NOTES:

1. 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.
2. 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 — 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 $\overline{RESET}$ will result in entering the reset sequence from any point in the flowchart.
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 following 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 eight bits of the address to be used. The upper eight 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 15 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 15 — INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

Post-Byte Register Bit 7	6	5	4	3	2	1	0	Indexed Addressing Mode
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 b7 = 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, twos 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 offset are available:

5-bit (−16 to +15)
8-bit (−128 to +127)
16-bit (−32768 to +32767)

The twos complement 5-bit offset is included in the postbyte and, therefore, is most efficient in use of bytes and cycles. The twos complement 8-bit offset is contained in a single byte following the postbyte. The twos 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

3

TABLE 2 — INDEXED ADDRESSING MODE

Type	Forms	Non Indirect Assembler Form	Non Indirect Postbyte Opcode	+ ~	+ #	Indirect Assembler Form	Indirect Postbyte Opcode	+ ~	+ #
Constant Offset From R (2s 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 (2s 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 (2s 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

+~ and +# indicate the number of additional cycles and bytes respectively for the particular indexing variation.

3

ACCUMULATOR-OFFSET INDEXED — This mode is similar to constant offset indexed except that the twos 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 creating 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 allows 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 zero 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 auto increment
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 (one byte offset) and long (two 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]
```

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.

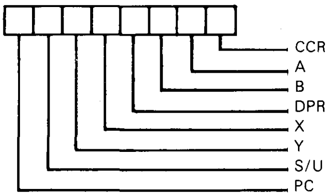
PSHU/PSHS

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.

Push/Pull Postbyte



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
↓

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 Postbyte

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 auto increment and auto decrement 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	Substracts 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

Auto increment-by-two and auto decrement-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 ($\overline{NMI}$) 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 in-line instruction. Figure 16 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 (Figure 16) 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 flowchart. $\overline{VMA}$ 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.

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	$\overline{VMA}$ Cycle
5	FFFF	*	1	$\overline{VMA}$ Cycle
6	A000	*	1	Computed Branch Address
7	FFFF	*	1	$\overline{VMA}$ 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/$\overline{W}$	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	$\overline{VMA}$ Cycle
5	A000	80	1	Read the Data
6	FFFF	*	1	$\overline{VMA}$ Cycle
7	FFFF	7F	0	Store the Decremented Data

*The data bus has the data at that particular address.

INSTRUCTION SET TABLES

The instructions of the MC6809E have been broken down into five different categories. They are as follows:

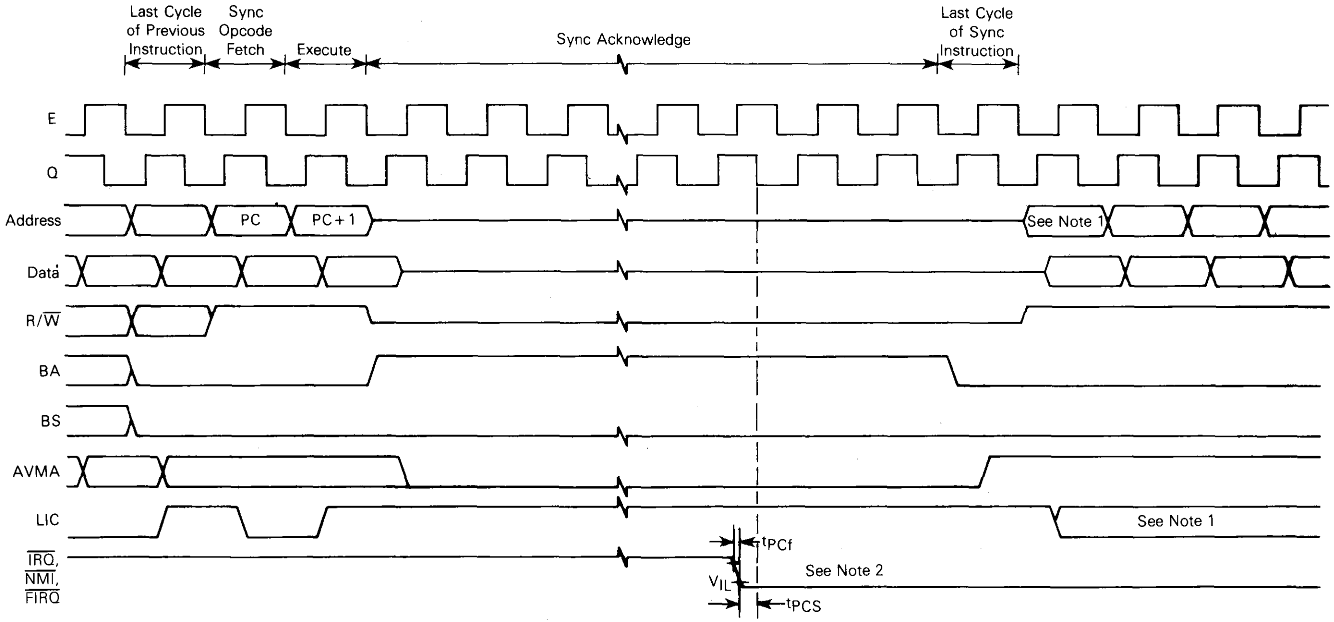
8-bit operation (Table 4)
16-bit operation (Table 5)
Index register/stack pointer instructions (Table 6)
Relative branches (long or short) (Table 7)
Miscellaneous instructions (Table 8)

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 16 — SYNC TIMING

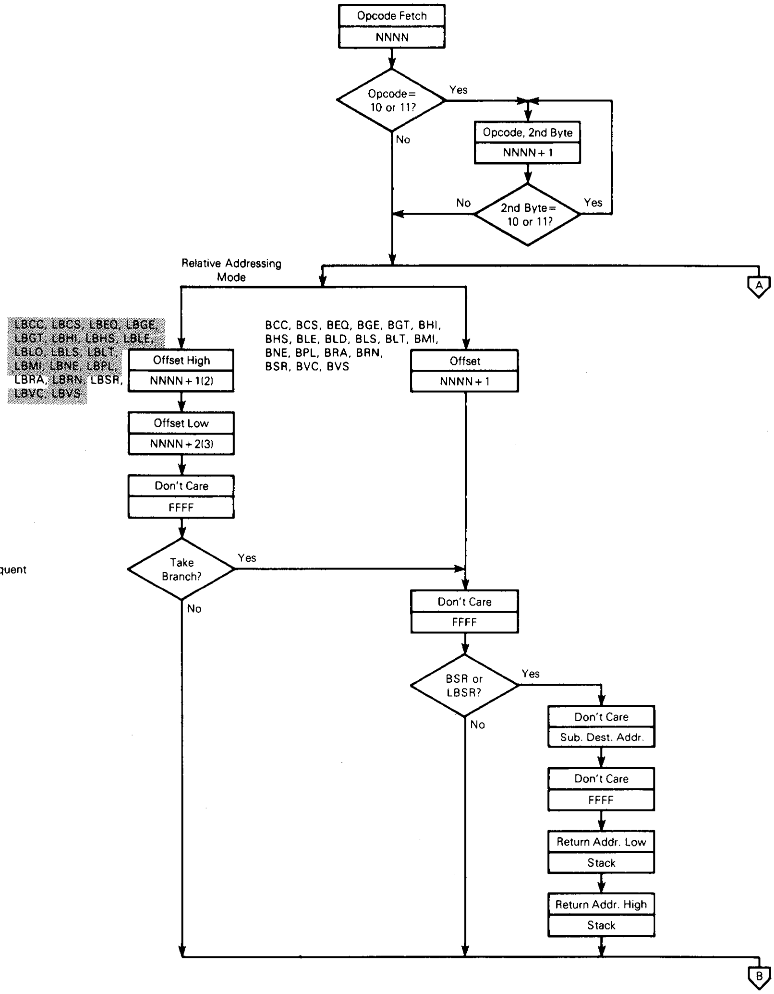


NOTES: 1. If the associated mask bit is set when the interrupt is requested, LIC will go low and this cycle will be an instruction fetch from address location PC + 1. However, if the interrupt is accepted ($\overline{NMI}$ or an unmasked $\overline{FIRQ}$ or $\overline{IRQ}$) LIC will remain high and interrupt processing will start with this cycle as m on Figures 8 and 9 (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.

3. 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 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 1 of 5)



NOTES:

1. Each state shows:

Data Bus	Offset High
Address Bus	NNNN + 1(2)

2. Address NNNN is location of opcode.
3. If opcode is a two byte opcode subsequent addresses are in parenthesis (–).
4. Two-byte opcodes are highlighted.

FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 2 of 5)

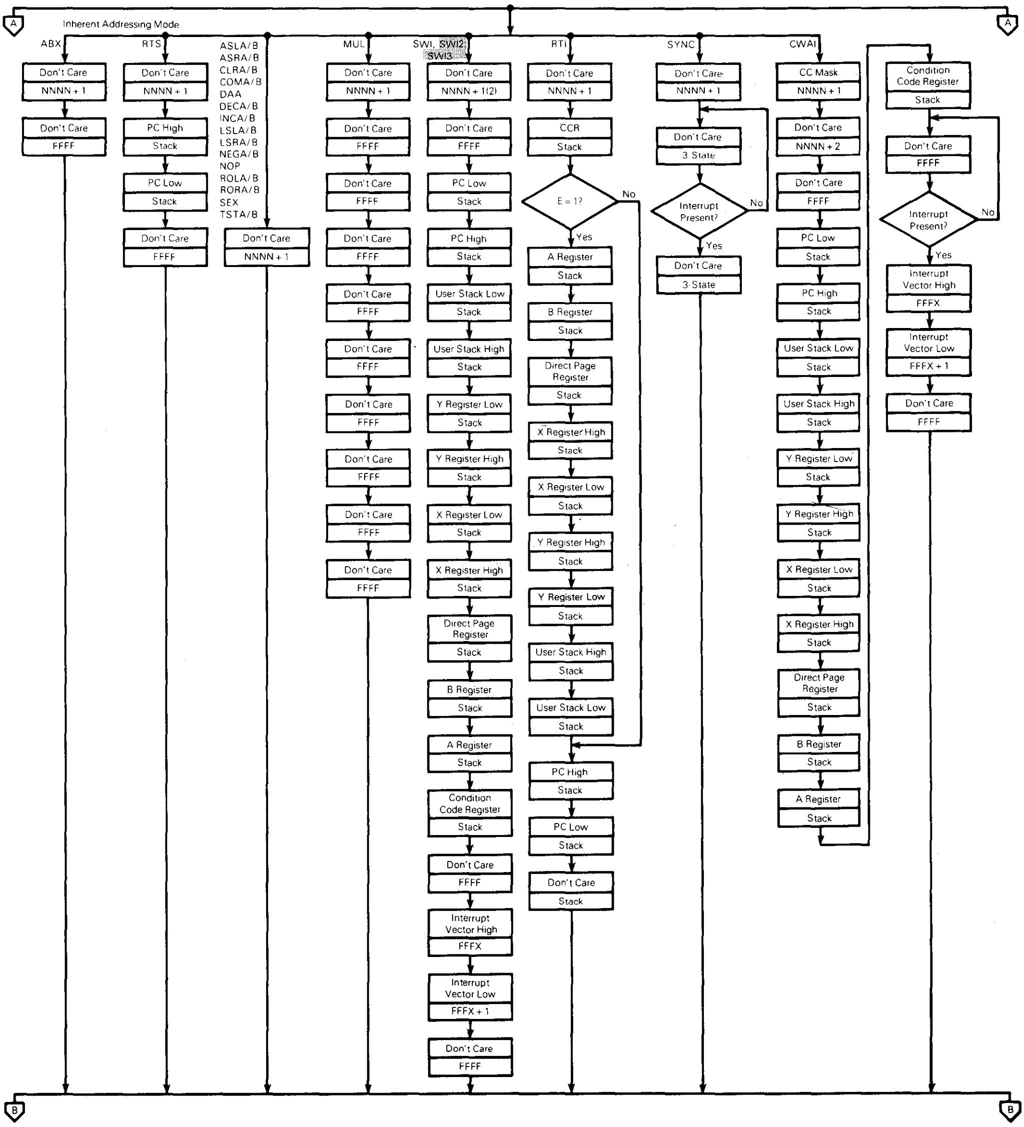


FIGURE 17 — CYCLE-BY-CYLE PERFORMANCE (Sheet 3 of 5)

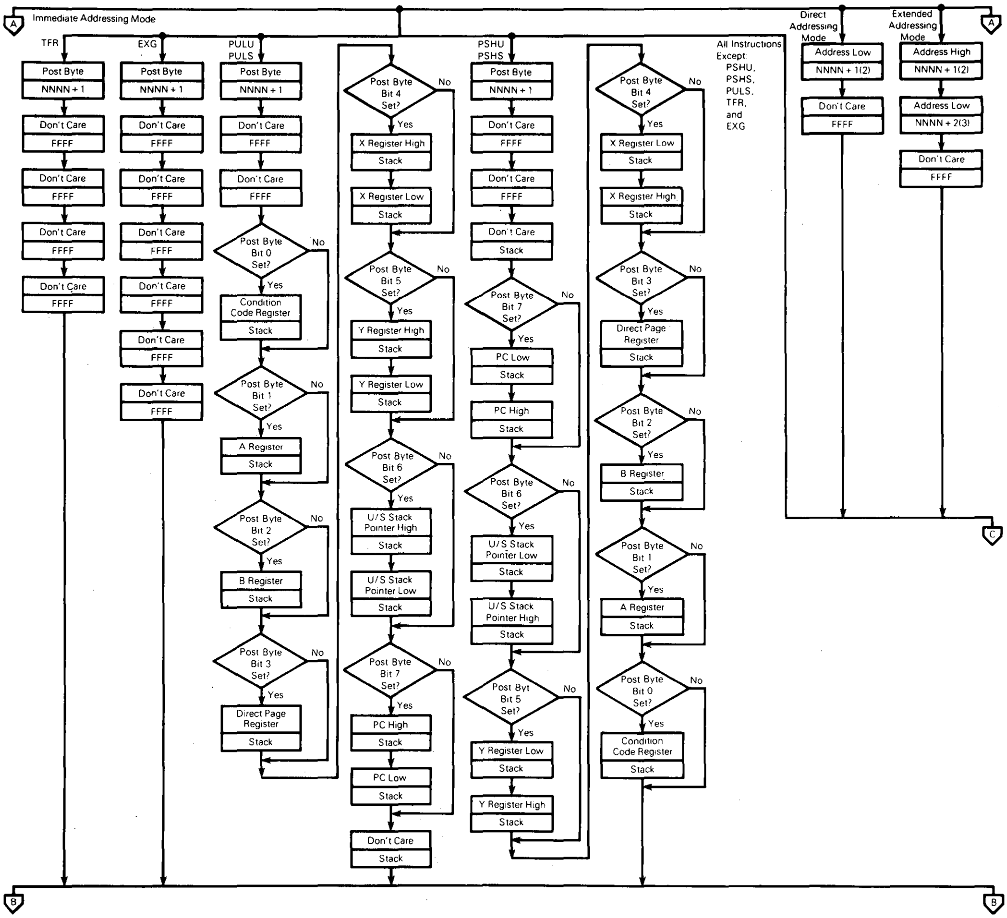
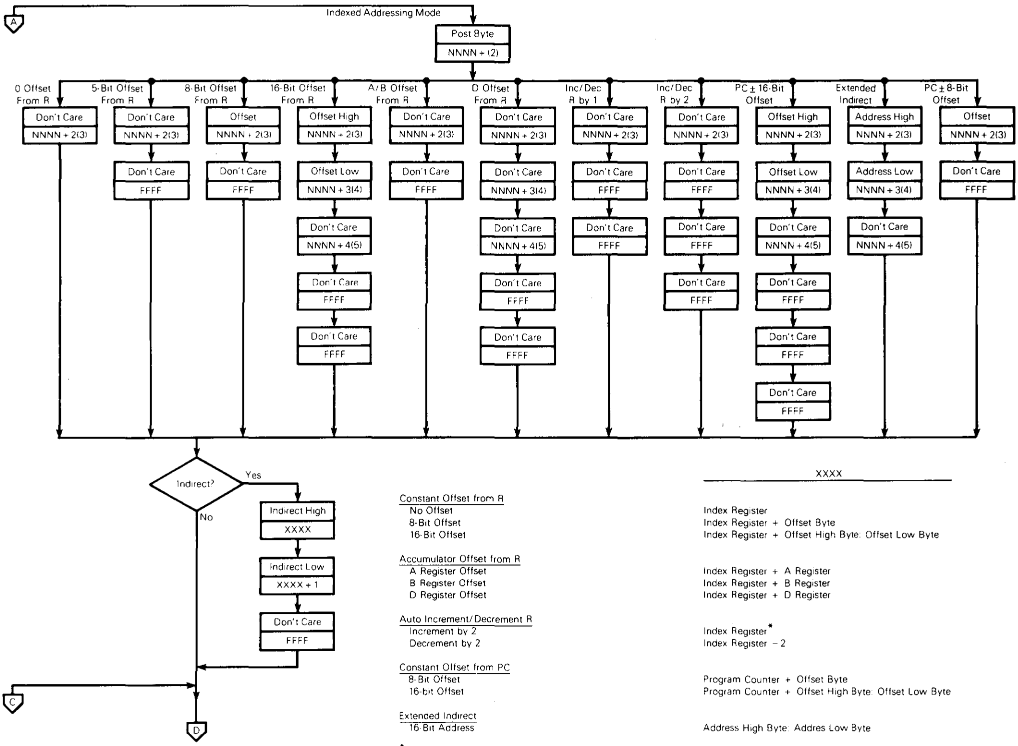


FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 4 of 5)

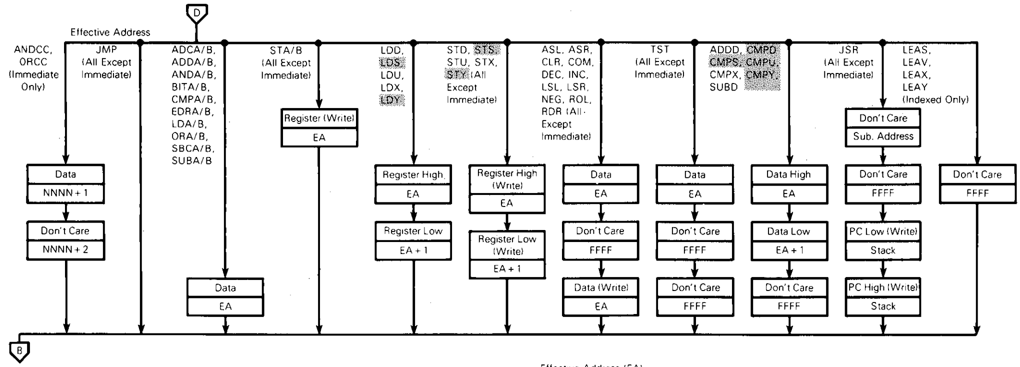


	XXXX
Constant Offset from R	
No Offset	Index Register
8-Bit Offset	Index Register + Offset Byte
16-Bit Offset	Index Register + Offset High Byte: Offset Low Byte
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 2	Index Register*
Decrement by 2	Index Register − 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-bit Offset	Program Counter + Offset High Byte: Offset Low Byte
Extended Indirect	
16-Bit Address	Address High Byte: Addres Low Byte

*The index register is incremented following the indexed access

3

FIGURE 17 — CYCLE-BY-CYCLE PERFORMANCE (Sheet 5 of 5)



	Effective Address (EA)
Constant Offset from R	
No Offset	Index Register
5-Bit Offset	Index Register
8-Bit Offset	Index Register + Post Byte
16-Bit Offset	Index Register + Post Byte High: Post Byte Low
Accumulator Offset from R	
A Register Offset	Index Register + A Register
B Register Offset	Index Register + B Register
D Register Offset	Index Register + D Register
Auto Increment/Decrement R	
Increment by 1	Index Register*
Increment by 2	Index Register*
Decrement by 1	Index Register − 1
Decrement by 2	Index Register − 2
Constant Offset from PC	
8-Bit Offset	Program Counter + Offset Byte
16-Bit Offset	Program Counter + Offset High Byte: Offset Low Byte
Direct	Direct Page Register: Address Low
Extended	Address High: Address Low
Immediate	NNNN + 1

*The index register is incremented following the indexed access.

3

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	Immed	5+	2	64	LSR		6+	2+
05	*				35	PULS	Immed	5+	2	65	*			
06	ROR		6	2	36	PSHU	Immed	5+	2	66	ROR		6+	2+
07	ASR		6	2	37	PULU	Immed	5+	2	67	ASR		6+	2+
08	ASL, LSL		6	2	38	*	—			68	ASL, LSL		6+	2+
09	ROL		6	2	39	RTS	Inherent	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	Inherent	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	Immed	8	2	4E	*				7E	JMP		4	3
1F	TFR	Immed	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

3

3

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

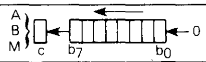
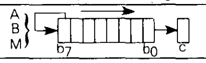
NOTE: All unused opcodes are both undefined and illegal

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 18 — PROGRAMMING AID

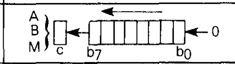
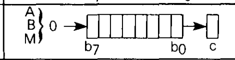
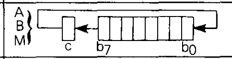
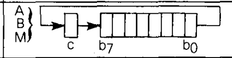
		Addressing Modes																				
		Immediate			Direct			Indexed			Extended			Inherent				5	3	2	1	0
Instruction	Forms	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#	Description	H	N	Z	V	C
ABX														3A	3	1	B+X→X (Unsigned)	•	•	•	•	•
ADC	ADCA	89	2	2	99	4	2	A9	4+	2+	B9	5	3				A+M+C→A	↕	↕	↕	↕	↕
	ADCB	C9	2	2	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 ∧ M→A	•	↕	↕	0	•
	ANDB	C4	2	2	D4	4	2	E4	4+	2+	F4	5	3				B ∧ M→B	•	↕	↕	0	•
	ANDCC	1C	3	2													CC ∧ 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 ∧ A)	•	↕	↕	0	•
	BITB	C5	2	2	D5	4	2	E5	4+	2+	F5	5	3				Bit Test B (M ∧ 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 ∧ 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
- ∧ Logical and
- ⊻ Logical Exclusive or

3

FIGURE 18 — PROGRAMMING AID (CONTINUED)

Instruction	Forms	Immediate Op	Immediate ~	Immediate #	Direct Op	Direct ~	Direct #	Indexed[1] Op	Indexed[1] ~	Indexed[1] #	Extended Op	Extended ~	Extended #	Inherent Op	Inherent ~	Inherent #	Description	5 H	3 N	2 Z	1 V	0 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	•	•	•	•	•
	SWI2[6]													10 3F	20	2	Software Interrupt 2	•	•	•	•	•
	SWI3[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 18 — PROGRAMMING AID (CONTINUED)

Branch Instructions

Instruction	Forms	Addressing Mode Relative OP	~5	#	Description	5 H	3 N	2 Z	1 V	0 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=1	•	•	•	•	•
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	•	•	•	•	•
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.

3

INDEXED ADDRESSING MODES

Type	Forms	Nondirect Assembler Form	Nondirect Post-Byte Opcode	+ ~	+ #	Indirect Assembler Form	Indirect Post-Byte Opcode	+ ~	+ #
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   RR: 00 = X   10 = U
X = Don't Care   01 = Y   11 = S

INDEXED ADDRESSING POSTBYTE REGISTER BIT ASSIGNMENTS

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]

Addressing Mode Field

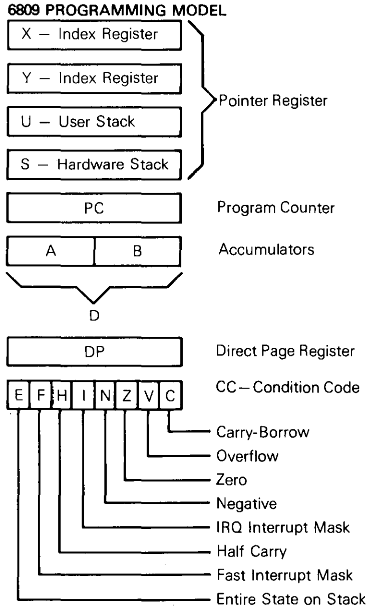
Indirect Field

(Sign bit when $b_7 = 0$)

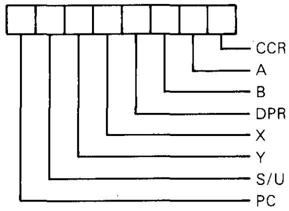
Register Field: RR
00 = X
01 = Y
10 = U
11 = S

X = Don't Care

6809 PROGRAMMING MODEL



Push/Pull Post Byte



Transfer/Exchange Post Byte

Register Field

0000 = D (A-B)	0101 = PC
0001 = X	1000 = A
0010 = Y	1001 = B
0011 = U	1010 = CCR
0100 = S	1011 = DPR

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 SW12
FFF2 SW13
FFF0 Reserved

3

ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Ceramic L Suffix	1.0 MHz	0°C to 70°C	MC6809EL
	1.0 MHz	−40°C to 85°C	MC6809ECL
	1.5 MHz	0°C to 70°C	MC68A09EL
	1.5 MHz	−40°C to 85°C	MC68A09ECL
	2.0 MHz	0°C to 70°C	MC68B09EL
	2.0 MHz	−40°C to 85°C	MC68B09ECL
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6809EP
	1.0 MHz	−40°C to 85°C	MC6809ECP
	1.5 MHz	0°C to 70°C	MC68A09EP
	1.5 MHz	−40°C to 85°C	MC68A09ECP
	2.0 MHz	0°C to 70°C	MC68B09EP
	2.0 MHz	−40°C to 85°C	MC68B09ECP
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6809ES
	1.0 MHz	−40°C to 85°C	MC6809ECS
	1.5 MHz	0°C to 70°C	MC68A09ES
	1.5 MHz	−40°C to 85°C	MC68A09ECS
	2.0 MHz	0°C to 70°C	MC68B09ES
	2.0 MHz	−40°C to 85°C	MC68B09ECS

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

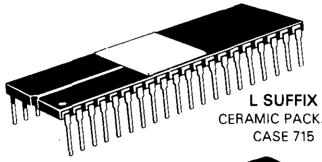
SOFTWARE FEATURES

- 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

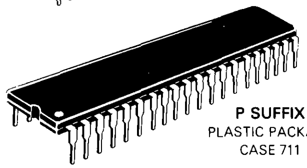
This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

HCMOS
(HIGH-DENSITY CMOS SILICON-GATE)

8-BIT HCMOS MICROPROCESSING UNIT



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	$\overline{\text{HALT}}$
$\overline{\text{NMI}}$	2	39	TSC
$\overline{\text{IRQ}}$	3	38	LIC
$\overline{\text{FIRQ}}$	4	37	$\overline{\text{RESET}}$
BS	5	36	AVMA
BA	6	35	Q
V_{CC}	7	34	E
A0	8	33	BUSY
A1	9	32	R/$\overline{\text{W}}$
A2	10	31	D0
A3	11	30	D1
A4	12	29	D2
A5	13	28	D3
A6	14	27	D4
A7	15	26	D5
A8	16	25	D6
A9	17	24	D7
A10	18	23	A15
A11	19	22	A14
A12	20	21	A13

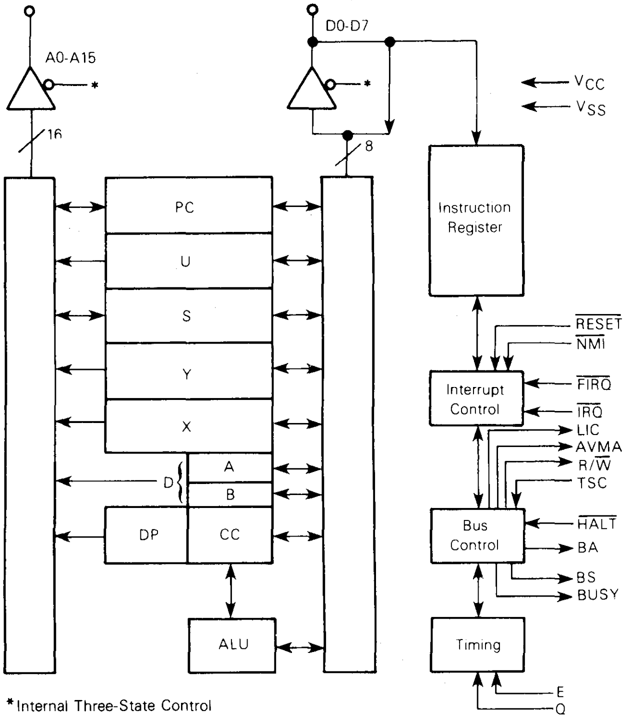
ARCHITECTURAL FEATURES

- 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

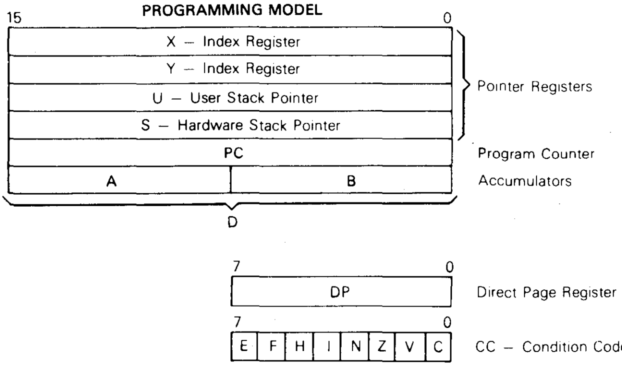
- Hardware — Interfaces with All M6800 Peripherals
- Software — Upward Source Code Compatible Instruction Set and Addressing Modes

EXPANDED BLOCK DIAGRAM

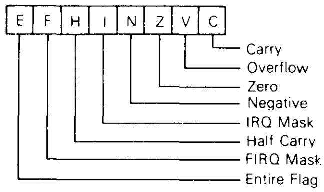


*Internal Three-State Control

PROGRAMMING MODEL



CONDITION CODE REGISTER FORMAT



MOTOROLA

MCM6810

128 × 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.

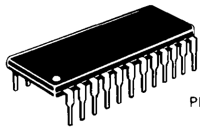
The memory is compatible with the M6800 Microcomputer Family, providing random storage in byte increments. Memory expansion is provided through multiple Chip Select inputs.

- 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

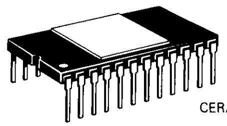
MOS

(N-CHANNEL, SILICON-GATE)

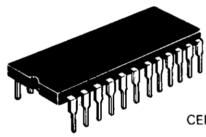
128 × 8-BIT STATIC RANDOM ACCESS MEMORY



P SUFFIX
PLASTIC PACKAGE
CASE 709



L SUFFIX
CERAMIC PACKAGE
CASE 716

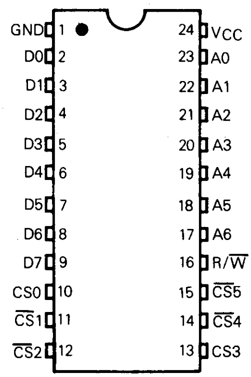


S SUFFIX
CERDIP PACKAGE
CASE 623

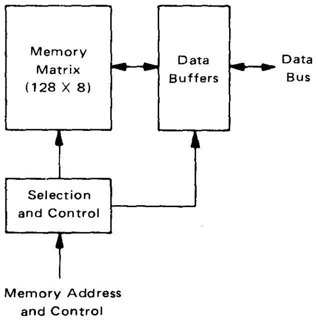
ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MCM6810L
	1.0	−40°C to 85°C	MCM6810CL
	1.5	0°C to 70°C	MCM68A10L
	1.5	−40°C to 85°C	MCM68A10CL
	2.0	0°C to 70°C	MCM68B10L
Plastic P Suffix	1.0	0°C to 70°C	MCM6810P
	1.0	−40°C to 85°C	MCM6810CP
	1.5	0°C to 70°C	MCM68A10P
	1.5	−40°C to 85°C	MCM68A10CP
	2.0	0°C to 70°C	MCM68B10P
Cerdip S Suffix	1.0	0°C to 70°C	MCM6810S
	1.0	−40°C to 85°C	MCM6810CS
	1.5	0°C to 70°C	MCM68A10S
	1.5	−40°C to 85°C	MCM68A10CS
	2.0	0°C to 70°C	MCM68B10S

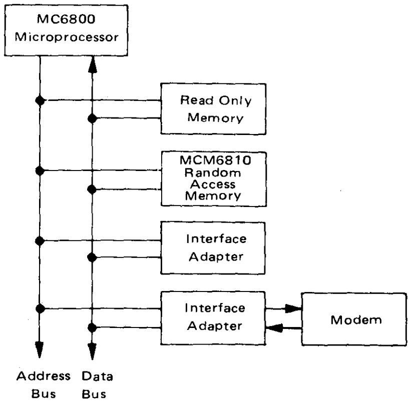
PIN ASSIGNMENT



MCM6810 RANDOM ACCESS MEMORY BLOCK DIAGRAM



M6800 MICROCOMPUTER FAMILY 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 MCM6810, MCM68A10, MCM68B10 MCM6810C, MCM68A10C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages 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 Plastic Cerdip	θ_{JA}	 60 120 65	°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:

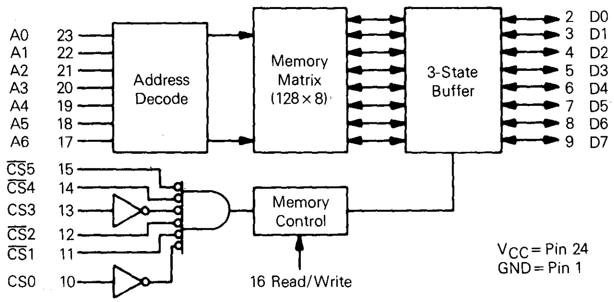
$$P_D = K \div (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives:

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

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.

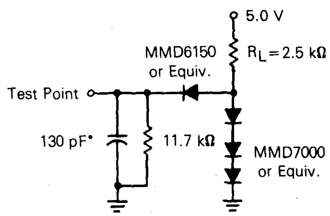
BLOCK DIAGRAM



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	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 Current (A_n, R/$\overline{W}$, $\overline{CS}_n$) (V_{in} = 0 to 5.25 V)	I_{in}	–	2.5	μA
Output High Voltage (I_{OH} = −205 μA)	V_{OH}	2.4	–	V
Output Low Voltage (I_{OL} = 1.6 mA)	V_{OL}	–	0.4	V
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	μA
Supply Current (V_{CC} = 5.25 V, All Other Pins Grounded) 1.0 MHz	I_{CC}	–	80	mA
Supply Current (V_{CC} = 5.25 V, All Other Pins Grounded) 1.5, 2.0 MHz	I_{CC}	–	100	mA
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

AC TEST LOAD



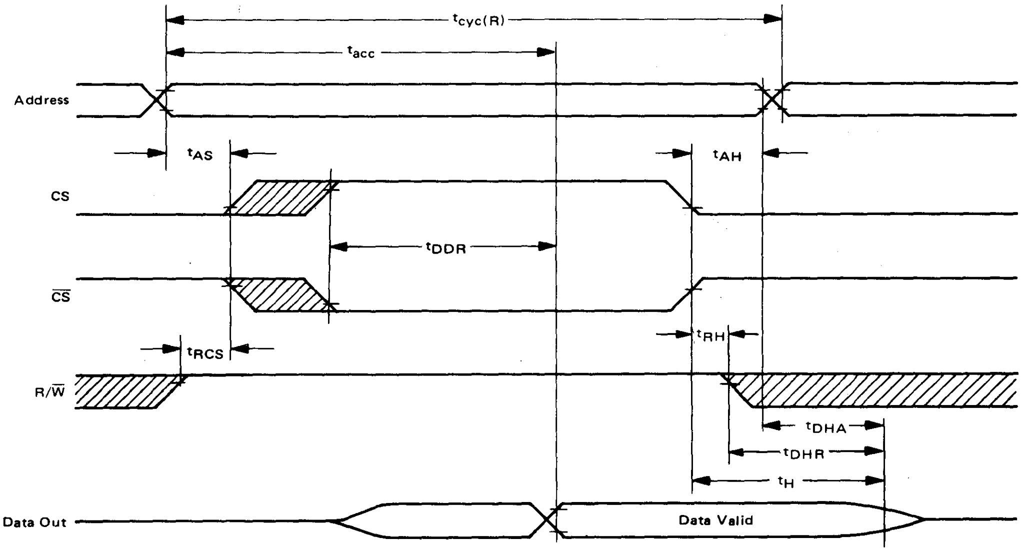
*Includes Jig Capacitance

AC OPERATING CONDITIONS AND CHARACTERISTICS

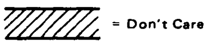
READ CYCLE (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = T_L to T_H unless otherwise noted.)

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



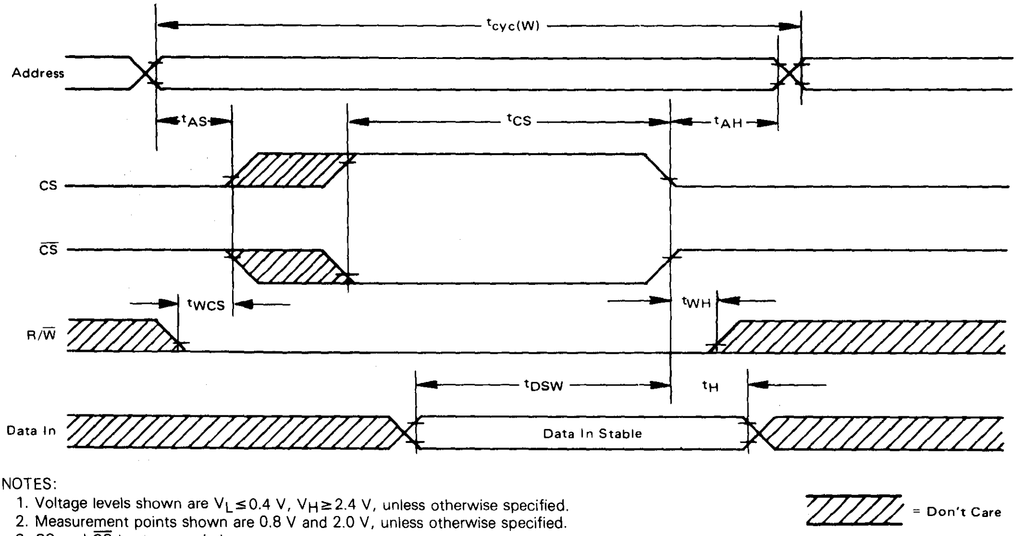
NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.
3. CS and $\overline{CS}$ have same timing.



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	–	0	–	0	–	ns

WRITE CYCLE 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.
3. CS and $\overline{CS}$ have same timing.

MOTOROLA

MC6821

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.

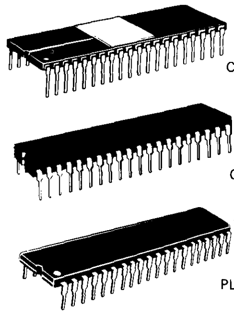
- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- 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

3



L SUFFIX
CERAMIC PACKAGE
CASE 715

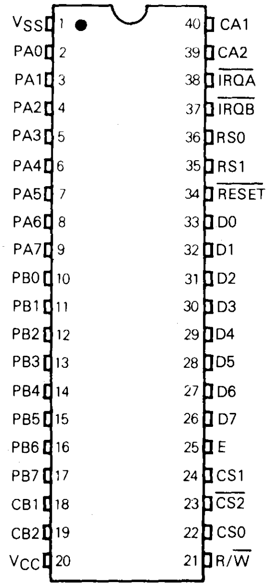
S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6821L
	1.0	−40°C to 85°C	MC6821CL
	1.5	0°C to 70°C	MC68A21L
	1.5	−40°C to 85°C	MC68A21CL
	2.0	0°C to 70°C	MC68B21L
Cerdip S Suffix	1.0	0°C to 70°C	MC6821S
	1.0	−40°C to 85°C	MC6821CS
	1.5	0°C to 70°C	MC68A21S
	1.5	−40°C to 85°C	MC68A21CS
	2.0	0°C to 70°C	MC68B21S
Plastic P Suffix	1.0	0°C to 70°C	MC6821P
	1.0	−40°C to 85°C	MC6821CP
	1.5	0°C to 70°C	MC68A21P
	1.5	−40°C to 85°C	MC68A21CP
	2.0	0°C to 70°C	MC68B21P

PIN ASSIGNMENT



3

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	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 applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $GND \le (V_{in}$ or $V_{out}) \le V_{CC}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either GND 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} = 1.6 mA)	V_{OL}	–	–	V_{SS}+0.4	V
Hi-Z 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
Hi-Z 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

DC ELECTRICAL CHARACTERISTICS (Continued)

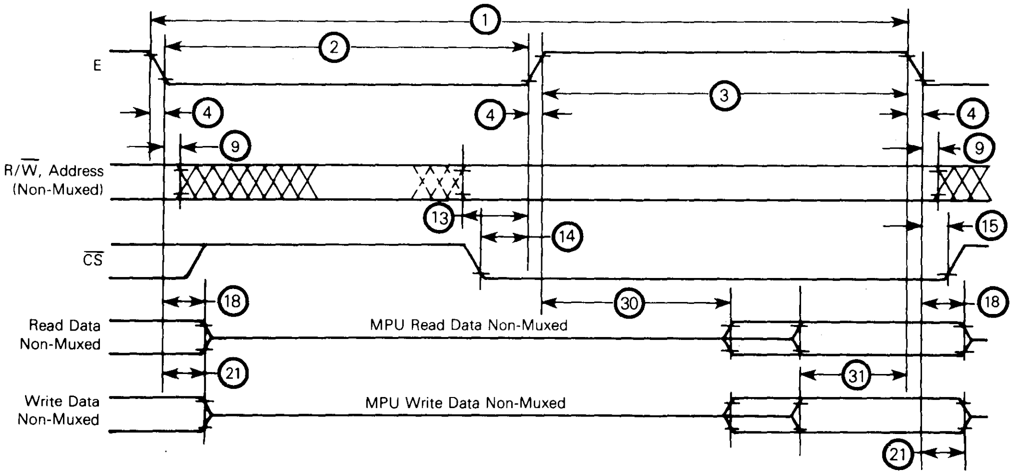
Characteristic		Symbol	Min	Typ	Max	Unit
PERIPHERAL BUS (PA0-PA7, PB0-PB7, CA1, CA2, CB1, CB2)						
Input Leakage Current (V_{in} = 0 to 5.25 V)	R/$\overline{W}$, $\overline{RESET}$, RS0, RS1, CS0, CS1, $\overline{CS2}$, CA1, CB1, Enable	I_{in}	–	1.0	2.5	μA
Hi-Z Input Leakage Current (V_{in} = 0.4 to 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	V_{OH}	V_{SS} + 2.4	–	–	V
Output High Voltage (I_{Load} = –10 μA)	PA0-PA7, CA2	V_{OH}	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_L = 0°C)		P_{INT}	–	–	550	mW

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristic	Symbol	MC6821 Min	MC6821 Max	MC68A21 Min	MC68A21 Max	MC68B21 Min	MC68B21 Max	Unit
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	–	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 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, $\overline{IRQA}$ and $\overline{IRQB}$	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
$\overline{RESET}$ Low Time*	t_{RL}	1.0	–	0.66	–	0.5	–	μs	15

*The $\overline{RESET}$ line must be high a minimum of 1.0 μs before addressing the PIA.

FIGURE 2 – BUS TIMING TEST LOADS

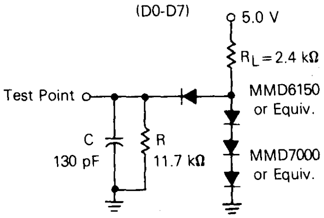
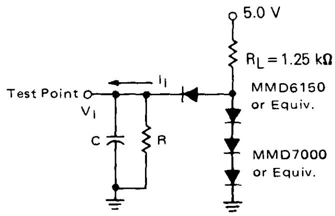


FIGURE 3 – TTL EQUIVALENT TEST LOAD

(PA0–PA7, PB0–PB7, CA2, CB2)



C = 30 pF, R = 12 k

FIGURE 4 – CMOS EQUIVALENT TEST LOAD

(PA0-PA7, CA2)

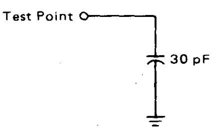


FIGURE 5 – NMOS EQUIVALENT TEST LOAD

($\overline{IRQ}$ Only)

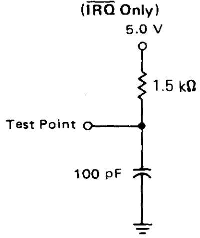


FIGURE 6 — PERIPHERAL DATA SETUP AND HOLD TIMES
(Read Mode)

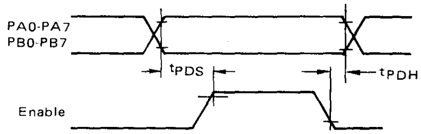
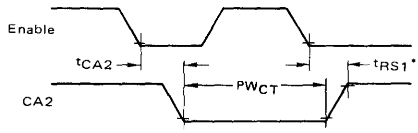


FIGURE 7 — CA2 DELAY TIME
(Read Mode; CRA-5 = CRA3 = 1, CRA-4 = 0)



*Assumes part was deselected during the previous E pulse.

FIGURE 8 — CA2 DELAY TIME
(Read Mode; CRA-5 = 1, CRA-3 = CRA-4 = 0)

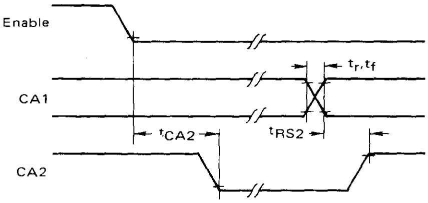


FIGURE 9 — PERIPHERAL CMOS DATA DELAY TIMES
(Write Mode; CRA-5 = CRA-3 = 1, CRA-4 = 0)

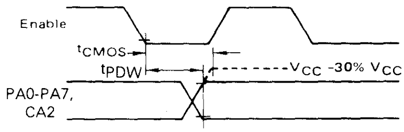
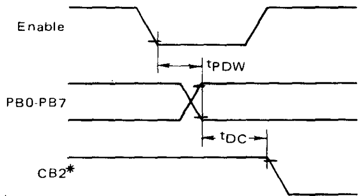
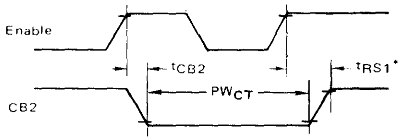


FIGURE 10 — PERIPHERAL DATA AND CB2 DELAY TIMES
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



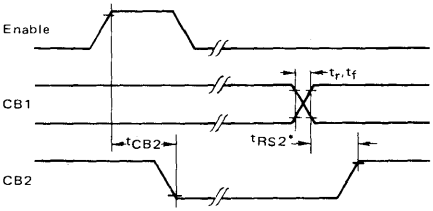
*CB2 goes low as a result of the positive transition of Enable.

FIGURE 11 — CB2 DELAY TIME
(Write Mode; CRB-5 = CRB-3 = 1, CRB-4 = 0)



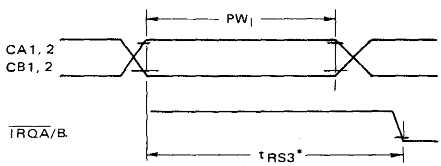
*Assumes part was deselected during the previous E pulse.

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

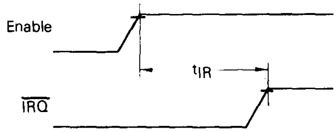
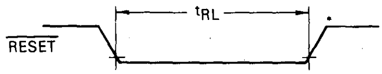


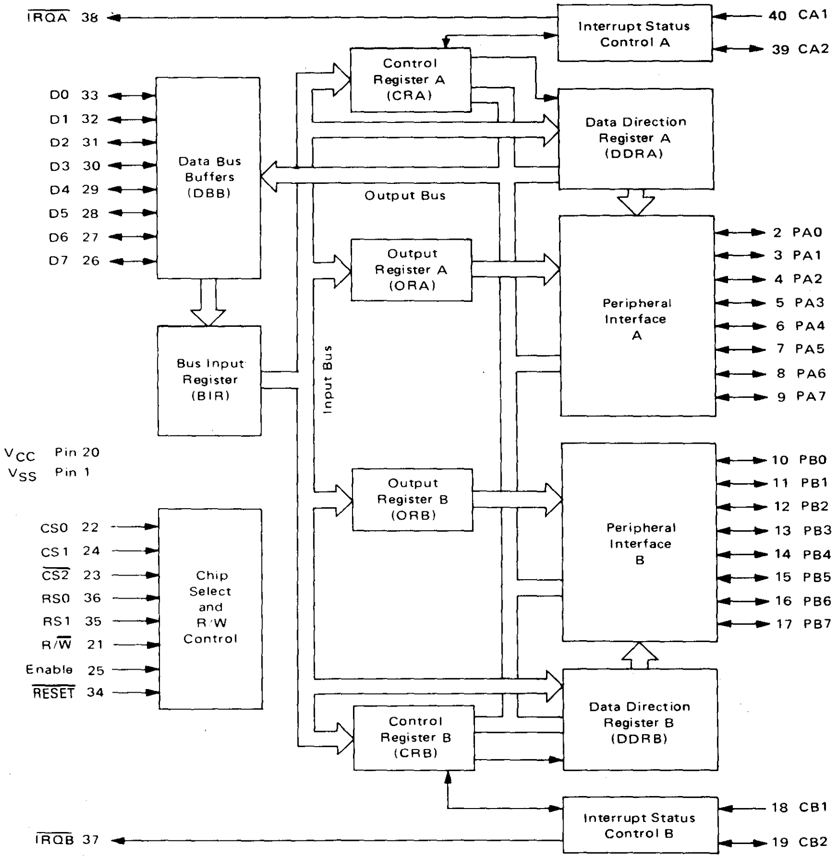
FIGURE 15 — $\overline{RESET}$ LOW TIME



*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/$\overline{\mathrm{W}}$) — 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.

$\overline{\mathrm{RESET}}$ — The active low $\overline{\mathrm{RESET}}$ 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 $\overline{\mathrm{CS2}}$) — These three input signals are used to select the PIA. CS0 and CS1 must be high and $\overline{\mathrm{CS2}}$ 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 ($\overline{\mathrm{IRQA}}$ and $\overline{\mathrm{IRQB}}$) — The active low Interrupt Request lines ($\overline{\mathrm{IRQA}}$ and $\overline{\mathrm{IRQB}}$) 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.

3

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 properly 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 at least 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 $\overline{\text{RESET}}$ 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 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

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 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)

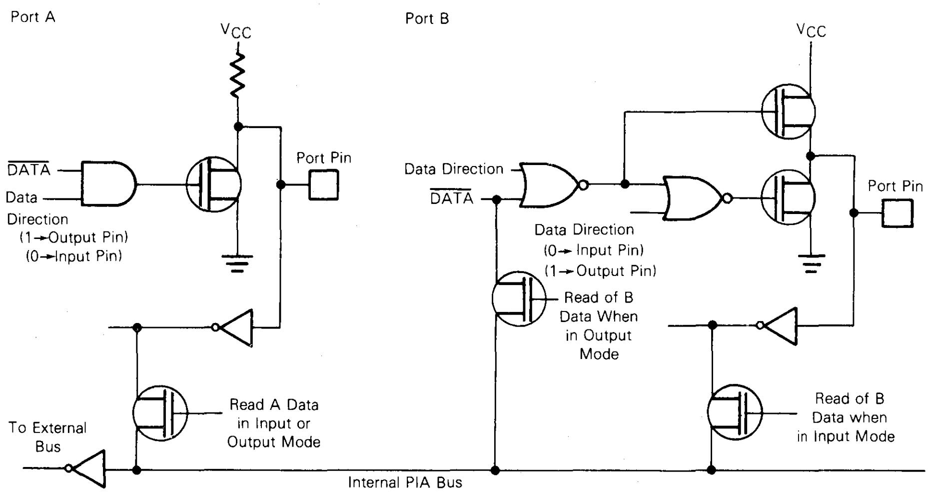
Bit 2, in each Control Register (CRA and CRB), determines selection of either a Peripheral Output Register or the corresponding Data Direction E Register when the proper register select signals are applied to RS0 and RS1. A "1" in bit 2 allows access of the Peripheral Interface Register, while a "0" 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-0, 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



ORDERING INFORMATION

MC68A21CP

- 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.

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(B) 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

Advance Information

INDUSTRIAL INTERFACE ADAPTER (IIA)

The MC6822 Industrial Interface Adapter (IIA) provides a 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 an 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 Circuits
- High Voltage Capability up to 18 Volts
- Program Controlled Interrupt and Interrupt Disable Capability
- Ports Output Compatible with CMOS at 15 Volts
- TTL Compatible
- Static Operation
- Pin Compatible with MC6821 PIA

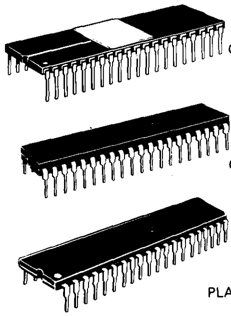
ORDERING INFORMATION

Package Type	Frequency	Operating Temperature	Part Number
Ceramic L Suffix	1.0 MHz	0°C to 70°C	MC6822L
	1.0 MHz	−40°C to 85°C	MC6822CL
	1.5 MHz	0°C to 70°C	MC68A22L
	1.5 MHz	−40°C to 85°C	MC68A22CL
	2.0 MHz	0°C to 70°C	MC68B22L
	2.0 MHz	−40°C to 85°C	MC68B22CL
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6822S
	1.0 MHz	−40°C to 85°C	MC6822CS
	1.5 MHz	0°C to 70°C	MC68A22S
	1.5 MHz	−40°C to 85°C	MC68A22CS
	2.0 MHz	0°C to 70°C	MC68B22S
	2.0 MHz	−40°C to 85°C	MC68B22CS
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6822P
	1.0 MHz	−40°C to 85°C	MC6822CP
	1.5 MHz	0°C to 70°C	MC68A22P
	1.5 MHz	−40°C to 85°C	MC68A22CP
	2.0 MHz	0°C to 70°C	MC68B22P
	2.0 MHz	−40°C to 85°C	MC68B22CP

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

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

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	CA1
PA0	2	39	CA2
PA1	3	38	$\overline{IRQA}$
PA2	4	37	$\overline{IRQB}$
PA3	5	36	RS0
PA4	6	35	RS1
PA5	7	34	$\overline{RESET}$
PA6	8	33	D0
PA7	9	32	D1
PB0	10	31	D2
PB1	11	30	D3
PB2	12	29	D4
PB3	13	28	D5
PB4	14	27	D6
PB5	15	26	D7
PB6	16	25	E
PB7	17	24	CS1
CB1	18	23	$\overline{CS2}$
CB2	19	22	CS0
V_{CC}	20	21	R/$\overline{W}$

3

MAXIMUM RATINGS

Characteristic	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to 7.0	V
Input Voltage PA0-PA7, CA1, CA2, PB0-PB7, CB1, CB2 All Others	V_{in}	 −0.3 to 18.0 −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

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}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Cerdip	θ_{JA}	 50 100 60	°C/W

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C
$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W
$P_D \equiv P_{INT} + P_{PORT}$
$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{PORT} \equiv$ Port Power Dissipation, Watts — User Determined

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

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

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

Solving equations 1 and 2 for K gives:

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

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

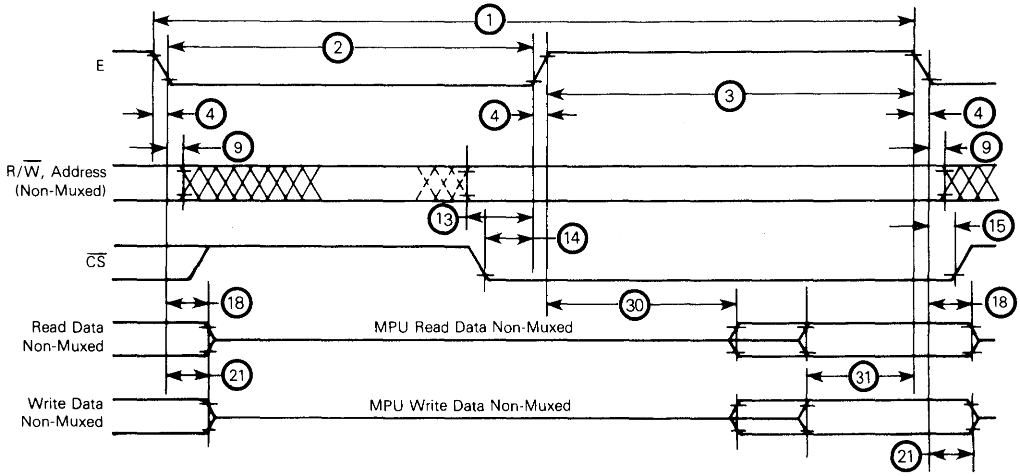
DC ELECTRICAL CHARACTERISTICS (V_{CC}=5.0 Vdc ±5%, V_{SS}=0, T_A=T_L to T_H, unless otherwise noted)

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}=1.6 mA)	V_{OL}	–	–	$V_{SS}+0.4$	V
Hi-Z 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
Hi-Z 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)					
Port Leakage High Current (V_{in}=16 V)	I_{PLKH}	–	–	10	μA
Port Leakage Low Current (V_{in}=10 V)	I_{PLKL}	–	–	2.5	μA
Output Low Voltage (I_{Load}=1 mA)	V_{OL}	–	–	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	MC6822 Min	MC6822 Max	MC68A22 Min	MC68A22 Max	MC68B22 Min	MC68B22 Max	Unit
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	–	20	–	20	–	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

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 noted)

Characteristic	Symbol	MC6822 Min	MC6822 Max	MC68A22 Min	MC68A22 Max	MC68B22 Min	MC68B22 Max	Unit	Reference Fig. No.
Data Setup Time	t_{PDS}	200	–	135	–	100	–	ns	4
Data Hold Time	t_{PDH}	0	–	0	–	0	–	ns	4
Delay Time, Enable Negative Transition to CA2 Negative Transition	t_{CA2}	–	1.0	–	0.670	–	0.500	µs	1, 5, 6
Delay Time, Enable Negative Transition to CA2 Positive Transition	t_{RS1}	–	1.0	–	0.670	–	0.500	µs	1, 5
Rise and Fall Times for CA1 and CA2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µs	6
Delay Time from CA1 Active Transition to CA2 Postive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	1, 6
Delay Time, Enable Negative Transition to Data Valid	t_{PDW}	–	1.0	–	0.670	–	0.5	µs	1, 7, 8
Delay Time, Enable Positive Transition to CB2 Negative Transition	t_{CB2}	–	1.0	–	0.670	–	0.5	µs	1, 9, 10
Delay Time, Data Valid to CB2 Negative Transition	t_{DC}	20	–	20	–	20	–	ns	1, 8
Delay Time, Enable Positive Transition to CB2 Positive Transition	t_{RS1}	–	1.0	–	0.670	–	0.5	µs	1, 9
Control Output Pulse Width, CA2/CB2	PW_{CT}	550	–	200	–	50	–	ns	1, 9
Rise and Fall Time for CB1 and CB2 Input Signals	t_r, t_f	–	1.0	–	1.0	–	1.0	µs	10
Delay Time, CB1 Active Transition to CB2 Positive Transition	t_{RS2}	–	2.0	–	1.35	–	1.0	µs	1, 10
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	t_{IR}	–	1.60	–	1.10	–	0.85	µs	3, 12
Interrupt Response Time	t_{RS3}	–	1.0	–	0.8	–	0.6	µs	3, 11
Interrupt Input Pulse Time	PW_I	500	–	330	–	250	–	ns	11
$\overline{RESET}$ Low Time*	t_{RL}	1.0	–	0.66	–	0.5	–	µs	13

* The $\overline{RESET}$ line must be high a minimum of 1.0 µs before addressing the IIA.

FIGURE 1 — BUS TIMING TEST LOADS

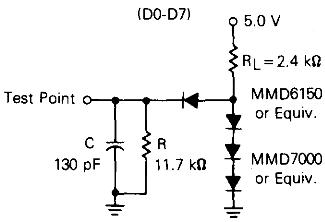


FIGURE 2 — TEST LOADS FOR PA0-PA7, PB0-PB7, CA2, CB2

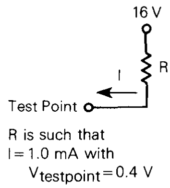


FIGURE 3 — NMOS EQUIVALENT TEST LOAD

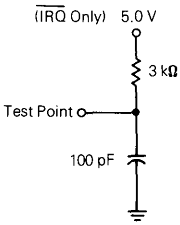
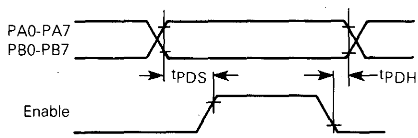


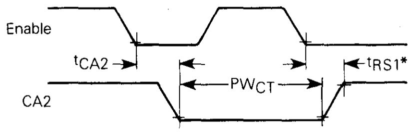
FIGURE 4 — PERIPHERAL DATA SETUP AND HOLD TIMES (READ MODE)



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.

3

FIGURE 5 — CA2 DELAY TIME
(READ MODE: CRA-5 = CRA-3 = 1,
CRA-4 = 0)



*Assumes part was deselected during the previous E pulse.

FIGURE 6 — CA2 DELAY TIME
(READ MODE: CRA-5 = 1,
CRA-3 = CRA-4 = 0)

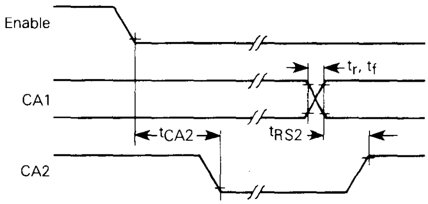


FIGURE 7 — PERIPHERAL CMOS DATA
DELAY TIMES
(WRITE MODE: CRA-5 = CRA-3 = 1,
CRA-4 = 0)

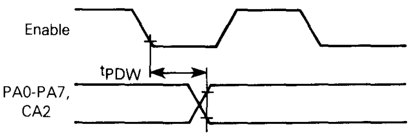
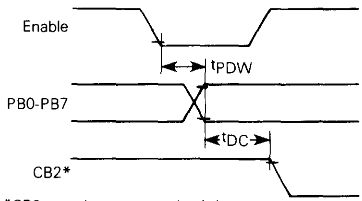
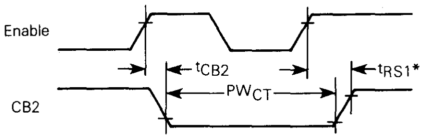


FIGURE 8 — PERIPHERAL DATA AND
CB2 DELAY TIMES
(WRITE MODE: CRB-5 = CRB-3 = 1,
CRB-4 = 0)



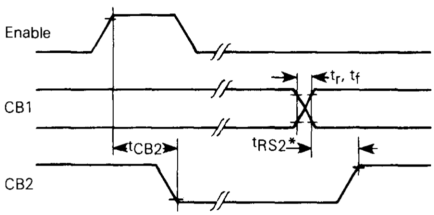
*CB2 goes low as a result of the positive transition of enable.

FIGURE 9 — CB2 DELAY TIME
(WRITE MODE: CRB-5 = CRB-3 = 1,
CRB-4 = 0)



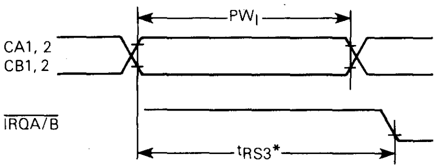
*Assumes part was deselected during the previous E pulse.

FIGURE 10 — CB2 DELAY TIME
(WRITE MODE: CRB-5 = 1,
CRB-3 = CRB-4 = 0)



*Assumes part was deselected during any previous E pulse.

FIGURE 11 — INTERRUPT PULSE WIDTH
AND $\overline{IRQ}$ RESPONSE



*Assumes interrupt enable bits are set.

FIGURE 12 — $\overline{IRQ}$ RELEASE TIME

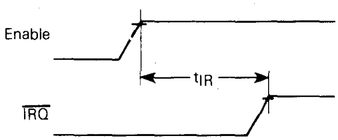
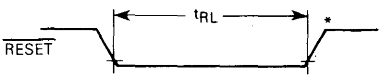


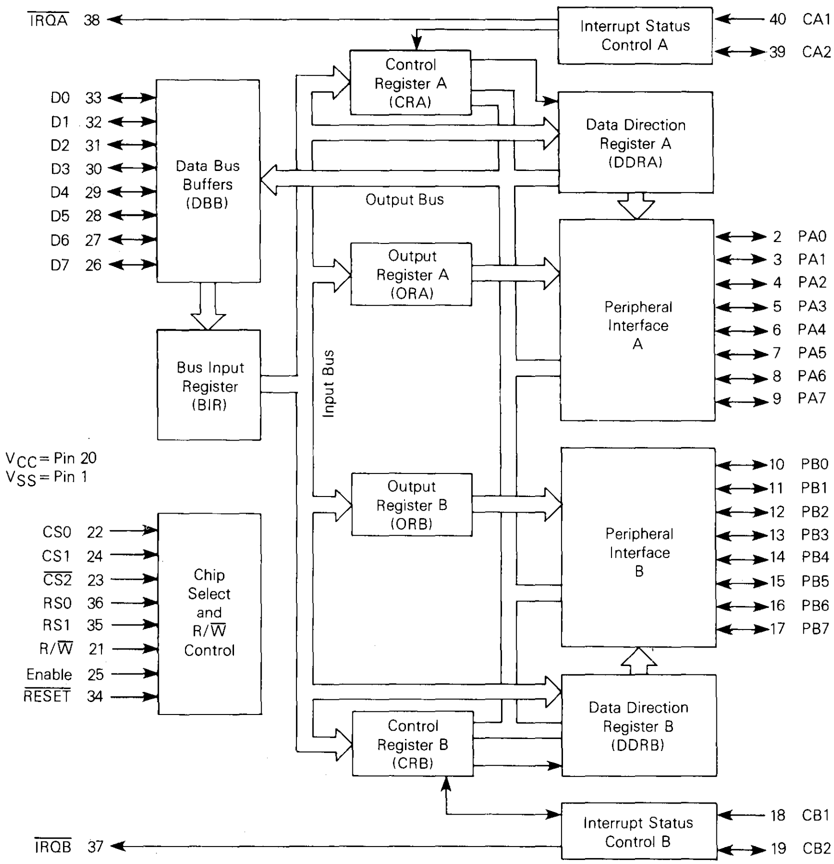
FIGURE 13 — $\overline{RESET}$ LOW TIME



*The $\overline{RESET}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the IIA.

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 — EXPANDED BLOCK DIAGRAM



IIA 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 IIA. The data bus output drivers are three-state devices that remain in the high-impedance (off) state except when the MPU performs an IIA 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 IIA. 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 IIA for a transfer of data to the MPU data bus. The IIA output buffers are enabled when the proper address and the enable pulse, E, are present.

RESET ($\overline{\text{RESET}}$)

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 IIA 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.

IIA 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 an open-drain output. This is accomplished by setting a one in the corresponding data direction register bit for those lines which are to be outputs. A zero 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.

The data in output register A will appear on the data lines that are programmed to be outputs. A logical one written into the register will cause the corresponding data line to go into a high-impedance state, and may be pulled up externally to a maximum of 18 volts. A logical zero written into the register results in a low on the corresponding data line. 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 properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic one output and less than 0.8 volts for a logic zero output.

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 manner similar 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 volts 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 an open-drain output. 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. This line is programmed by control register B.

INTERNAL CONTROLS

INITIALIZATION

A $\overline{\text{RESET}}$ has the effect of zeroing all IIA registers. This will set PA0-PA7, PB0-PB7, CA2, and CB2 as inputs, and disable all interrupts. The IIA 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 registers, as shown in Table 1.

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

Details of possible configurations of the data direction and control register are given in the following paragraphs.

PORT A-B HARDWARE CHARACTERISTICS

As shown in Figure 15, the MC6822 has a pair of I/O ports whose characteristics differ slightly.

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 16.

DATA DIRECTION ACCESS CONTROL BIT (CRA-2 AND CRB-2)

Bit 2 of each control register (CRA and CRB) determines selection of either a peripheral output register or the corresponding data direction registers when the proper register select signals are applied to RS0 and RS1. A one in bit 2 allows access of the peripheral data register, while a zero causes the data direction register to be addressed.

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.

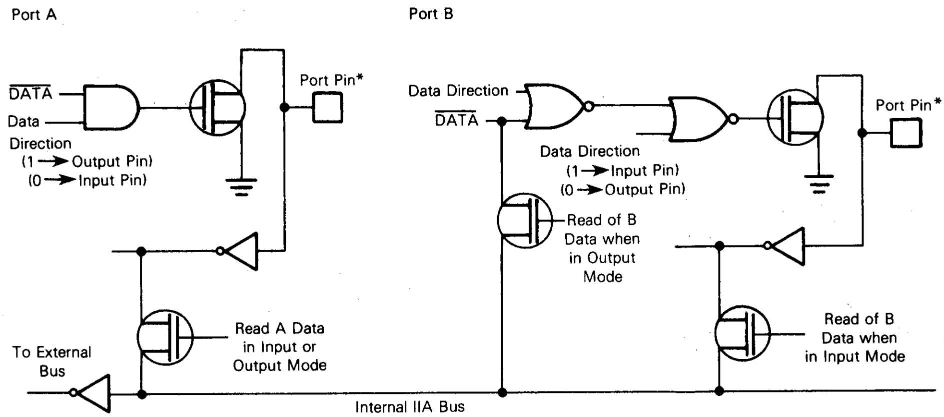
CONTROL OF CA1 AND CB1 INTERRUPT INPUT LINES (CRA-0, CRA-1, CRB-0, AND CRB-1)

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.

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.

FIGURE 15 — PORT A AND PORT B EQUIVALENT CIRCUITS



*Port pins are open drain and must be pulled up externally.

FIGURE 16 — 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(B) 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).

3

MOTOROLA

MC6829

Advance Information

MEMORY MANAGEMENT UNIT

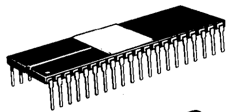
The principle function of the MC6829 Memory Management Unit (MMU) is to expand the address space of the MC6809 from 64K bytes to a maximum of 2 Megabytes. Each MMU is capable of handling four different concurrent tasks including DMA. The MMU can also protect the address space of one task from modification 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 multiprocess 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

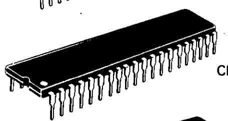
HMOS

(HIGH DENSITY N-CHANNEL, SILICON-GATE)

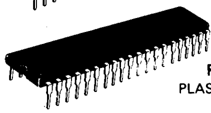
MEMORY MANAGEMENT UNIT (MMU)



L SUFFIX
CERAMIC PACKAGE
CASE 715

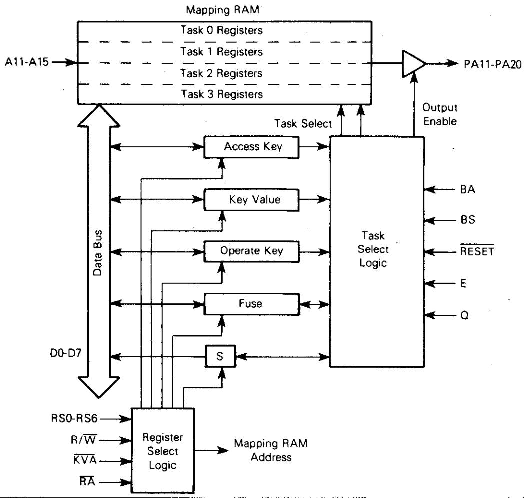


S SUFFIX
CERDIP PACKAGE
CASE 734

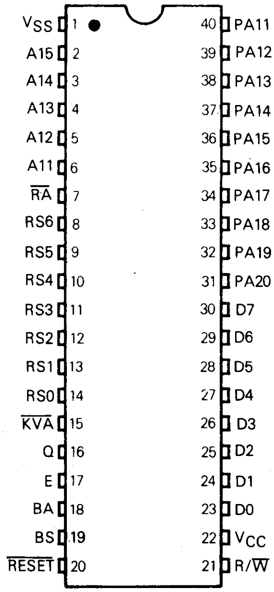


P SUFFIX
PLASTIC PACKAGE
CASE 711

BLOCK DIAGRAM



PIN ASSIGNMENT



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

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:

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

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)

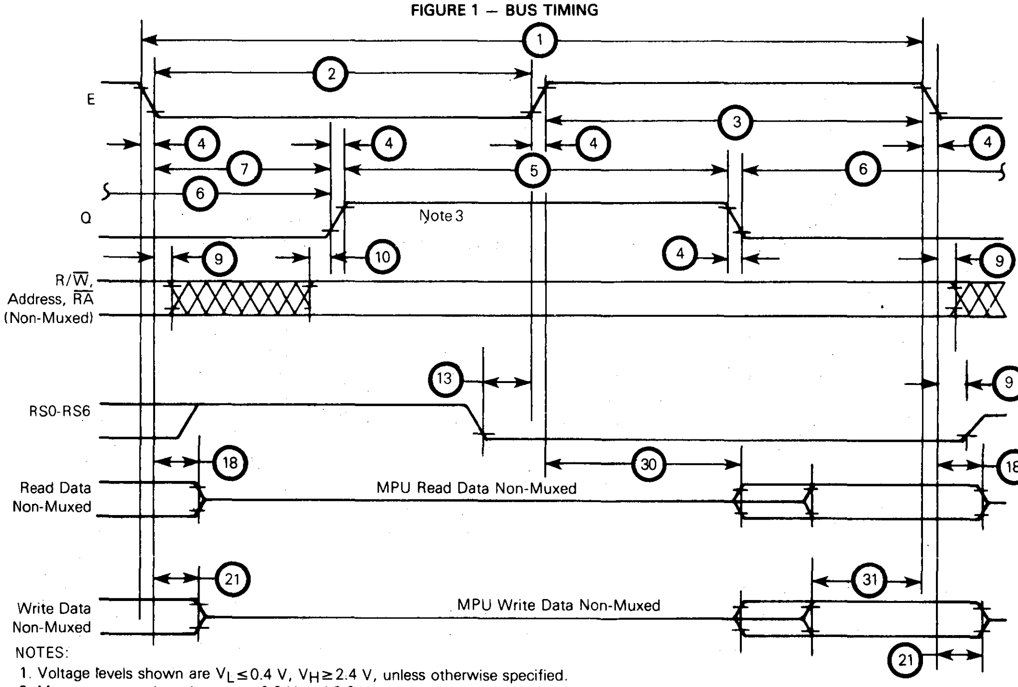
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	$V_{SS}+2.0$	–	V_{CC}	V
Input Low Voltage	All Inputs	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	V_{CC} = Max	I_{in}	–	1.0	2.5	μA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4 V)	D0-D7	I_{IZ}	–	2.0	10	μA
Output High Voltage (I_{Load} = −205 μA, V_{CC} = Min) (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} = Min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Internal Power Dissipation (Measured at T_A = 0°C)		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 Min	MC6829 Max	MC68A29 Min	MC68A29 Max	MC68B29 Min	MC68B29 Max	Unit
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 (RS0-RS6)	t_{AS}	80	–	60	–	40	–	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 Figures 2 and 3	Hi-Z Address Delay	t_{TAD}	–	90	–	80	–	60	ns
See Figure 2	Mapped Address Delay	t_{MAD}	–	200	–	145	–	110	ns

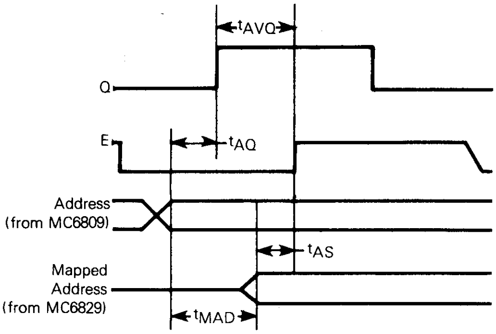
*At specified cycle time.
†The data bus output buffers are no longer sourcing or sinking current by t_{DHR} max. (High Impedance)

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.
3. Depends on speed and bus structure (see bus timing example).

Bus Timing Calculation Example:



1 MHz Case:

t_{AVQ} (Q to E rise delay time) = 250 ns (max)
t_{AQ} (address setup time before Q from MC6809) = 50 ns (min)
t_{MAD} (mapped address delay) = 200 ns (max)
t_{AS} (address setup time before E for peripheral) = 80 ns

Then, the mapped address setup time before $E = t_{AVQ} + t_{AQ} - t_{MAD} = 100$ ns which means $(100 - t_{AS}) = 20$ ns is allowed for address buffering. More buffer time can be achieved by using 1.5 MHz peripheral or 1.5 MHz MC6829.

1.5 MHz Case:

$t_{AVQ} = 165$ ns (max)
$t_{AQ} = 25$ ns (min)
$t_{MAD} = 145$ ns (max)
$t_{AS} = 60$ ns (min)

The mapped address setup time before $E = t_{AVQ} + t_{AQ} - t_{MAD} = 45$ ns which is less than the required setup time for peripheral. Two solutions can be found as following:

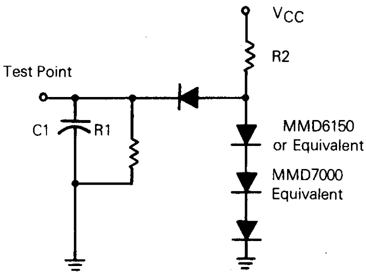
1. If using 2 MHz peripherals, then $t_{AS} = 40$ ns. It will be good for a non-buffered system.
2. If using 2 MHz MC68B29, then $t_{MAD} = 110$ ns. There will be a 20 ns system address buffer time for using 1.5 MHz peripherals and 40 ns for using 2 MHz peripherals.

2 MHz Case:

$t_{AVQ} = 125$ ns (max)
$t_{AQ} = 15$ ns (min)
$t_{MAD} = 110$ ns (max)
$t_{AS} = 40$ ns (min)

The mapped address setup time before $E = t_{AVQ} + t_{AQ} - t_{MAD} = 30$ ns which is less than the 40 ns that a peripheral required. A clock stretch is needed for peripheral access using mapped address in 2 MHz system. However, it can still access the memory devices at 2 MHz bus speed.

LOAD A (D0-D7, PA11-PA20)



R1 = 1.7 k for D0-D7
R1 = 16.5 k for PA11-PA20
R2 = 2.2 k
C1 = 82 pF for D0-D7
C1 = 100 pF for PA11-PA20

FIGURE 2 — MAP SWITCHING, ADDRESS MAPPING

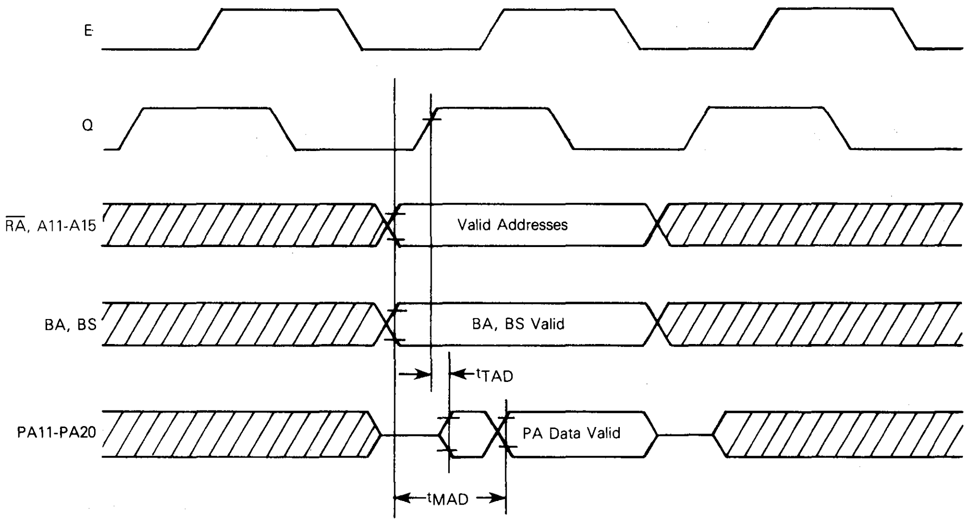
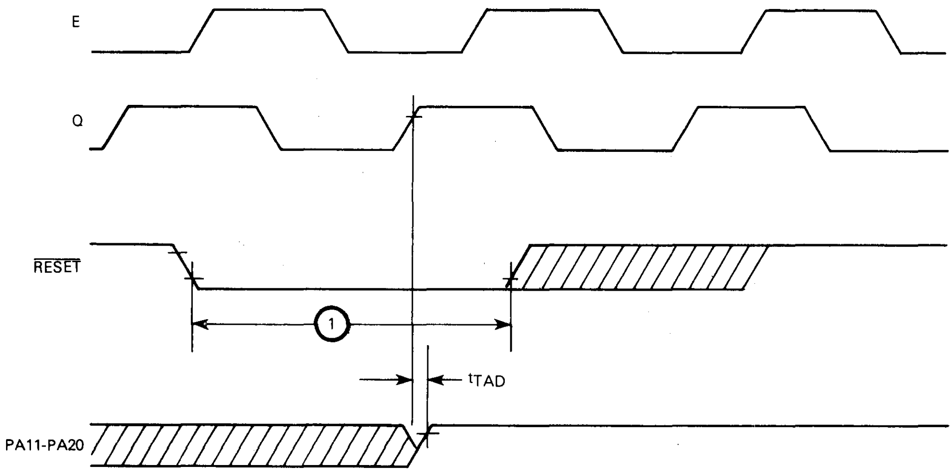


FIGURE 3 — RESET TIMING



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/$\overline{\text{W}}$ — 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.

A11-A15 — Logical Address Lines (Input to 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 TTL loads and a maximum of 90 pF.

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.

$\overline{\text{RA}}$ — 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.

$\overline{\text{KVA}}$ — 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, $\overline{\text{RA}}$ = 0 (asserted), RS6-RS0 are within the range $40-$47 and $\overline{\text{KVA}}$ = 0 (also asserted). Writing the Key Value Register has the additional requirement of having the S-bit set.

$\overline{\text{RESET}}$ — $\overline{\text{RESET}}$ (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. $\overline{\text{RESET}}$ 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 bytes 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 ($\overline{\text{RA}}$) signal which is similar to a chip select line. The $\overline{\text{RA}}$ signal will normally be derived from processor address lines A7-A10 using a simple 4-input gate. For example, a 4-input NOR gate would place the MMU registers at $F800 to $F87F. In systems using DMA, the $\overline{\text{RA}}$ 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 ($\overline{\text{RA}}$) 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.

3

*Refer to Register Select Truth Table for exact procedure to change this register.

FIGURE 4 — LOGIC-TO-PHYSICAL ADDRESS TRANSLATION DIAGRAM

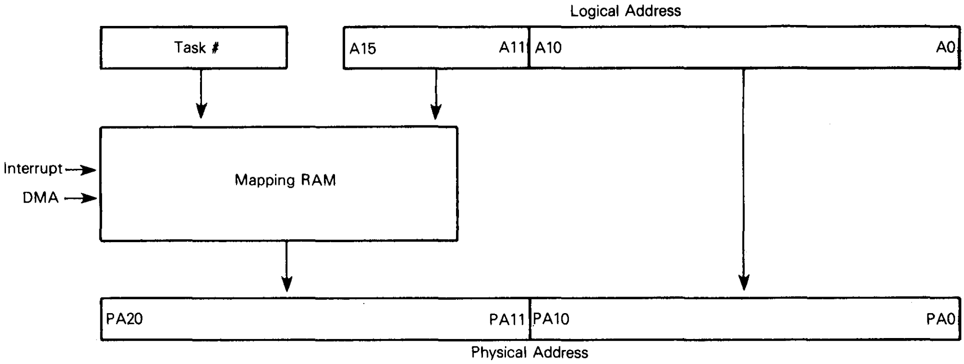


FIGURE 5 — MMU REGISTER MODEL

	Register	D7	D6	D5	D4	D3	D2	D1	D0	Logical Address
	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	
Access	04							PA20	PA19	$1000-$17FF
Key	05	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	
"Window"					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					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 logical 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 $\overline{\text{KVA}}$ input is low. In systems using a single MMU, the $\overline{\text{KVA}}$ 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 ($\overline{\text{KVA}}$) 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.

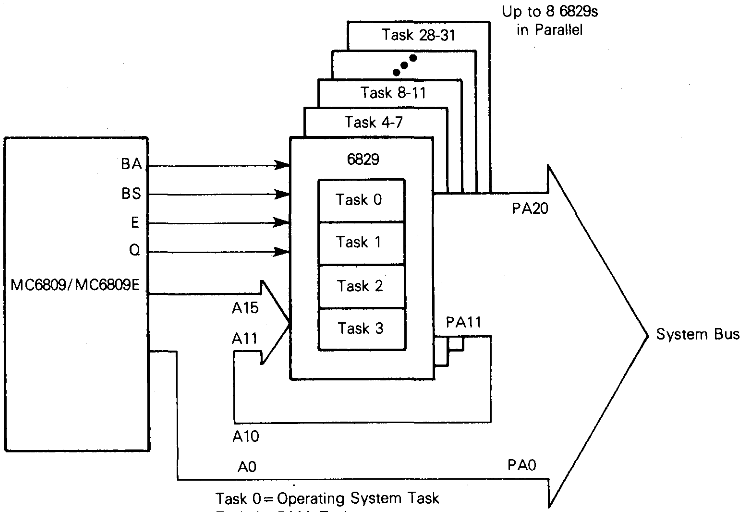
TABLE 1 — REGISTER SELECT TRUTH TABLE

$\overline{\text{RA}}$	R/$\overline{\text{W}}$	$\overline{\text{KVA}}$	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

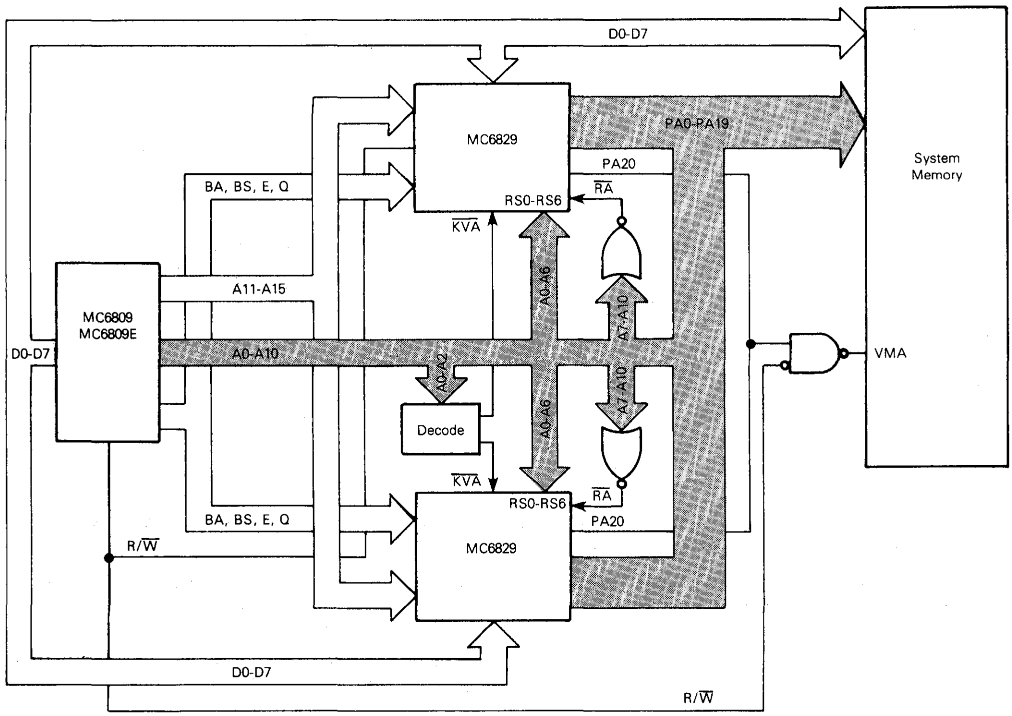
Notes:

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/$\overline{\text{W}}$).
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



3

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

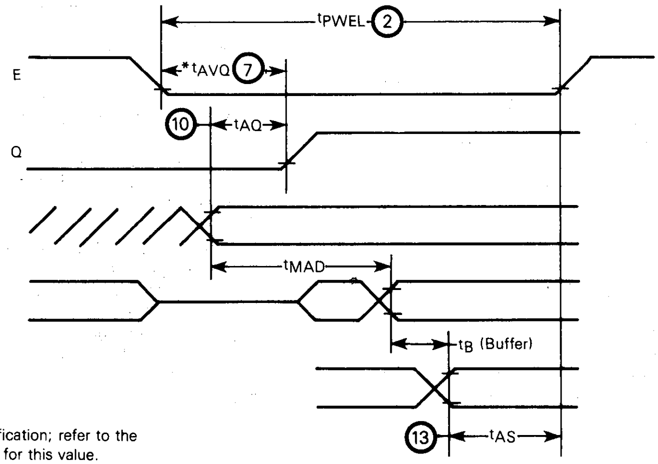
This equation must be satisfied:

$$t_{PWEL} \geq t_{AVQ} - t_{AQ} + t_{MAD} + t_B + t_{AS}$$

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
```

3

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.

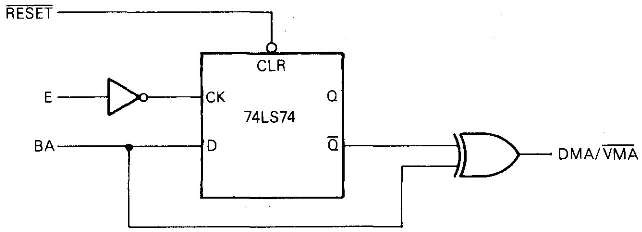
The primary benefit of this scheme is to allow the MC6809 stack to grow dynamically. When a task starts to run, the stack could be initialized to $FFFF with no real memory at that location. When the task did its first subroutine call or stack push, the FIFO interrupt would catch the information and the 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)
MMUO	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 —

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.

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6829L
	1.0	−40°C to 85°C	MC6829CL
	1.5	0°C to 70°C	MC68A29L
	1.5	−40°C to 85°C	MC68A29CL
	2.0	0°C to 70°C	MC68B29L
Cerdip S Suffix	1.0	0°C to 70°C	MC6829S
	1.0	−40°C to 85°C	MC6829CS
	1.5	0°C to 70°C	MC68A29S
	1.5	−40°C to 85°C	MC68A29CS
	2.0	0°C to 70°C	MC68B29S
Plastic P Suffix	1.0	0°C to 70°C	MC6829P
	1.0	−40°C to 85°C	MC6829CP
	1.5	0°C to 70°C	MC68A29P
	1.5	−40°C to 85°C	MC68A29CP
	2.0	0°C to 70°C	MC68B29P

MOTOROLA

MCM68HC34

Advance Information

DUAL-PORT RAM MEMORY UNIT

The MCM68HC34 is a dual-port RAM memory (DPM) unit which enables two processors, arbitrarily referred to as "A" and "B", operating on two separate buses to exchange data without interfering with devices on the other bus. It contains 256 bytes of dual-port RAM which is the medium actually used for the interchange of data.

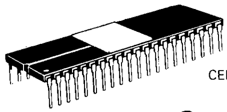
The dual-port memory unit contains six semaphore registers that provide a means for controlling access to the dual-port RAM or any other shared resources. It also contains interrupt registers which provide a means for the processors to interrupt each other.

- High-Speed CMOS (HCMOS) Structure
- Six Read/Write Semaphore Registers
- 256 Bytes of Dual-Port RAM
- Eight Address Lines

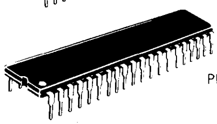
HCMOS

(HIGH DENSITY CMOS SILICON-GATE)

DUAL-PORT RAM MEMORY UNIT



L SUFFIX
CERAMIC PACKAGE
CASE 715



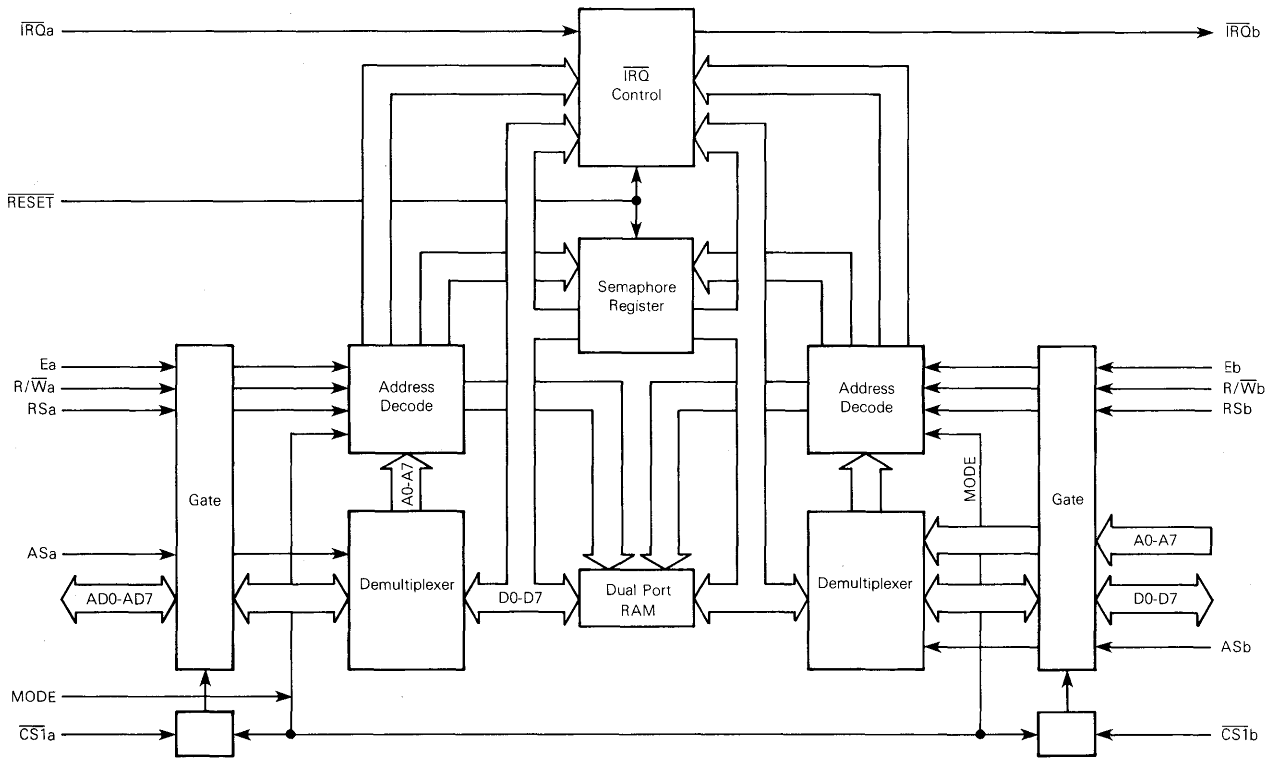
P SUFFIX
PLASTIC PACKAGE
CASE 711

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{CC}	1	40	$\overline{\text{CS1}}$b
$\overline{\text{RESET}}$	2	39	Eb
$\overline{\text{CS1}}$a	3	38	RSb
Ea	4	37	R/$\overline{\text{W}}$b
R/$\overline{\text{W}}$a	5	36	ASb
RSa	6	35	A0
ASa	7	34	A1
MODE	8	33	A2
AD0	9	32	A3
AD1	10	31	A4
AD2	11	30	A5
AD3	12	29	A6
AD4	13	28	A7
AD5	14	27	D7
AD6	15	26	D6
AD7	16	25	D5
$\overline{\text{IRQ}}$a	17	24	D4
V_{SS}	18	23	D3
$\overline{\text{IRQ}}$b	19	22	D2
D0	20	21	D1

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

FIGURE 1 — BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to 7.0	V
Input Voltage, All Inputs	V_{in}	$V_{SS}-0.3$ to $V_{CC}+0.5$	V
Operating Temperature	T_A	0 to 70	°C
Storage Temperature	T_{stg}	−55 to 150	°C

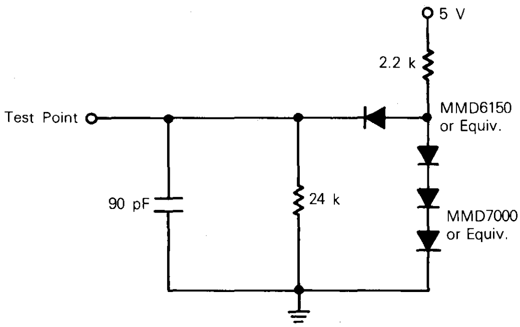
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. Unused inputs must be tied to an appropriate logic level (either V_{CC} or V_{SS}) to reduce leakage currents and increase reliablity.

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Plastic		100	

3

FIGURE 2 — BUS TIMING LOAD



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

Characteristics	Symbol	Min	Max	Unit
Input High Voltage (see Note 1)	V_{IH}	2.0	$V_{CC}+0.3$	V
Input Low Voltage (see Note 2)	V_{IL}	$V_{SS}-0.3$	0.8	V
Input Current (V_{in} = 0 to V_{CC})	I_{in}	–	1.0	μA
Output Leakage Current	I_{OZ}	–	10.0	μA
Output High Voltage (I_{Load} = −100 μA)	V_{OH}	2.4	–	V
(I_{Load} = < 10.0 μA)		$V_{CC}-0.1$	–	
Output Low Voltage (I_{Load} = 1.6 mA)	V_{OL}	–	0.4	V
(I_{Load} = < 10.0 μA)		–	0.1	
Current Drain — Outputs Unloaded				
Standby — CEa and CEb at V_{SS}	I_{DDS}	–	0.1	mA
Operating — Ea, Eb = 1 MHz, Both Sides Active	I_{DD}	–	30	mA
Input Capacitance	C_{in}	–	10	pF
Output Capacitance (AD0-AD7 and D0-D7)	C_{out}	–	12	pF

NOTES:
1. Input high voltage as stated is for all inputs except MODE. In the case of MODE, input high voltage is tied to V_{CC}.
2. Input low voltage as stated is for all inputs except MODE. In the case of MODE, input low voltage is tied to V_{SS} or is floating. If floating, the voltage will be internally pulled to V_{SS}.

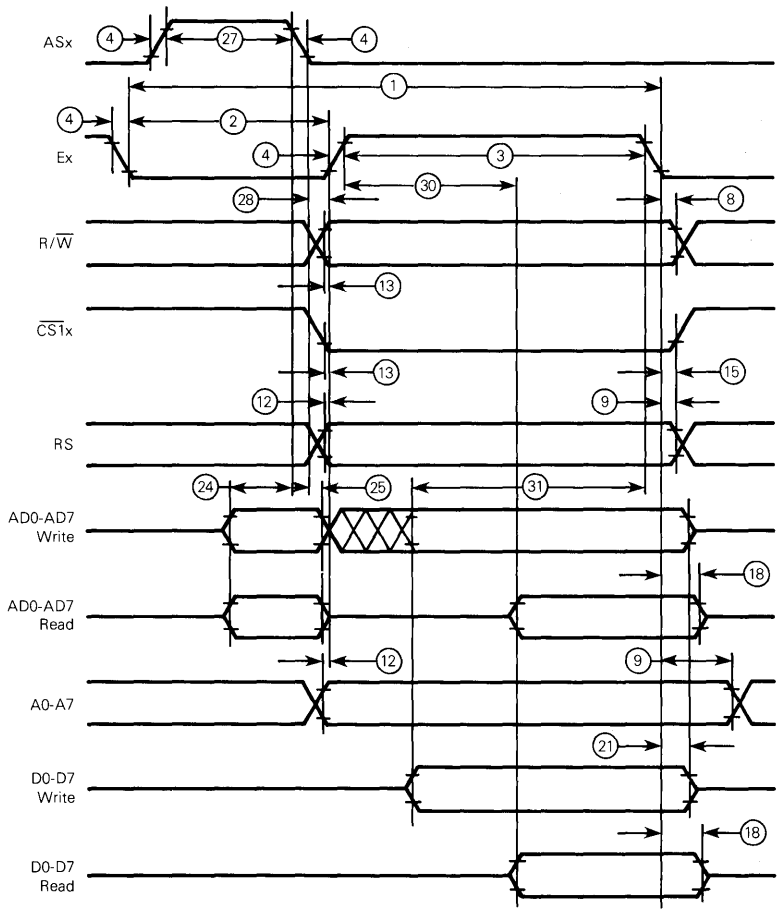
BUS TIMING (See Notes 1 and 2 and Figure 2)

Ident Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	800	–	ns
2	Pulse Width, E Low	P_{WEL}	300	–	ns
3	Pulse Width, E High	P_{WEH}	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	30	ns
8	Read/Write Hold Time	t_{RWH}	10	–	ns
9	Non-Multiplexed Address, RS Hold Time	t_{AH}	10	–	ns
12	Non-Multiplexed Address, RS Valid Time to Eb	t_{AV}	20	–	ns
13	R/$\overline{W}$, Chip Select Setup Time	t_{RWS}	20	–	ns
15	Chip Select Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	20	75	ns
21	Write Data Hold Time	t_{DHW}	10	–	ns
24	Address Setup Time for Latch	t_{ASL}	20	–	ns
25	Address Hold Time for Latch	t_{AHL}	20	–	ns
27	Pulse Width, AS High	PW_{ASH}	110	–	ns
28	Address Strobe to E Delay	t_{ASED}	20	–	ns
30	Read Data Delay Time	t_{DDR}	–	240	ns
31	Write Data Setup Time	t_{DSW}	100	–	ns

NOTES:
1. Timing numbers relative to one side only. No numbers are intended to be cross-referenced from one side to the other.
2. Measurement points shown for ac timing are 0.8 V and 2.0 V, unless otherwise specified.

3

BUS TIMING DIAGRAMS



3

SIGNAL DESCRIPTION

The following paragraphs contain a brief description of the input and output signals.

V_{CC} AND V_{SS}

These pins supply power to the DPM. V_{CC} is +5 volts ±5% and V_{SS} is 0 volts or ground.

E CLOCK INPUTS (Ea AND Eb)

These are the input clocks from the respective processors and are positive during the latter portion of the bus cycle.

REGISTER SELECT INPUTS (RSa AND RSb)

These inputs function as register select inputs. A high on the RSa for side A or RSb for side B input allows selection of the semaphore and interrupt registers respectively for side A and side B by the lower three address bits. A low on RSa or RSb selects 256 bytes of RAM from side A or side B respectively.

CHIP SELECT INPUTS ($\overline{CS1}$a AND $\overline{CS1}$b)

These inputs function as chip select inputs for their respective sides. $\overline{CS1}$a must be low to select side A and $\overline{CS1}$b must be low to select side B. If $\overline{CS1}$a is high, side A is deselected. If $\overline{CS1}$b is high, side B is deselected.

MODE SELECT (MODE)

In normal operation, this pin should always be connected to V_{CC} (MODE=1). Each side has three states controlled by RSa and $\overline{CS1}$a for side A and RSb and $\overline{CS1}$b for side B.

If $\overline{CS1}$a is high, side A cannot be accessed. If $\overline{CS1}$a is low, side A accesses either 256 bytes of RAM or the six semaphore registers and the two interrupt registers depending on the level of RSa. If RSa is low, 256 bytes of RAM are accessed and if RSa is high, the six semaphores and two interrupt registers are accessed.

The six semaphore and two interrupt registers are redundantly mapped in the 256 byte mode. That is, only the low order three bits select one of eight registers and the upper five bits of address are not decoded. Refer to Table 1.

TABLE 1 — SIDE A CONTROL SIGNAL OPERATION

Mode	$\overline{CS1}$a	RSa	Operation
1	0	0	Access 256 Byte RAM Side A
1	0	1	Access Semaphore/$\overline{IRQ}$ Side A on Lower Three Bits of Address
1	1	X	Side A Not Selected

The three states for side B in the 256 byte mode are controlled in the manner as side A using RSb and $\overline{CS1}$b except that side B uses separated address and data inputs. Refer to Table 2.

TABLE 2 — SIDE B CONTROL SIGNAL OPERATION

Mode	$\overline{CS1}$b	RSb	Operation
1	0	0	Access 256 Byte RAM Side B
1	0	1	Access Semaphore/$\overline{IRQ}$ Side B on Lower Three Bits of Address
1	1	X	Side B Not Selected

INTERRUPT REQUEST OUTPUTS ($\overline{IRQ}$a AND $\overline{IRQ}$b)

These pins are active low open-drain outputs. A write to address F9 from one side asserts an interrupt, if not masked. On the other side, a write to address F9 sets this pin low.

B SIDE ADDRESS BUS INPUTS (A0-A7) AND B SIDE BIDIRECTIONAL DATA BUS (D0-D7)

When the B side is run from a multiplexed bus processor, the B side address pins are connected to the B side data pins, respectively (A0 to D0, A1 to D1, etc.).

SYSTEM RESET INPUT ($\overline{RESET}$)

A low level on this input causes the semaphore registers to be set to the states shown in Table 5 under **SEMAPHORE REGISTERS** and clears both bits of both $\overline{IRQ}$ registers to zeros. The RAM data is unaffected by $\overline{RESET}$.

ADDRESS STROBE INPUTS (ASa AND ASb)

The ASa input demultiplexes the eight low order address lines from the data lines on the A side. The falling edge of ASa latches the A side address within the DPM. The ASb input is used in the same manner when the B side is connected to a multiplexed bus. It must be connected to a high level when the B side is connected to a non-multiplexed bus.

A SIDE MULTIPLEXED ADDRESS/ BIDIRECTIONAL DATA BUS (AD0-AD7)

The A side can only be used with a multiplexed address/data bus. The A side addresses are on these lines during the time ASa is high. The lines are used as bidirectional data lines during the time Ea is high.

DUAL-PORT RAM

The dual-port memory unit contains 256 bytes of dual-port RAM that is accessed from either processor. It is selected in either case by eight address lines, register select, and chip select inputs. The direction of data transfer is controlled by the respective read/write (R/$\overline{W}$a or R/$\overline{W}$b) line. The dual-port RAM enables the processors to exchange data without interfering with devices on the other bus.

Simultaneous accesses by both sides of different locations of dual-port RAM will cause no ambiguities. Simultaneous reads by both sides of the same dual-port RAM location gives the proper data to both sides. On a simultaneous write and read of the same location, the data written is put into RAM but the data read is undefined. Simultaneous writes to the same RAM location result in undefined data being stored. Thus, simultaneous writes and simultaneous write and read to the same location should be avoided. The semaphore registers provide a tool for determining when the shared RAM is available.

SEMAPHORE REGISTERS

The dual-port memory unit contains six read/write semaphore registers. Only two bits of each register are used. Bit 7 is the semaphore (SEM) bit and bit 6 is the ownership (OWN) bit. The remaining six bits will read all zeros.

Each semaphore register is able to arbitrate simultaneous accesses to it. The semaphore register bits provide a mechanism for controlling accesses to the shared RAM but there are no hardware controls of the dual-port RAM by the semaphore registers.

Table 3 is the truth table for when a semaphore register is accessed by one of the processors. When a semaphore register is written, the actual data written is disregarded but the SEM bit is set to zero. When the register is read, the resulting SEM bit is one (for the next read). The data obtained from the read is interpreted as: SEM bit equals zero — resource available, SEM bit equals one — resource not available.

TABLE 3 — ONE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	$R/\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	$\overline{W}$	–	0
1	$\overline{W}$	–	0

*0 = Resource Available
1 = Resource Not Available

Table 4 shows the truth table if both processors read or read and write the same semaphore register at the same time. The A processor always reads the actual SEM bit. The B processor reads the SEM bit except during the simultaneous read of a clear SEM bit. This insures that during a simultaneous read, only the A processor reads a clear SEM bit and therefore has priority to the shared RAM.

TABLE 4 — SIMULTANEOUS ACCESS OF OF SEMAPHORE REGISTER TRUTH TABLE

Original SEM Bit	A Processor $R/\overline{W}$	A Processor Data Read	B Processor $R/\overline{W}$	B Processor Data Read	Resulting SEM Bit
0	R	0*	R	1*	1
1	R	1*	$\overline{W}$	–	0
1	$\overline{W}$	–	R	1*	0
1	R	1*	R	1*	1

*0 = Resource Available
1 = Resource Not Available

The ownership bit is a read-only bit that indicates which processor last set the SEM bit. The OWN bit is set to a one whenever the SEM bit is set from zero to one. The OWN bit as read by one processor is the complement of the bit read by the other processor.

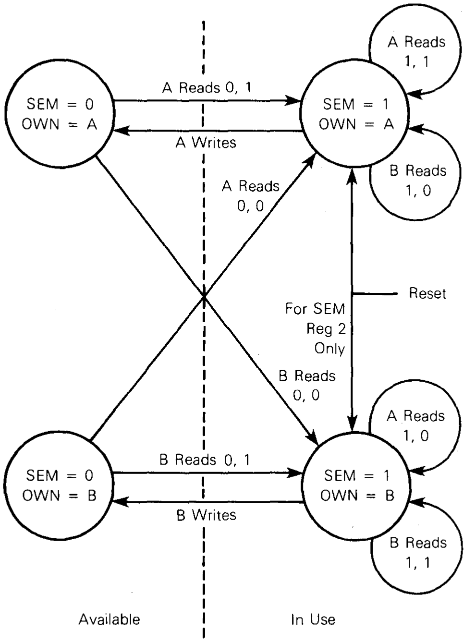
The reset state of the semaphore registers is defined in Table 5. The A processor owns all of the semaphore registers except the second semaphore register which is owned by the B processor.

TABLE 5 — RESET STATE OF SEMAPHORE REGISTERS

Semaphore Register Number	A Processor SEM Bit	A Processor OWN Bit	B Processor SEM Bit	B Processor OWN Bit
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

A state diagram for a semaphore register is shown in Figure 3.

FIGURE 3 — STATE DIAGRAM FOR SEMAPHORE REGISTER



NOTES:
1. Writes to a semaphore register are valid only if SEM = 1 and OWN = 1.
2. When A and B simultaneously read a semaphore register, the hardware handles it as a read by A followed by a read by B.

INTERRUPT REGISTERS

The dual-port memory unit contains two addressable locations at F8 and F9 on both sides that control the interrupt ($\overline{IRQ}$) operation between the processors. Although there is only one hardware register for each side, for purposes of explanation the register accessed at location F8 is referred to as the IRQX status register and the register accessed at location F9 is referred to as the IRQX control register (refer to Table 6). The registers each consisting of two bits have identical bit arrangements. Bit 6 is the enable bit and bit 7 is the flag bit. The other six bits are not used and always read as zero. When $\overline{\text{RESET}}$ is asserted, both bits are cleared to zero.

Table 7 summarizes the bits involved when reading or writing to the status or control registers at F8 or F9. The enable bits on either side (A or B) track the data that is written into the status register from that side. Writes to the control register do not alter data. The actual data written is disregarded but the action sets the flag bit in the other side's register and asserts an interrupt signal if enabled.

The following describes how the B side interrupt is asserted from the A side. The A side interrupt is controlled in a similar manner.

When the enable bit in the IRQb status register is set (bit 6 = 1), a write to IRQa control register sets the flag bit in the IRQb status register (bit 7 = 1) and causes an interrupt on the B side by setting the IRQb pin low. Reading the IRQb status register reads the state of the B side enable and flag bits. Reading the IRQb control register also reads the enable and flag bits but in addition, clears the B side flag bit (bit 7 = 0) and clears the B side interrupt by removing the low condition on the IRQb pin.

The enable bit in the IRQb status register (bit 6) is changed by writing the proper data to bit 6 of the IRQb status register. If the B side enable bit is zero, interrupts are prevented on the B side. However, a write to the IRQa control register still sets the B side flag bit.

INTERNAL REGISTER ADDRESSES

Table 8 shows the address of the RAM, $\overline{\text{IRQ}}$, and semaphore registers. The addresses to these registers are the same whether accessed from the A or B side. The address and data buses are multiplexed on the A side. The B side has separate address and data buses. The B side can be used on a multiplexed bus by connecting the corresponding address and data bit pins together (A0 to D0, A1 to D1, etc.) and using the B side address strobe input pin.

TABLE 6 — IRQ REGISTERS

Location	Register Name	Bit 7	Bit 6	Bits 5 to 0
A Side F8	IRQa Status	Flag	Enable	Not Used
A Side F9	IRQa Control	Flag	Enable	Not Used
B Side F8	IRQb Status	Flag	Enable	Not Used
B Side F9	IRQb Control	Flag	Enable	Not Used

TABLE 7 — INTERRUPT OPERATION

Operation	Action Taken
A Reads IRQa Status at F8	Read EA and FA
A Writes IRQa Status at F8	Writes to EA
A Reads IRQa Control at F9	Read EA and FA; Clear FA
A Writes IRQa Control at F9	Set FB; Assert IRQB if Enabled
B Reads IRQb Status at F8	Read EB and FB
B Writes IRQb Status at F8	Writes to EB
B Reads IRb Control at F9	Read EB and FB; Clear FB
B Writes IRQb Control at F9	Set FA; Assert IRQA if Enabled

F8 and F9 are Address Locations
EA and FA are A Side Enable and Flag Bits
EB and FB are B Side Enable and Flag Bits

TABLE 8 — REGISTER LOCATIONS

RS	Address	Register Name
0	00-FF	Dual Ported RAM
1	00-07	IRQ and Semaphore
1	08-0F	IRQ and Semaphore
1	10-17	IRQ and Semaphore
1	18-1F	IRQ and Semaphore
1	• • •	IRQ and Semaphore
1	E0-E7	IRQ and Semaphore
1	E8-EF	IRQ and Semaphore
1	F0-F7	IRQ and Semaphore
1	F8-FF	IRQ and Semaphore

Where:
X is 0 through F of the upper four bits of the address (note that only the lower three bits of the address are decoded):
X0 and X8 IRQa or IRQb Status
X1 and X9 IRQa or IRQb Control
X2 and XA Semaphore 1
X3 and XB Semaphore 2
X4 and XC Semaphore 3
X5 and XD Semaphore 4
X6 and XE Semaphore 5
X7 and XF Semaphore 6

3

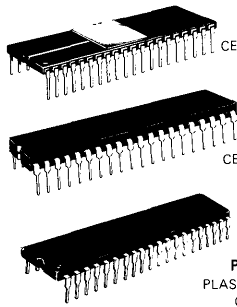
MOTOROLA

MC6835

Advance Information

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

MASK PROGRAMMED CRT CONTROLLER (CRTC)



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

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.

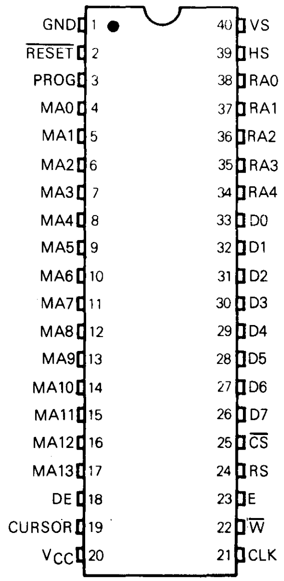
- Cost Effective ROM Based CRTC Which Supports Two Screen Formats
- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games; Information Displays
- Alphanumeric, Semigraphic, and Full Graphic Capability
- Timing 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

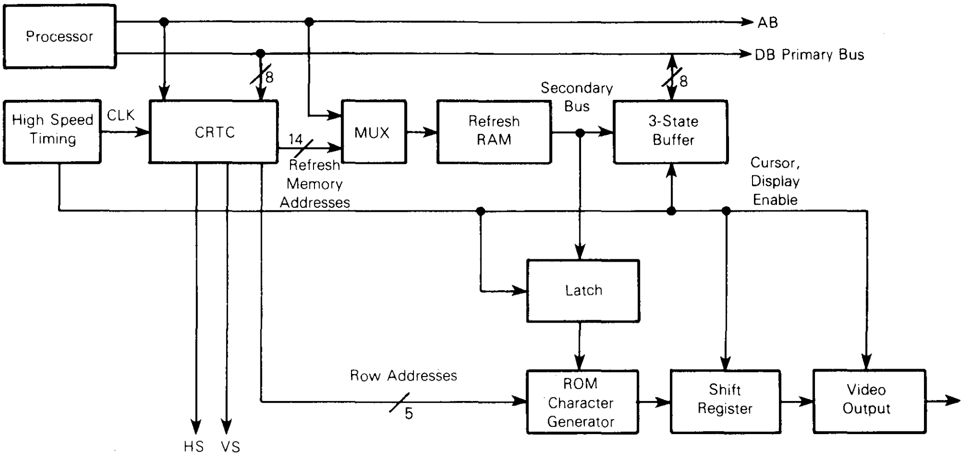
*With respect to GND (V_{SS}).

PIN ASSIGNMENT



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

FIGURE 1 — TYPICAL CRT CONTROLLER APPLICATION



3

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

POWER CONSIDERATIONS

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

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

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.

3

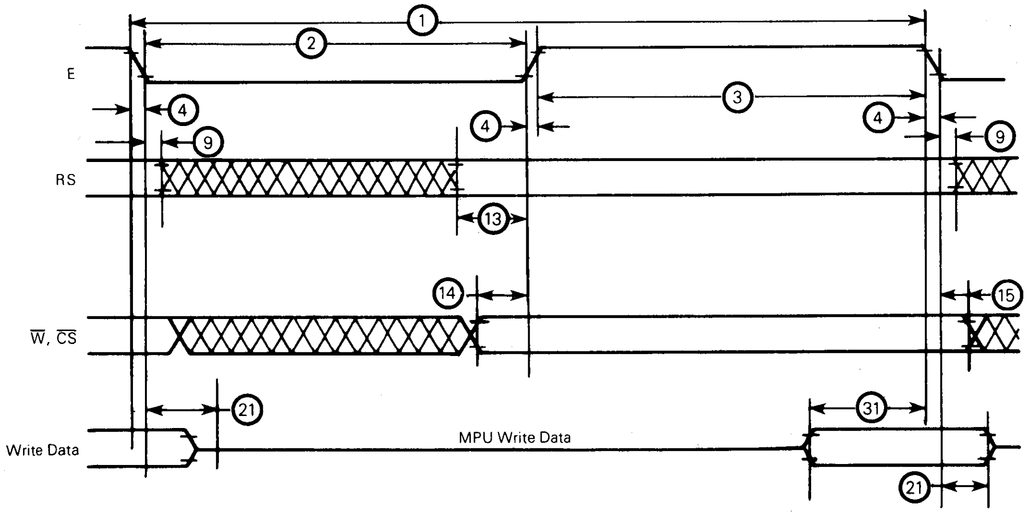
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted) (Reference Figures 2-4)

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
Hi-Z (Off State) Input Current (V_{CC} = 5.25 V) (V_{in} = 0.4 to 2.4 V)		I_{TSI}	−10	–	10	μA
Output High Voltage (I_{Load} = −100 μA)			2.4	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_D	–	150	300	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 2 and 3)

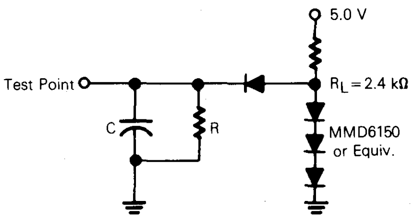
Ident. Number	Characteristics	Symbol	MC6835		MC68A35		MC68B35		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	280	–	210	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	280	–	220	–	ns
4	Clock Transition Time	t_r, t_f	–	25	–	25	–	20	ns
9	Address Hold Time (RS)	t_{AH}	10	–	10	–	10	–	ns
13	RS Setup Before E	t_{AS}	80	–	60	–	40	–	ns
14	$\overline{W}$ and $\overline{CS}$ Setup Before E	t_{CS}	80	–	60	–	40	–	ns
15	Hold Time for $\overline{W}$ and $\overline{CS}$	t_{CH}	10	–	10	–	10	–	ns
21	Write Data Hold Time Required	t_{DHW}	10	–	10	–	10	–	ns
31	Peripheral Input Data Setup	t_{DSW}	165	–	80	–	60	–	ns

FIGURE 2 — MC6835 BUS TIMING



NOTES:
1. Voltage levels shown are $V_L \leq 0.4$ V, $V_H \geq 2.4$ V unless otherwise noted.
2. Measurement points shown are 0.8 V and 2.0 V unless otherwise noted.

FIGURE 3 — BUS TIMING TEST LOAD

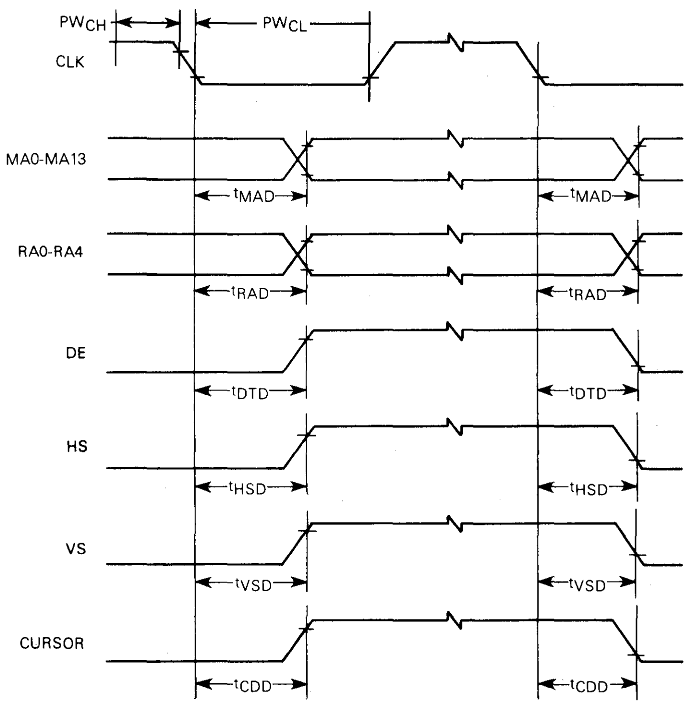


C = 130 pF for D0-D7
= 30 pF for MA0-MA13, RA0-RA4, DE, HS, VS, and CURSOR
R = 11 kΩ for D0-D7
= 24 kΩ for All Other Outputs

CRTC TIMING CHARACTERISTICS (See Figure 4)

Characteristics	Symbol	MC6835 Min	MC6835 Max	MC68A35 Min	MC68A35 Max	MC68B35 Min	MC68B35 Max	Unit
Minimum Clock Pulse Width, Low	PW_{CL}	150	–	140	–	130	–	ns
Minimum Clock Pulse Width, High	PW_{CH}	150	–	140	–	130	–	ns
Clock Frequency	f_c	330	–	300	–	270	–	ns
Rise and Fall Time for Clock Input	t_r, t_f	–	20	–	20	–	20	ns
Memory Address Delay Time	t_{MAD}	–	160	–	160	–	160	ns
Raster Address Delay Time	t_{RAD}	–	160	–	160	–	160	ns
Display Timing Delay Time	t_{DTD}	–	250	–	250	–	200	ns
Horizontal Sync Delay Time	t_{HSD}	–	250	–	250	–	200	ns
Vertical Sync Delay Time	t_{VSP}	–	250	–	250	–	200	ns
Cursor Display Timing Delay Time	t_{CDD}	–	250	–	250	–	200	ns

FIGURE 4 — 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.

3

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 5 and 6. Non-interlacing scanning consists of one field per frame. The scan lines in Figure 5 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 6, 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 7. The figure also shows the corresponding timing and levels for a video signal that would generate the characters.

FIGURE 5 — RASTER SCAN SYSTEM (NON-INTERLACE)

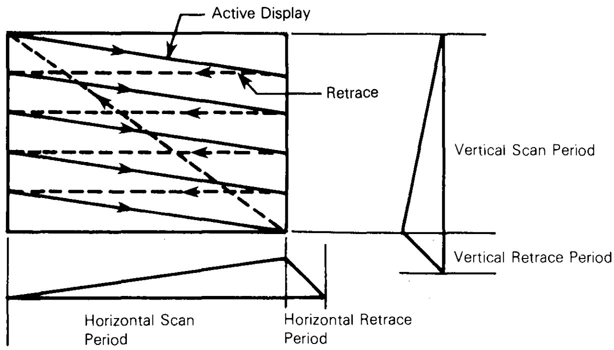


FIGURE 6 — RASTER SCAN SYSTEM (INTERLACE)

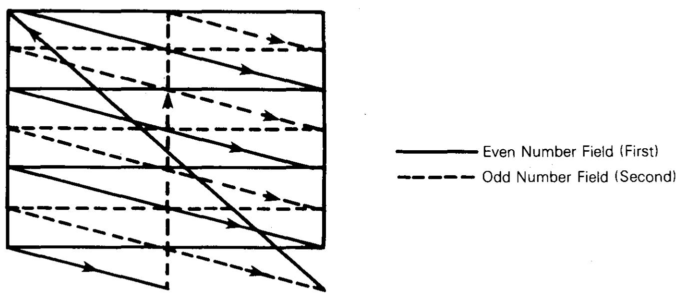
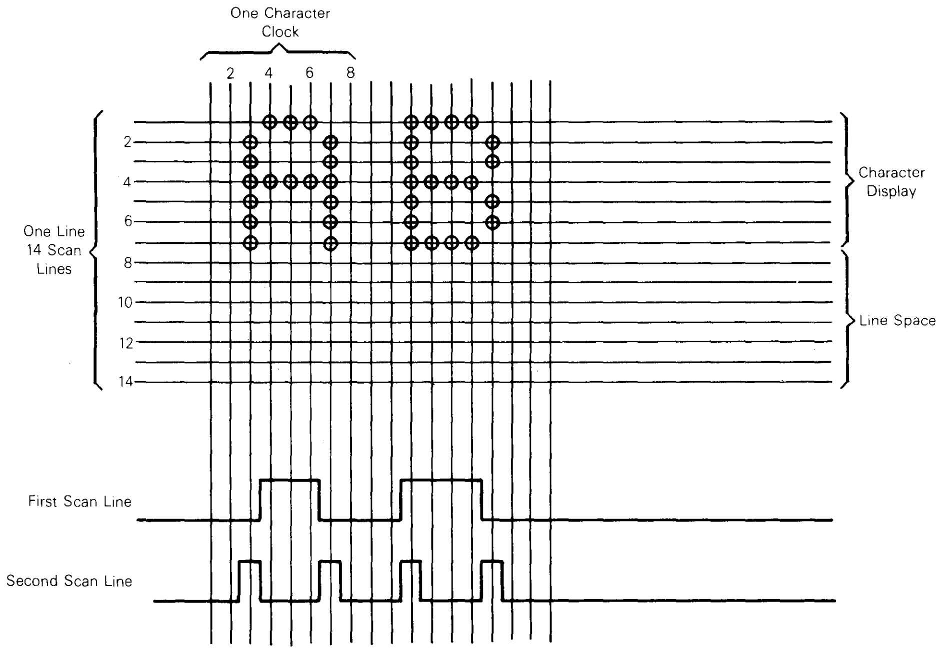


FIGURE 7 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL



Referring to Figure 1, 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).

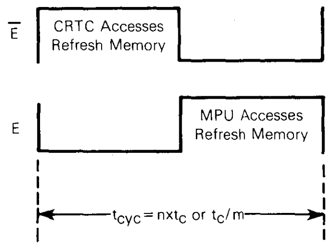
All timing in the CRTC is derived from the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high speed logic (TTL) to generate the CLK signal. The high speed logic must also generate the timing and control signals necessary for the Shift Register, Latch and MUX Control shown in Figure 1.

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 8. 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 8 — 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 data bus (D0-D7) using $\overline{\text{CS}}$, RS, E, and $\overline{\text{W}}$ for control signals.

Data Bus (D0-D7) — The data lines (D0-D7) comprise the write only data bus.

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 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 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.

Write ($\overline{\text{W}}$) — The $\overline{\text{W}}$ line is a high-impedance TTL/MOS-compatible input which determines whether the internal register file gets written. 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 4	3	2	1	0	Register #	Register File	Program Unit	Read	Write	Number of Bits 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	\	\	\					
							R0	Horizontal Total	Char.	No	No								
							R1	Horizontal Displayed	Char.	No	No								
							R2	H. Sync Position	Char.	No	No								
Note 3							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.

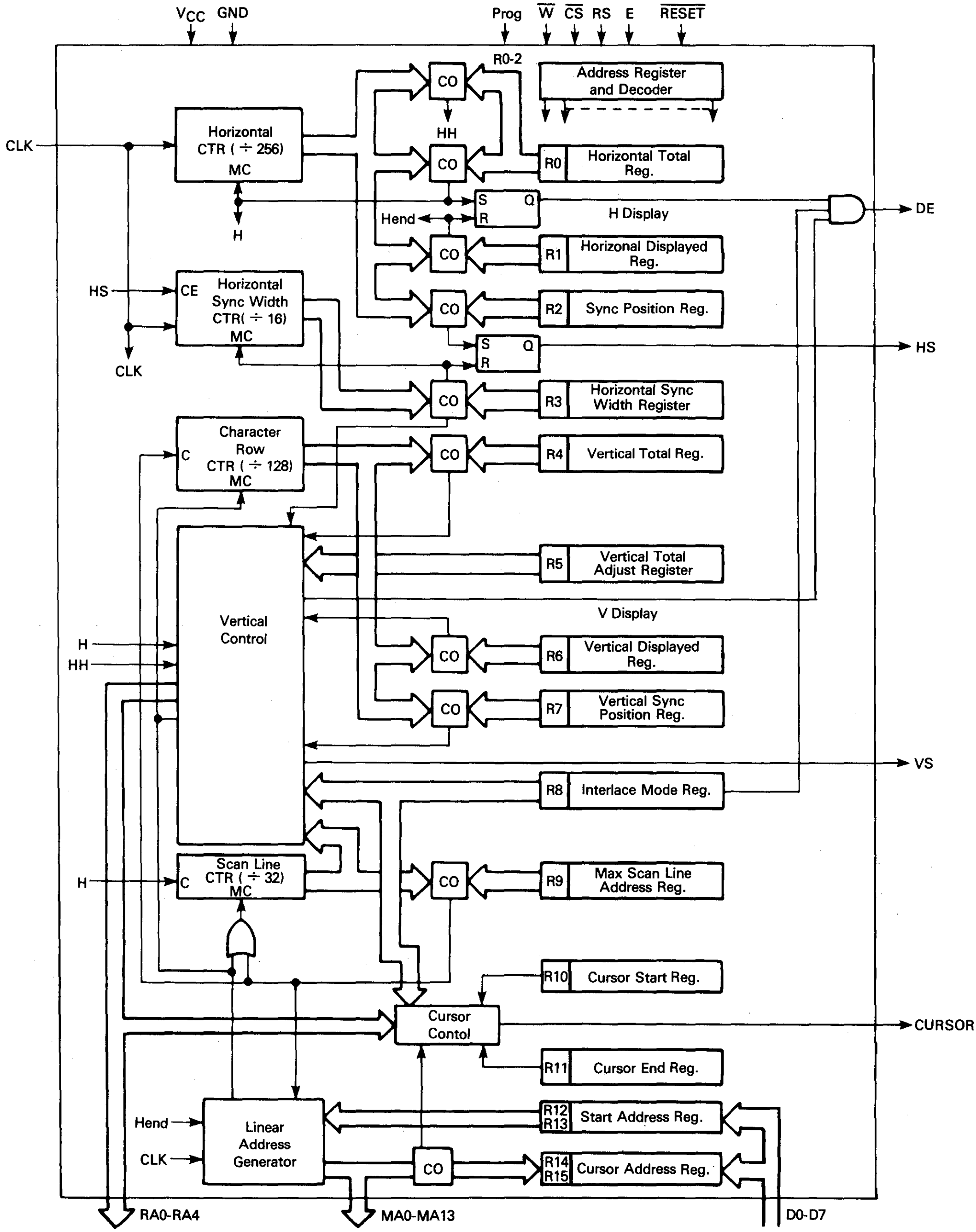
The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory and their positions on the screen. Fourteen outputs, MA0-MA13, are available for addressing up to four pages of 4K characters, eight pages of 2K characters, etc.

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 — $\overline{W}$, $\overline{CS}$, RS, and E. Refer to Figure 9.

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, and start address. The register addresses and sizes are shown in Table 1.

FIGURE 9 — 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 10 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 11. 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 12.

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.

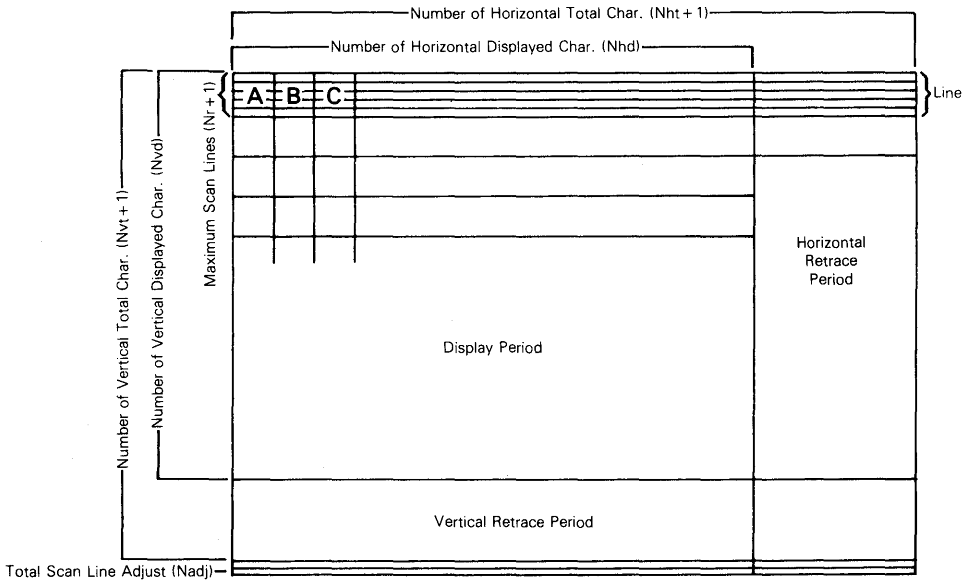
Horizontal Displayed Register (R1) — This 8-bit register determines the number of displayed characters per line. Any 8-bit number may be programmed as long as the contents of R0 are greater than the contents of R1.

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.

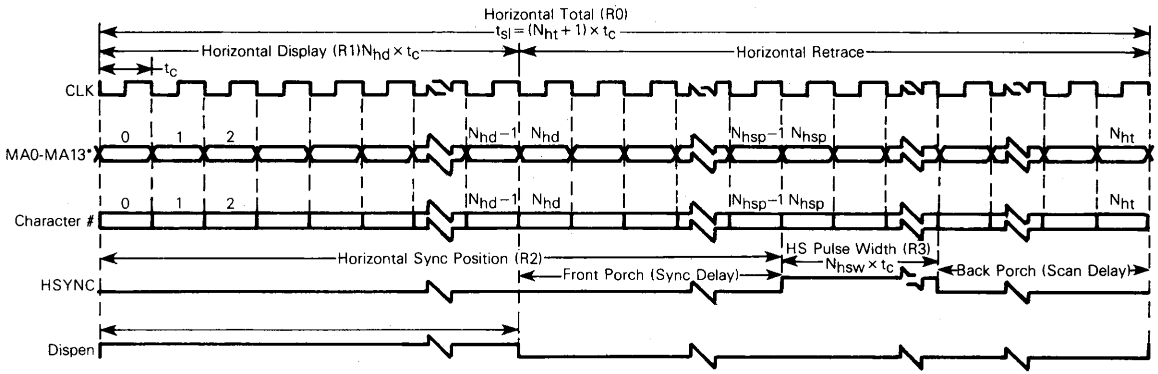
Horizontal Timing Summary (Figure 11) — The difference between R0 and R1 is the horizontal blanking interval. This interval in the horizontal scan period allows the beam to return (retrace) to the left side of the screen. The retrace time is determined by the monitor's horizontal scan components. Retrace 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 10 — ILLUSTRATION OF THE CRT SCREEN FORMAT



NOTE 1: Timing values are described in Table 8.

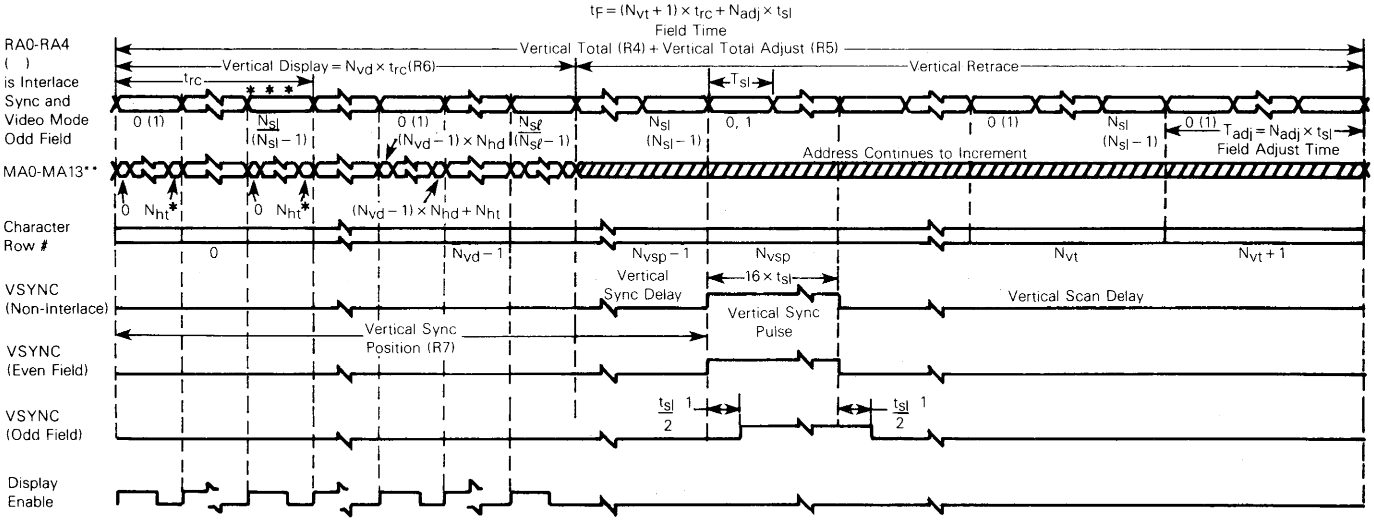
FIGURE 11 — CRTC HORIZONTAL TIMING



* Timing is shown for first displayed scan row only. See Chart in Figure 15 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 12 — 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 6 — The Odd Field is offset ½ horizontal scan time.
2. Timing values are described in Table 5.

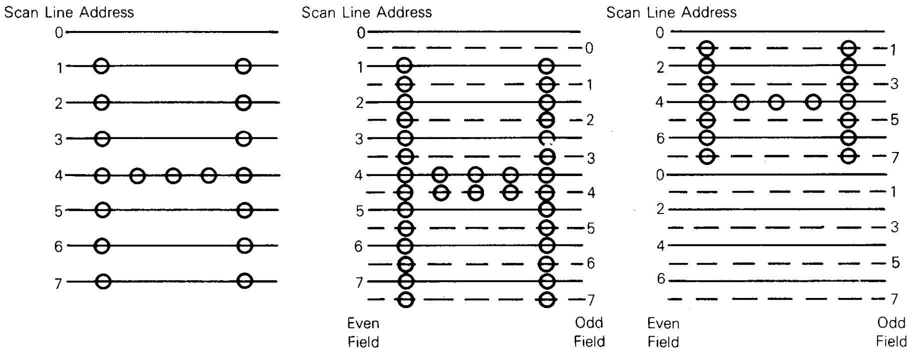
TABLE 2 — INTERLACE MODE REGISTER

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 13 — 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 Figure 5 and 13a. Each scan line is refreshed at the VS frequency (e.g., 50 or 60 Hz).

Two interlace modes are available as shown in Figures 6, 13b, and 13c. 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 13b. 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 14. 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 four 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 $\overline{CS}$ are low, the Address Register is selected. When $\overline{CS}$ is low and RS is high, the register pointed to by the Address Register is selected.

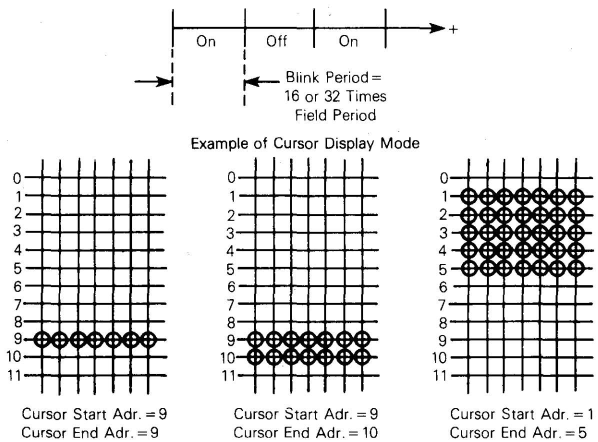
Start Address Register (R12-H, R13-L) — This 14-bit write-only register pair controls the first address output by the CRTC after vertical blanking. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. The start address register determines which portion of the 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 15 shows an M6800 program which could be used to program the CRT Controller.

FIGURE 14 — CURSOR CONTROL



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 15 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 18.

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 18.

FIGURE 15 — 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)		________

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)	27	18
R5	Vertical Adjust (Line 9 Lines)	2	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)	10	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

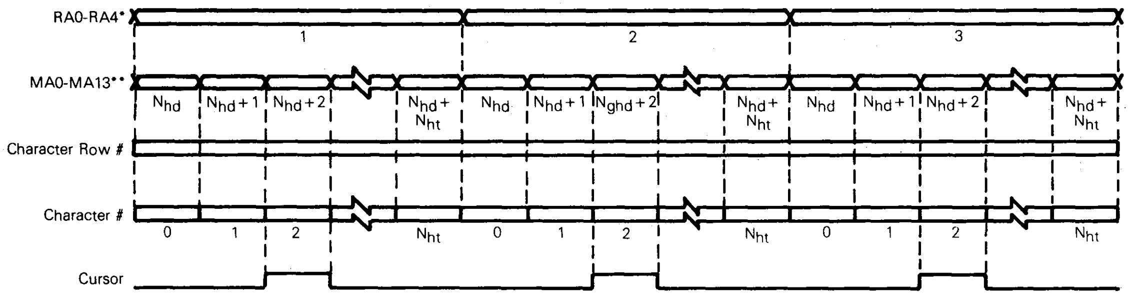
OPERATION OF THE CRTC

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 11, 12, 16, and 17). The screen format of this example is shown in Figure 10. Figure 17 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	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		
R11	Cursor End		
R12	Start Address (H)	0	
R13	Start Address (L)	0	
R14	Cursor (H)		
R15	Cursor (L)		

FIGURE 16 — 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 17 — REFRESH MEMORY ADDRESSING (MA0-MA13) STATE CHART

	Character Row	Scan Line	Horizontal Display				Horizontal Retrace (Non-Display)		
			Character						
Vertical Display	0	0 N_{sl}	0 ⋮ 0	1 ⋮	→ →	$N_{hd}-1$ ⋮ $N_{hd}-1$	N_{hd} ⋮ N_{hd}	→ →	N_{ht} ⋮ N_{ht}
	1	0 N_{sl}	N_{hd} ⋮ N_{hd}	$N_{hd}+1$ ⋮ $N_{hd}+1$	→ →	$2XN_{hd}-1$ ⋮ $2XN_{hd}-1$	$2XN_{hd}$ ⋮ $2XN_{hd}$	→ →	$N_{hd}+N_{ht}$ ⋮ $N_{hd}+N_{ht}$
	2	0 N_{sl}	$2XN_{hd}$ ⋮ $2XN_{hd}$	$2XN_{hd}+1$ ⋮ $2XN_{hd}+1$	→ →	$3XN_{hd}-1$ ⋮ $3XN_{hd}-1$	$3XN_{hd}$ ⋮ $3XN_{hd}$	→ →	$2N_{hd}+N_{ht}$ ⋮ $2N_{hd}+N_{ht}$
			↓	↓		↓	↓		↓
	$N_{vd}-1$	0 N_{sl}	$(N_{vd}-1)\times N_{hd}$ ⋮ $(N_{vd}-1)\times N_{hd}$	$(N_{vd}-1)\times N_{hd}+1$ ⋮ $(N_{vd}-1)\times N_{hd}+1$	→ →	$N_{vd}\times N_{hd}+1$ ⋮ $N_{vd}\times N_{hd}-1$	$N_{vd}\times N_{hd}$ ⋮ $N_{vd}\times N_{hd}$	→ →	$(N_{vd}-1)\times N_{hd}+N_{ht}$ ⋮ $(N_{vd}-1)\times N_{hd}+N_{ht}$
Vertical Retrace (Non-Display)	N_{vd}	0 N_{sl}	$N_{vd}\times N_{hd}$ ⋮ $N_{vd}\times N_{hd}$	$N_{vd}\times N_{hd}+1$ ⋮ $N_{vd}\times N_{hd}+1$	→ →	$(N_{vd}+1)\times N_{hd}-1$ ⋮ $(N_{vd}+1)\times N_{hd}-1$	$(N_{vd}+1)\times N_{hd}$ ⋮ $(N_{vt}+1)\times N_{hd}$	→ →	$N_{vd}\times N_{hd}+N_{ht}$ ⋮ $N_{vd}+N_{hd}+N_{ht}$
	N_{vt}	0 N_{sl}	$N_{vt}\times N_{hd}$ ⋮ $N_{vt}\times N_{hd}$	$N_{vt}\times N_{hd}+1$ ⋮		$(N_{vt}+1)\times N_{hd}-1$ ⋮ $(N_{vt}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$ ⋮ $(N_{vt}+1)\times N_{hd}$		$N_{vt}\times N_{hd}+N_{ht}$ ⋮ $N_{vt}\times N_{hd}+N_{ht}$
	$N_{vt}+1$	0 $N_{adj}-1$	$(N_{vt}+1)\times N_{hd}$ ⋮ $(N_{vt}+1)\times N_{hd}$	$(N_{vt}+1)\times N_{hd}+1$ ⋮ $(N_{vt}+1)\times N_{d}+1$		$(N_{vt}+2)\times N_{hd}-1$ ⋮ $(N_{vt}+2)\times N_{hd}-1$	$(N_{vt}+2)\times N_{hd}$ ⋮ $(N_{vt}+2)\times N_{hd}$		$(N_{v}+1)N_{hd}+N_{ht}$ ⋮ $(N_{vt}+1)N_{hd}+N_{ht}$

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 18 — 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

Package Type	Frequency (MHz)	Temperature*	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6835L
	1.0	−50°C to 85°C	MC6835CL
	1.5	0°C to 70°C	MC68A35L
	1.5	−50°C to 85°C	MC68A35CL
	2.0	0°C to 70°C	MC68B35L
	2.0	−50°C to 85°C	MC68B35CL
Cerdip S Suffix	1.0	0°C to 70°C	MC6835S
	1.0	−50°C to 85°C	MC6835CS
	1.5	0°C to 70°C	MC68A35S
	1.5	−50°C to 85°C	MC68A35CS
	2.0	0°C to 70°C	MC68B35S
	2.0	−50°C to 85°C	MC68B35CS
Plastic P Suffix	1.0	0°C to 70°C	MC6835P
	1.0	−50°C to 85°C	MC6835CP
	1.5	0°C to 70°C	MC68A35P
	1.5	−50°C to 85°C	MC68A35CP
	2.0	0°C to 70°C	MC68B35P
	2.0	−50°C to 85°C	MC68B35CP

MOTOROLA

MCM6836E16
MCM6836R16

Advance Information

128K-BIT COMBINATION ROM/EEPROM MEMORY UNIT

The MCM6836E16/MCM6836R16 Combination ROM/EEPROM Memory (CREEM) is a 16K byte combination memory device with 14K bytes of mask programmable ROM and 2K bytes of electrically erasable programmable ROM (EEPROM). It is designed for handling data in applications requiring nonvolatile memory and in-system reprogramming to a portion of the memory. The MCM6836 saves time and money because of the in-system erase and reprogram capability of its 2K bytes of EEPROM. The industry standard pinout in a 28-pin dual-in-line package makes the MCM6836( )16 compatible with 128K-bit ROMs and EPROMs.

For easy use, the MCM6836( )16 device operates in the read mode from a single power supply and has a static power-down mode. The MCM6836R16 version has a 256 byte user programmable redundancy EEPROM on chip. It can be programmed by the user to replace any page of 256 bytes of memory in the mask ROM or EEPROM sections.

The following are some of the major features of the MCM6836( )16.

- 128K-Bit ROM/EEPROM Combination Memory Organized as 16,384 × 8 Bytes
- Lowest Order 2K Bytes are Bulk Erasable EEPROM
- Remaining 14K Bytes are Mask Programmed ROM
- Packaged in Standard 28-Pin DIP
- Pin Compatible with 128K-Bit ROMs and EEPROMs
- In the Read Operating Mode Only +5 V Power Supply is Required
- +21 Vdc Programming Power Supply
- Bulk Erase
- 256 Bytes of Spare Memory are Included on Chip (MCM6836R16 Only)
- Seven Operating Modes: Read, Standby, Program, Erase, Verify, Replace (MCM6836R16 Only), and Erase-of-Replace (MCM6836R16 Only)

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

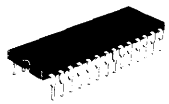
Package Type	Order Number
Cerdip S Suffix	MCM6836E16S MCM6836R16S
Plastic P Suffix	MCM6836E16P MCM6836R16P

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

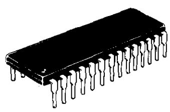
HMOS

HIGH-DENSITY N-CHANNEL PROCESS

128K-BIT COMBINATION ROM/EEPROM MEMORY



S SUFFIX
CERDIP PACKAGE
CASE 733



P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENT

V_{PP}	1	28	V_{CC}
A12	2	27	$\overline{W}$
A7	3	26	A13
A6	4	25	A8
A5	5	24	A9
A4	6	23	A11
A3	7	22	$\overline{G}$
A2	8	21	A10
A1	9	20	$\overline{E}$
A0	10	19	DQ7
DQ0	11	18	DQ6
DQ1	12	17	DQ5
DQ2	13	16	DQ4
V_{SS}	14	15	DQ3

Pin Names

A0-A13 . . . . Address
$\overline{E}$ . . . . Chip Enable
$\overline{G}$ . . . . Output Enable
$\overline{W}$ . . . . Write
DQ0-DQ7 . . . . Data
V_{PP} . . . . Program Voltage
V_{CC} . . . . +5 V Power Supply
V_{SS} . . . . Ground

3

FIGURE 1 — MCM6836( )16 EEPROM MEMORY UNIT BLOCK DIAGRAM

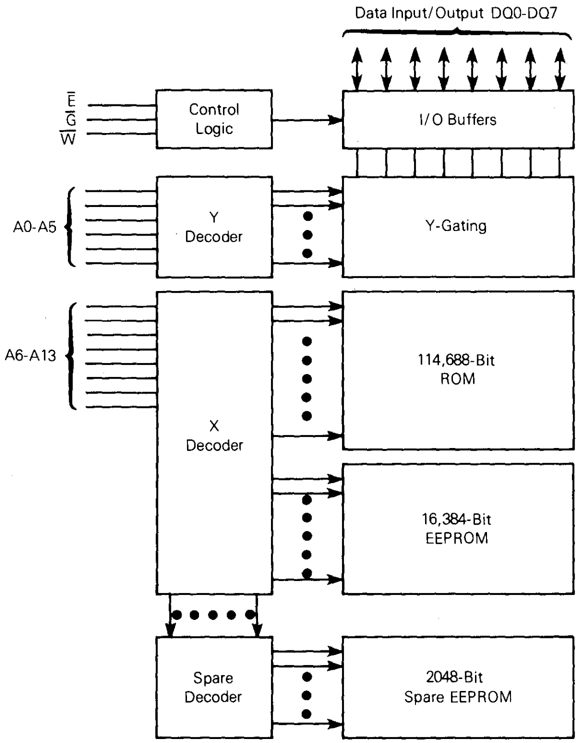
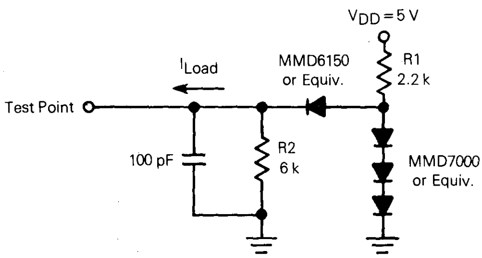


FIGURE 2 — AC TEST LOAD



MAXIMUM RATINGS (Voltages Referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Programming Voltage	V_{PP}	−0.3 to +22	V
Input Voltage Mode Programming Pin All Other Inputs	 V_{IHH} V_{in}	 −0.3 to +19 −0.3 to +7	 V 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 of electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{CC}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Unit
Thermal Resistance Cerdip Plastic	 θ_{JA}	 60 100	 °C/W

POWER CONSIDERATIONS

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

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

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 DC ELECTRICAL CHARACTERISTICS ($V_{CC} = V_{PP} = 5.0$ V ± 10%, $V_{SS} = 0$ Vdc, $T_A = 0°$ to 70°C unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Output High Voltage ($I_{Load} = -400$ μA)	V_{OH}	2.4	–	V
Output Low Voltage ($I_{Load} = 2.1$ mA)	V_{OL}	–	0.4	V
Input High Voltage	V_{IH}	2.0	V_{CC}	V
Input Low Voltage All Inputs (Except V_{PP})	V_{IL}	−0.1	0.8	V
Input High Voltage V_{PP} (Normal Operating Mode)	V_{IH}	V_{CC}	V_{CC}	V
Supply Current Measured at $T_A = 0°C$ in Read Mode Operation ($V_{CC} = 4.5$ to 5.5 V)	I_{CC}	–	100	mA
Input Low Current ($V_{IL} = 0$)	I_{IL}	–	−10	μA
Input High Current ($V_{IH} = 5.25$ V)	I_{IH}	–	10	μA
Hi-Z Output Leakage Current Low ($V_{out} = 0.4$ V)	I_{OZL}	–	−10	μA
Hi-Z Output Leakage Current High ($V_{out} = 5.5$ V)	I_{OZH}	–	10	μA
Capacitance Output ($V_{out} = 0$) Input ($V_{in} = 0$)	 C_{out} C_{in}	 – –	 12 10	 pF pF
V_{PP} Current	I_{PP}	–	12	mA
Supply Current During Standby, Measured at $T_A = 0°C$ ($V_{CC} = 4.5$ to 5.5 V, $\overline{E} \geq V_{IH}$, $\overline{G} \geq V_{IH}$)	$I_{CC(SB)}$	–	25	mA

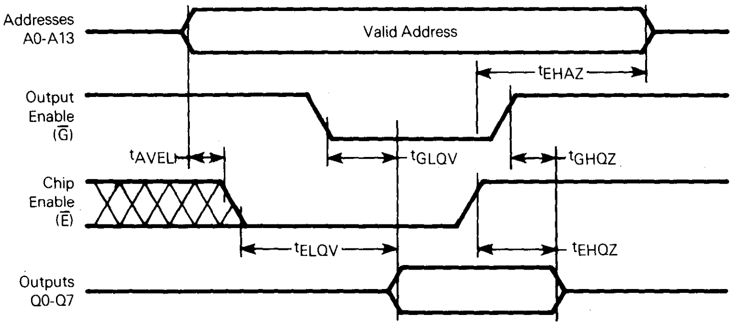
NOTES: 1. In normal read operation, if the V_{PP} pin is connected to V_{CC}, then the total I_{CC} current will be the sum of the total supply and the V_{PP} current.
2. In all cases, V_{CC} and V_{IHH} must be applied simultaneously with or prior to V_{PP}, V_{CC} and V_{IHH} must be switched off simultaneously with or after V_{PP}.

3

READ MODE AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0 to 70°C)

Characteristic	Symbol	Min	Max	Unit
Access Time (From Chip Enable)	t_{ELQV}	–	250	ns
Access Time (From Output Enable)	t_{GLQV}	–	100	ns
Address Hold Time (From Chip Enable)	t_{EHAZ}	0	–	ns
Address Setup Time	t_{AVEL}	0	–	ns
Disable Time (From Output Enable)	t_{GHQZ}	0	80	ns
Disable Time (From Chip Enable)	t_{EHQZ}	10	80	ns

READ MODE TIMING DIAGRAM



NOTES: 1. Voltage levels shown are $V_{OL} \leq 0.4$ V and $V_{OH} \geq 2.4$ V unless otherwise specified.
2. Timing level measurement points are 0.8 V and 2.0 V unless otherwise specified.
3. $\overline{G}$ may be delayed up to t_{ELQV}-t_{GLQV} after the falling edge of $\overline{E}$ without impact on t_{ELQV}.

PROGRAMMING OPERATION DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 25°C unless otherwise noted)

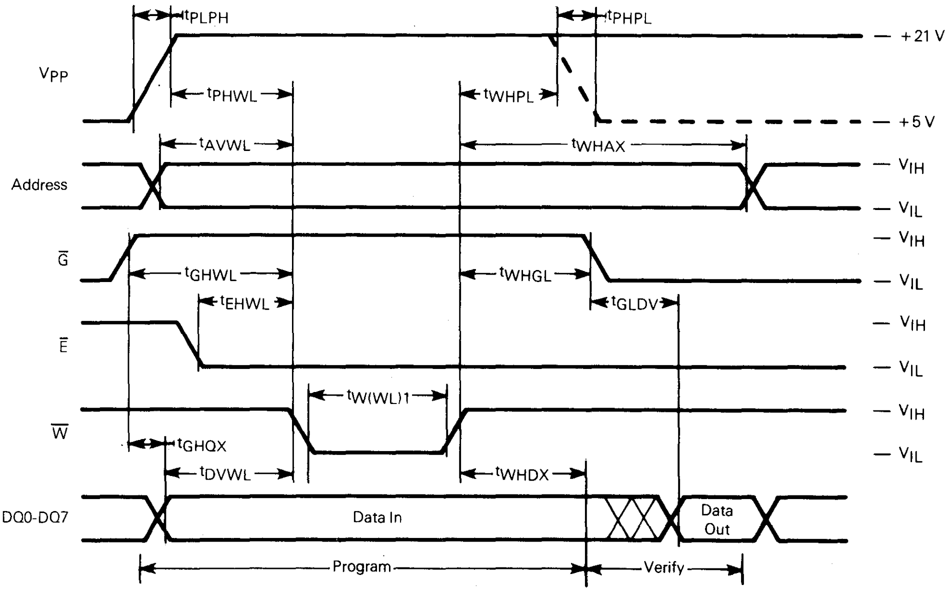
Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage (V_{PP} Pin)	V_{PP}	20	21	22	V
Input High Voltage For Data	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	−0.1	–	0.8	V
Address, $\overline{E}$, $\overline{G}$, and $\overline{W}$ Sink Current (V_{in} = 5.25 V/0.4 V)	I_{leak}	–	–	10	μA
V_{PP} Supply Current (V_{PP} = 21 ± 1 V, $\overline{W}$ = V_{IH})	I_{PP1}	–	–	10	mA
V_{PP} Programming Pulse Supply Current (V_{PP} = 21 ± 1 V, $\overline{W}$ = V_{IL})	I_{PP2}	–	–	10	mA
V_{CC} Supply Current	I_{CC}	–	–	115	mA

PROGRAMMING OPERATION AC TIMING CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 21 ± 1 V, T_A = 25°C)

Characteristic	Symbol	Min	Max	Unit
V_{PP} Rise Time	t_{PLPH}	50	–	ns
V_{PP} Fall Time	t_{PHPL}	50	–	ns
V_{PP} Setup Time	t_{PHWL}	2.0	–	μs
V_{PP} Hold Time	t_{WHPL}	2.0	–	μs
Address Setup Time	t_{AVWL}	2.0	–	μs
Address Hold Time	t_{WHAX}	2.0	–	μs
Output Enable High to Program Pulse	t_{GHWL}	2.0	–	μs
Output Enable Hold Time	t_{WHGL}	2.0	–	μs
Chip Enable Setup Time	t_{EHWL}	2.0	–	μs
Output Disable to Hi-Z Output	t_{GHQX}	0.1	100	ns
Data Setup Time	t_{DVWL}	2.0	–	μs
Data Hold Time	t_{WHDX}	2.0	–	μs
Program Pulse Width	$t_{W(WL)1}$	1.0	10	ms
Output Enable to Valid Data	t_{GLDV}	–	200	ns

3

PROGRAMMING OPERATION TIMING DIAGRAM



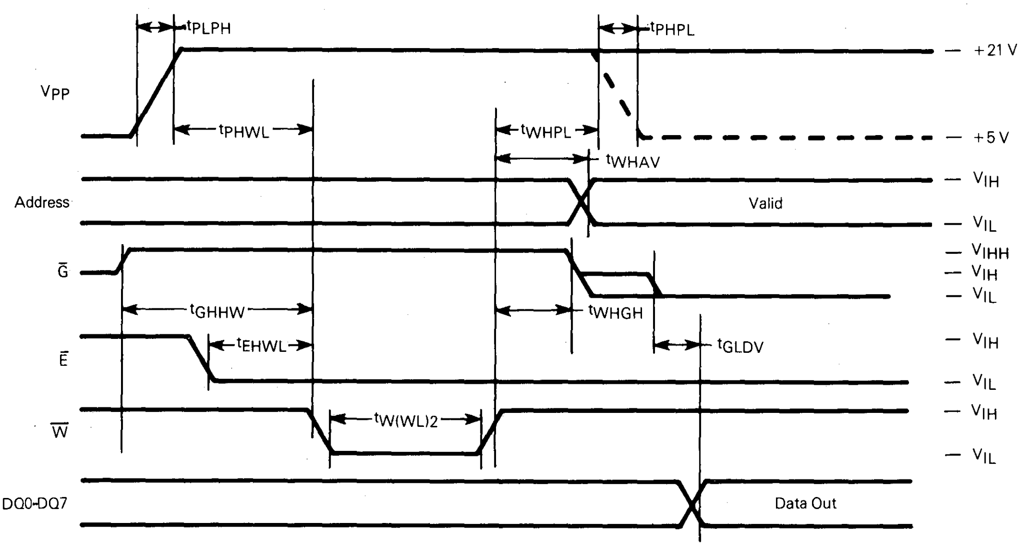
ERASE OPERATION DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 21 ± 1 Vdc, T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current for Any Input @ V_{in}	I_{leak}	–	–	10	μA
V_{CC} Supply Current (Outputs Open, $\overline{W}$ = V_{IL})	I_{CC}	–	–	115	mA
V_{PP} Supply Current ($\overline{W}$ = V_{IL})	I_{PP}	–	5	10	mA
Input Low Level	V_{IL}	−0.1	–	0.8	V
Input High Level	V_{IH}	2.0	–	V_{CC}	V
Input Mode Select High	V_{IHH}	12	15	19	V

ERASE OPERATION AC TIMING CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 21 ± 1 Vdc, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
V_{PP} Rise Time	t_{PLPH}	50	–	–	ns
V_{PP} Fall Time	t_{PHPL}	50	–	–	ns
V_{PP} Setup Time	t_{PHWL}	2.0	–	–	μs
V_{PP} Hold Time	t_{WHPL}	2.0	–	–	μs
Address Delay Time	t_{WHAV}	2.0	–	–	μs
Output Enable Setup Time	t_{GHHWL}	2.0	–	–	μs
Output Enable Hold Time	t_{WHGH}	2.0	–	–	μs
Chip Enable Setup Time	t_{EHWL}	2.0	–	–	μs
Erase Pulse Width	$t_{W(WL)2}$	1.0	10	100	ms
Output Enable to Invalid Data	t_{GLDV}	–	–	200	ns

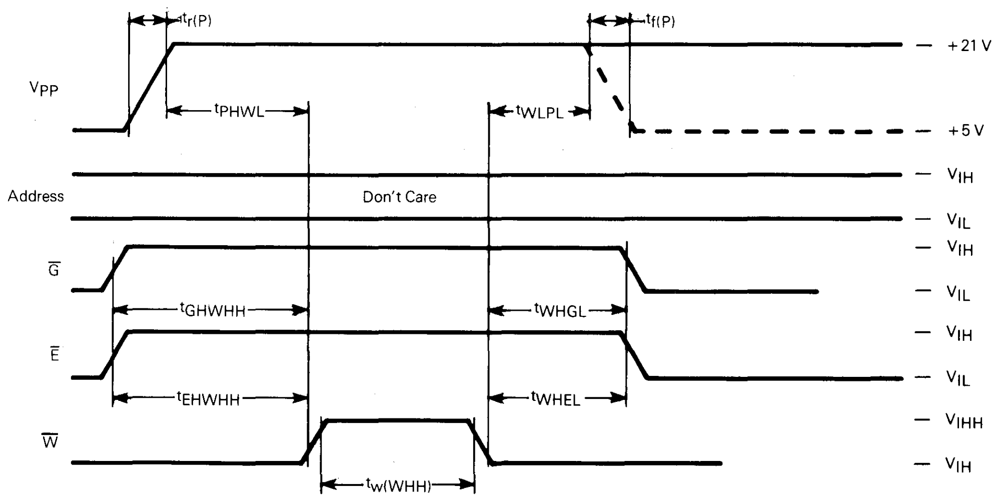
ERASE OPERATION TIMING DIAGRAM



ERASE-OF-REPLACE OPERATION AC TIMING CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 21 ± 1 Vdc, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
V_{PP} Rise Time	$t_{r(P)}$	50	–	–	ns
V_{PP} Fall Time	$t_{f(P)}$	50	–	–	ns
V_{PP} Setup Time	t_{PHWL}	2.0	–	–	μs
V_{PP} Hold Time	t_{WLPL}	2.0	–	–	μs
Output Enable Setup Time	t_{GHWHH}	2.0	–	–	μs
Output Enable Hold Time	t_{WHGL}	2.0	–	–	μs
Chip Enable Setup Time	t_{EHWHH}	2.0	–	–	μs
Chip Enable Hold Time	t_{WHEL}	2.0	–	–	μs
Erase-of-Replace Pulse Width	$t_{w(WHH)}$	10	–	–	ms

ERASE-OF-REPLACE OPERATION TIMING DIAGRAM

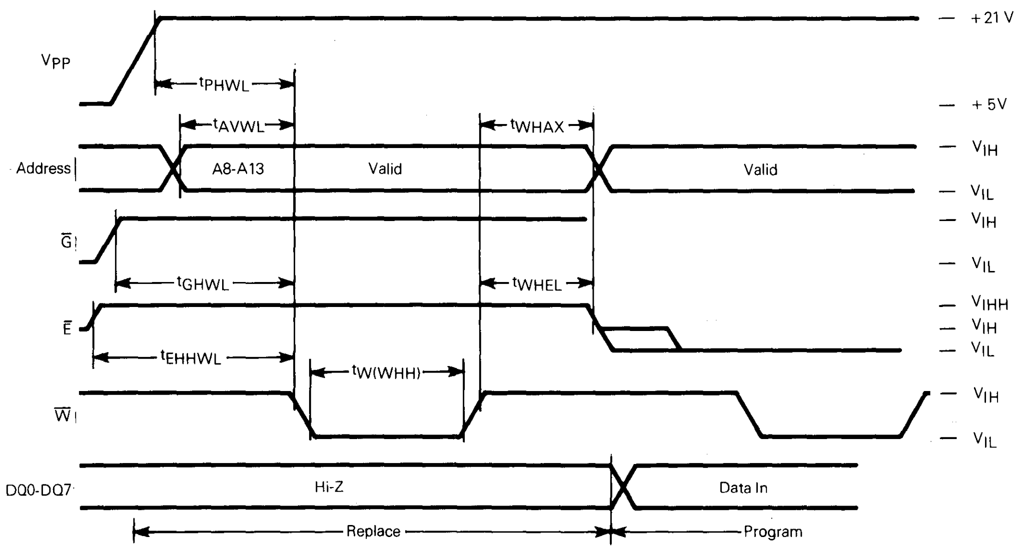


3

REPLACE OPERATION AC TIMING CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, V_{PP} = 21 ± 1 Vdc, T_A = 25°C)

Characteristic	Symbol	Min	Typ	Max	Unit
V_{PP} Setup Time	t_{PHWL}	2.0	–	–	μs
Address Setup Time	t_{AVWL}	2.0	–	–	μs
Address Hold Time	t_{WHAX}	2.0	–	–	μs
Output Enable Setup Time	t_{GHWL}	2.0	–	–	μs
Chip Enable Setup Time	t_{EHHWL}	2.0	–	–	μs
Chip Enable Hold Time	t_{WHEH}	2.0	–	–	μs
Replace Pulse Width	$t_{w(WL)3}$	50	100	–	ms

REPLACE OPERATION TIMING DIAGRAM



FUNCTIONAL DESCRIPTION

INTRODUCTION

The MCM6836( )16 Combination ROM/EEPROM (CREEM) is a 128K bit memory device containing 2K bytes of EEPROM and 14K bytes of mask programmed ROM. The EEPROM is located in the lower 2K byte section of memory, at addresses \$0000 to \$07FF, and the mask ROM is located in the upper 14K byte section of memory at addresses \$0800 to \$3FFF. The MCM6836R16 contains an additional 256 bytes of spare memory. This redundant memory allows for the replacement of a 256 byte block of memory in either mask ROM or EEPROM. The MCM6836E16, without redundancy, is also available. The MCM6836( )16 is contained in a standard 28-pin dual in-line package.

The MCM6836( )16 incorporates several operating modes which make the device easy to use and test. These modes which are illustrated in Figure 3 include: Read, Standby, Program, Erase, Verify, Replace, and Erase-Of-Replace (Replace and Erase-Of-Replace modes are used in the MCM6836R16 only). The pin voltages (signals) required for each mode are also illustrated in Figure 3 and a functional description of each operating mode is provided below. The read and standby modes allow the device to be used as a conventional ROM, the program mode allows programming of individual bytes in the EEPROM, and the erase mode allows the entire EEPROM contents to be erased to the logic high state in approximately 10 milliseconds.

In the MCM6836R16, the replace mode allows substitution of any 256-byte page in the mask ROM or EEPROM memory space with an erased page of EEPROM which can then be programmed. The substitution is performed as a single block of memory, and on-chip logic determines if mask ROM or EEPROM has been replaced. If EEPROM has been replaced, the redundant memory and the memory it has replaced are erased when the standard EEPROM is erased. If the substitution is for mask ROM, the spare memory is erased only by the erase replace mode which has unique control functions. This allows the spare memory to contain the same characteristics as the normal memory for which it is substituted.

OPERATING MODES

The MCM6836E16/MCM6836R16 (CREEM) incorporates five common operating modes, plus two more modes for the MCM6836R16, which make the device easy to use and test. The following paragraphs provide a detailed discussion of each of these modes. In addition, Figure 3 provides a chart illustrating how the various pins are affected during each of the operating modes.

NOTE

It is possible to erase spare EEPROM even if it is used as ROM (or isn't being used) when the following erroneous pin connections are made: E and G = V_{IHH}, V_{PP} = V_{PP}, and W = V_{IL}.

Read Mode — this mode allows the MCM6836( )16 to be used like any conventional mask ROM. In order to read the device in this mode, E and $\overline{G}$ must be held low (V_{IL}), V_{PP} is connected to V_{CC}, and a valid address accessed for data output. The W pin can be in either state (don't care). Some characteristics of the read mode are:

1. Data is available 250 nanoseconds after valid addresses or after the falling edge of $\overline{E}$.
2. Data is valid 100 nanoseconds after the trailing edge of $\overline{G}$ provided $\overline{E}$ and stable addresses have been present for 150 nanoseconds or more.
3. Current is less than 100 milliamperes at 0°C.

Standby Mode — In this mode the MCM6836( )16 is disabled. In order to enter this mode, $\overline{E}$ and $\overline{G}$ must be at a logic high level (V_{IH}), and V_{PP} must be connected to V_{CC}. The $\overline{W}$ and address line can be at any state ("don't care") and the data bus will be in the high-impedance state. (Hi-Z). Some characteristics of the standby mode are:

1. Data outputs are high impedance.
2. Current is reduced 75% to less than 25 milliamperes at 0°C.

Program Mode — In this mode, individual bytes (memory locations) in the EEPROM may be programmed in approximately 10 milliseconds. (A memory location must be erased to the all ones state before it can be programmed.) In order to enter this mode and program the EEPROM, $\overline{E}$ must be at a logic low (V_{IL}), $\overline{G}$ at a logic high (V_{IH}), and V_{PP} must be held at +21 Vdc. A 10 millisecond negative-going pulse on $\overline{W}$ will then allow the input data to be programmed into the addresses accessed in the EEPROM. Some characteristics of the program mode are:

1. Although only zeros are programmed into the device, both ones and zeros can be present in the data word.
2. Requires +21 Vdc programming voltage supply.

FIGURE 3 — OPERATING MODES AND CONTROL VOLTAGES

	$\overline{E}$	$\overline{G}$	V_{PP}	$\overline{W}$	Address	Data
Read	V_{IL}	V_{IL}	V_{CC}	X	Valid	D_{out}
Standby	V_{IH}	V_{IH}	V_{CC}	X	X	Hi-Z
Program	V_{IL}	V_{IH}	V_{PP}	V_{IH}	Valid	D_{in}
Erase	V_{IL}	V_{IHH}	V_{PP}	V_{IH}	X	Hi-Z
Verify	V_{IL}	V_{IL}	V_{PP}	V_{IH}	Valid	D_{out}
Replace#	V_{IHH}	V_{IH}	V_{PP}	V_{IH}	Valid	Hi-Z
Erase-of-Replace#	V_{IH}	V_{IH}	V_{PP}	V_{IHH}	X	Hi-Z

# Indicates used in MCM6836R only.

NOTE: It is possible to erase spare EPROM even if it is used as ROM (or isn't being used) when the following erroneous pin connections are made: $\overline{E}$ and $\overline{G}$ = V_{IHH}, V_{PP} = V_{PP}, and $\overline{W}$ = V_{IL}.

3

Erase Mode — This mode allows the contents of the EEPROM to be erased to all ones. In order to enter this mode and erase the EEPROM, $\overline{E}$ must be held low (V_{IL}), $\overline{G}$ must be held at V_{IHH}, and V_{PP} must be held at +21 Vdc. A 10 millisecond negative-going pulse on $\overline{W}$ will then erase the EEPROM to the all ones state. Address lines can be in any state and the data bus will be in the high-impedance state (Hi-Z). Some characteristics of the erase mode are:

1. Bulk erase returns the entire EEPROM array to all ones.
2. A +21 Vdc programming voltage supply is required.

Verify Mode — In this mode the contents of the EEPROM can be verified as all ones after erasure and the contents of the data byte can be verified after programming. In order to enter this mode and verify EEPROM and/or data byte contents, $\overline{E}$ and $\overline{G}$ must be held at V_{IL}, and V_{PP} must be held at +21 Vdc. The $\overline{W}$ line must be held high (V_{IH}) and a valid address must be applied to the address lines accessing the EEPROM locations (to obtain data output). Some characteristics of the verify mode are:

1. Allows quick verification of the data byte which was written during the previous cycle.
2. Verification may be performed after each program or erase cycle.
3. Verification is accomplished by performing a read cycle with V_{PP} at +21 Vdc and $\overline{W}$ held at V_{IH}.

Replace Mode (MCM6836R16 only) — The replace mode allows for substitution of any 256 byte page in the mask ROM or EEPROM memory with an erased page of EEPROM which can then be programmed. The substitution is performed as a single block of memory and on-chip logic determines if mask ROM or EEPROM is to be replaced. If EEPROM is replaced, the redundant memory and the memory it has replaced is erased when the standard EEPROM is erased. If the substitution is for mask ROM, the spare memory can be erased only in the erase-of-replace mode, which has unique control functions. Thus, the spare memory assumes the same characteristics as the normal memory for which it was substituted.

To replace a block of memory, $\overline{E}$ must be held at V_{IHH}, $\overline{G}$ must be held at V_{IH}, and V_{PP} must be held at +21 Vdc. Then, a 100 millisecond negative-going pulse on $\overline{W}$ will substitute the spare memory when the beginning address of the section of memory to be replaced is set on address lines A8-A13.

The replace operation programs special EEPROM devices which: (1) program replacement addresses into a spare row decoder, (2) determine if the address space is in mask ROM or EEPROM, (3) enable the spare memory, and (4) prevent "overprogramming" the replacement address. Data is then programmed into the spare memory by using the program mode. If this section of memory is addressed during the read or program mode, a signal is generated that disables all normal row decoders.

Some characteristics of the replace mode are:

1. Substitutes 256 bytes of spare EEPROM for 256 bytes of either mask ROM or EEPROM.
2. Performed as a single block of memory.
3. On-chip logic determines if mask ROM or EEPROM is to be replaced.
4. When in the replace mode, special EEPROM devices are programmed which:
   A. Program replacement addresses into a spare row decoder,
   B. Determine if the address space is in mask ROM or EEPROM,
   C. Enable the spare memory, and
   D. Prevent "overprogramming" the replacement address.

Data is then programmed into spare memory using the program mode.

Erase-Of-Replace Mode (MCM6836R16 only) — This mode is used, when spare memory (redundancy) is being used, to erase the replace mask ROM. To erase the spare memory to all ones, $\overline{E}$ and $\overline{G}$ must be held at V_{IH}, and V_{PP} must be held at +21 Vdc. Then, a 10 millisecond positive-going (to V_{IHH}) pulse on $\overline{W}$ will erase the spare memory to the all ones state. This mode also erases the programmed address to the redundancy EEPROM. During the erase-of-replace mode, the address lines can be at any state and the data bus is in the high-impedance state. Some characteristics of the erase-of-replace mode are:

1. Returns the device to its original condition by erasing the replace circuitry, spare decoder, and spare memory.
2. Needed only for a device which contains redundancy as a user option.
3. False erasure of redundancy memory is unlikely due to unique control function ($\overline{W}$ pulse).

NOTE

The erase-of-replace mode need only be used if spare memory is being used to replace a section of mask ROM. This operation erases the replacement circuitry, spare decoder, and spare memory after which the device is returned to its original condition.

FUNCTIONAL PIN DESCRIPTION

V_{PP}

This pin is used as the +21 Vdc input voltage during EEPROM programming and erasing operations. It is connected to V_{CC} in the normal operating read and standby modes. V_{PP} should not, in any case, be applied before the device has been powered by V_{CC} or after V_{CC} has been removed from the device.

WRITE ($\overline{W}$)

The active low state (V_{IL}) of this input pin is used to program and erase the EEPROM. It is also used as a mode select pin for the erase-of-replace mode when V_{IHH} is applied to its input. In the normal read and standby operating modes, this pin is a "don't care".

CHIP ENABLE ($\overline{E}$)

The active low state (V_{IL}) of this input pin is used as a chip select signal for the read, program, erase, and verify operating modes. It is also used as a mode select input signal for the replace mode when V_{IHH} is applied. It is used as a mode select signal for the standby and erase-of-replace modes when V_{IH} is applied.

OUTPUT ENABLE ($\overline{G}$)

The active low state (V_{IL}) of this input pin is used in conjunction with $\overline{E}$ to enable the output buffer of this device. It is also used as a mode select signal for the erase mode when V_{IHH} is applied.

DATA BUS (DQ0-DQ7)

These eight pins provide a bidirectional data link to the system bus.

ADDRESS INPUTS (A0-A13)

These 14 address inputs allow any of the 14K bytes of mask ROM and 2K bytes of EEPROM to be uniquely selected in the read mode. Addresses \$0000 to \$07FF are designated as EEPROM, and addresses \$0800 to \$3FFF are designated as the mask programmable ROM. These address inputs are also used to select an address byte for programming, verifying, and replacing.

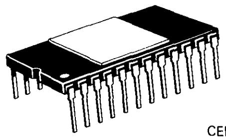
MOTOROLA

MC6839

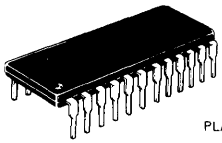
Advance Information

MOS
(N-CHANNEL, SILICON-GATE)

FLOATING-POINT READ-ONLY MEMORY



L SUFFIX
FRIT-SEAL
CERAMIC PACKAGE
CASE 716



P SUFFIX
PLASTIC PACKAGE
CASE 709

FLOATING-POINT ROM

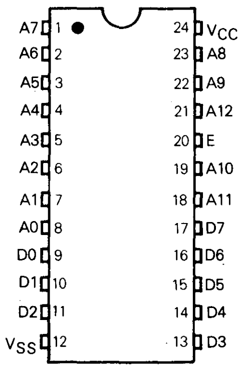
The MC6839 standard product ROM provides floating point capability for the MC6809 or MC6809E MPU. The MC6839 implements the entire *IEEE Proposed Standard for Binary Floating Point Arithmetic Draft 8.0,* providing a simple, economical and reliable solution to a wide variety of numerical applications. The single- and double-precision formats provide results which are bit-for-bit reproducible across all Draft 8.0 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 8.0
  - Includes All Precisions, Modes, and Operations Required or Suggested by the Standard
  - Single, Double, and Extended Formats
  - Includes the Following Operations:

Add	Absolute Value
Subtract	Negate
Multiply	Predicate Compares
Divide	Condition Code Compares
Remainder	Convert Integer↔Floating Point
Square Root	Convert Binary Floating Point↔Decimal String
Integer Part	

PIN ASSIGNMENT

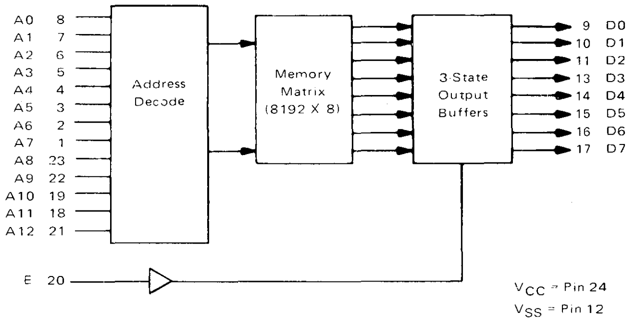


ORDER INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic	1.5	0°C to 70°C	MC68A39L
L Suffix	2.0	0°C to 70°C	MC68B39L
Plastic	1.5	0°C to 70°C	MC68A39P
P Suffix	2.0	0°C to 70°C	MC68B39P

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

BLOCK DIAGRAM



3

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

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}).

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

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
Hi-Z Output Leakage Current (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

TIMING PARAMETER ABBREVIATIONS

t X X X X

signal name from which interval is defined
transition direction for first signal
signal name to which interval is defined
transition direction for second signal

The transition definitions used in this data sheet are:

H = transition to high
L = transition to low
V = transition to valid
X = transition to invalid or don't care
Z = transition to off (high impedance)

TIMING LIMITS

The table of timing values shows either a minimum or a maximum limit for each parameter. Input requirements are specified from the external system point of view. Thus, address setup time is shown as a minimum since the system must supply at least that much time (even though most devices do not require it). On the other hand, responses from the memory are specified from the device point of view. Thus, the access time is shown as a maximum since the device never provides data later than that time.

WAVEFORMS

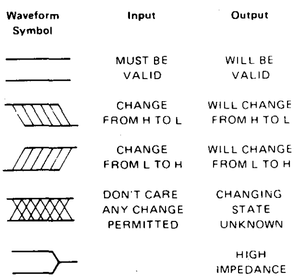
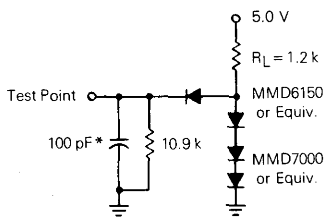
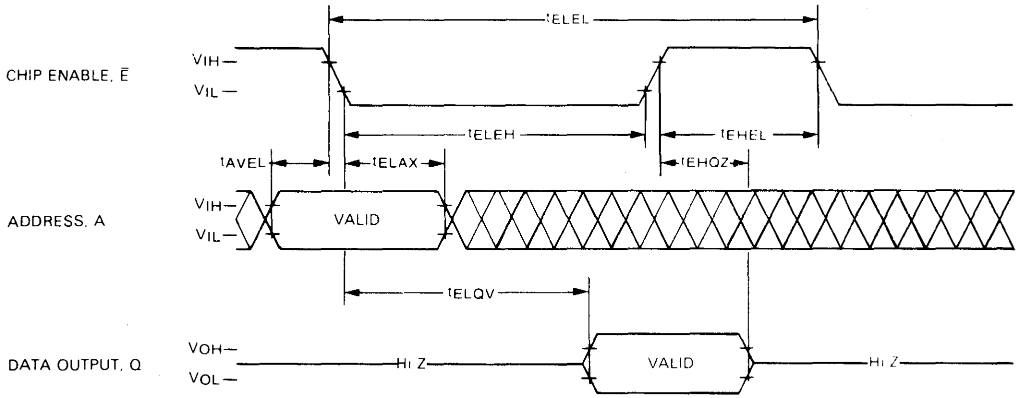


FIGURE 1 — AC TEST LOAD



* Includes Jig Capacitance

FIGURE 2 — TIMING DIAGRAM



NOTES:

1. Voltage levels shown are $V_L \leq 0.4$, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise noted.

INTRODUCTION

Since the earliest days of computers it 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 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:

1	←8→	←23 bits→
s	exponent	significand

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 =	1.0×2^0 =	$3F	80	00	00
+3.0 =	1.5×2^1 =	$40	40	00	00
−1.0 =	-1.0×2^0 =	$BF	80	00	00

DOUBLE FORMAT

All double precision numbers are represented by an 8-byte string as:

1	←11 bits→	←52 bits→
s	exponent	significand

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 =	-1.875×2^4 =	$C0	3E	00	00	00	00	00	00
0.25 =	1.0×2^{-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:

1	←15 bits→	←64 bits→
s	exponent	1. significand

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 =	1.0×2^{-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 … 4	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 special 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 $0 \leq remainder < arg2$. 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.

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).

3

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.

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
```

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.

3

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. significand)
Double: $X = (-1)^s \times 2^{-1022} \times$ (0. 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 FPREG 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 # significant digits requested)	35,000 – 48,000	67,000 – 85,000	–

3

MOTOROLA

MC6840

PROGRAMMABLE TIMER MODULE (PTM)

The MC6840 is a programmable subsystem component of the M6800 family designed to provide variable system time intervals.

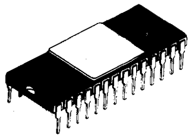
The MC6840 has three 16-bit binary counters, three corresponding control registers, and a status register. These counters are under software control and may be used to cause system interrupts and/or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, 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 ($\overline{\text{IRQ}}$) Output to MPU
- Readable Down Counter Indicates Counts to Go Until Time-Out
- Selectable Gating for Frequency or Pulse-Width Comparison
- $\overline{\text{RESET}}$ Input
- Three Asynchronous External Clock and Gate/Trigger Inputs Internally Synchronized
- Three Maskable Outputs

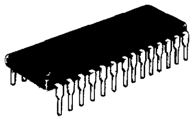
MOS

(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

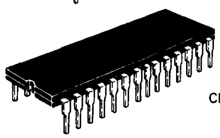
PROGRAMMABLE TIMER



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710

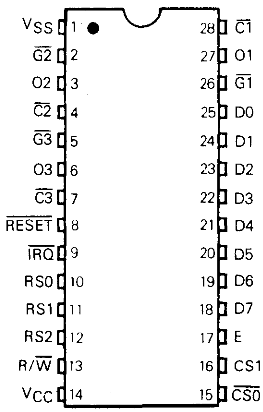


S SUFFIX
CERDIP PACKAGE
CASE 733

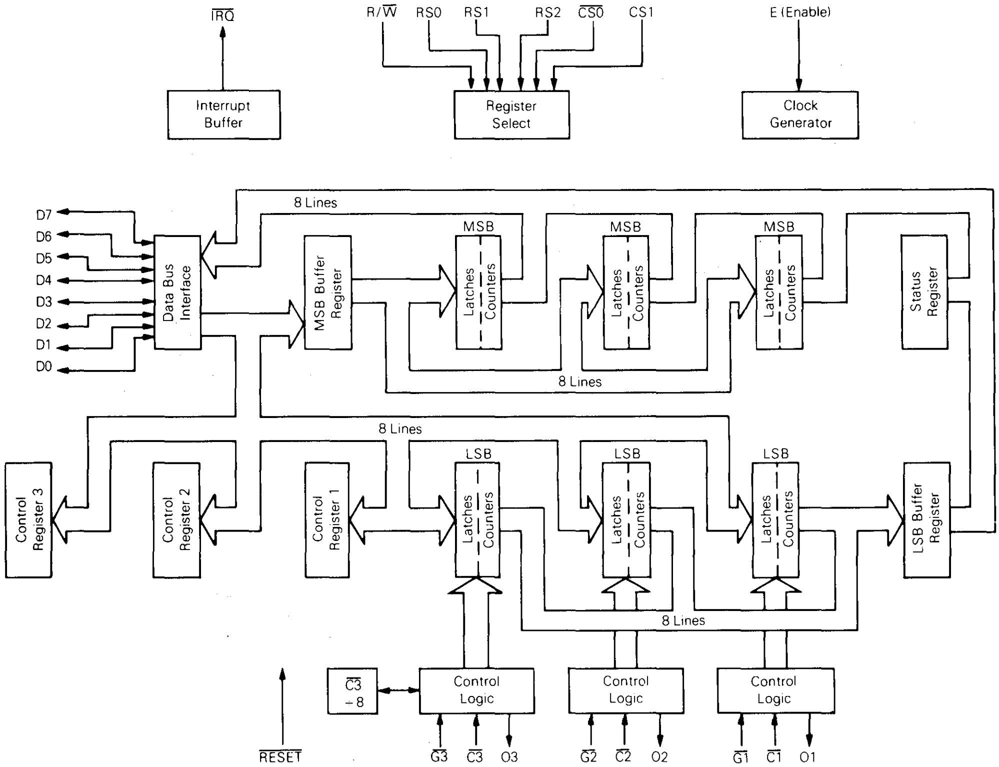
ORDERING INFORMATION

Package Type	Frequency	Temperature Range	Order Number
Ceramic Side Brazed L Suffix	1.0 MHz	0°C to 70°C	MC6840L
	1.0 MHz	−40°C to +85°C	MC6840CL
	1.5 MHz	0°C to 70°C	MC68A40L
	1.5 MHz	−40°C to +85°C	MC68A40CL
	2.0 MHz	0°C to 70°C	MC68B40L
Plastic P Suffix	1.0 MHz	0°C to 70°C	MC6840P
	1.0 MHz	−40°C to +85°C	MC6840CP
	1.5 MHz	0°C to 70°C	MC68A40P
	1.5 MHz	−40°C to +85°C	MC68A40CP
	2.0 MHz	0°C to 70°C	MC68B40P
Cerdip S Suffix	1.0 MHz	0°C to 70°C	MC6840S
	1.0 MHz	−40°C to +85°C	MC6840CS
	1.5 MHz	0°C to 70°C	MC68A40S
	1.5 MHz	−40°C to +85°C	MC68A40CS
	2.0 MHz	0°C to 70°C	MC68B40S

PIN ASSIGNMENT



BLOCK DIAGRAM



POWER CONSIDERATIONS

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

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

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.

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. 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}).

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
Hi-Z (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) (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)	 $\overline{IRQ}$, D0-D7 O1-O3	V_{OL}	 – –	 – –	 $V_{SS}+0.4$ $V_{SS}+0.4$	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}	–	470	700	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
Output Capacitance (V_{in}=0, T_A=25°C, f=1.0 MHz)	 $\overline{IRQ}$ O1, O2, O3	C_{out}	 – –	 – –	 5.0 10	pF

AC OPERATING CHARACTERISTICS (See Figures 2-7)

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 (V_{OH}=2.4 V, Load D) MOS (V_{OH}=0.7 V_{DD}, Load D) CMOS	 t_{co} t_{cm} t_{cmos}	 – – –	 700 450 2.0	 – – –	 460 450 1.35	 – – –	 340 340 1.0	 ns ns μs
Interrupt Release Time	t_{IR}	–	1.2	–	0.9	–	0.7	μs

*t_r and $t_f \leq t_{cycE}$

3

BUS TIMING CHARACTERISTICS (See Notes 1, 2, and 3)

Ident. Number	Characteristic	Symbol	MC6840 Min	MC6840 Max	MC68A40 Min	MC68A40 Max	MC68B40 Min	MC68B40 Max	Unit
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).

NOTES:
1. Not all signals are applicable to every part.
2. Voltage levels shown are $V_L \le 0.4$ V, $V_H \ge 2.4$ V, unless otherwise specified.
3. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

3

FIGURE 1 — BUS TIMING

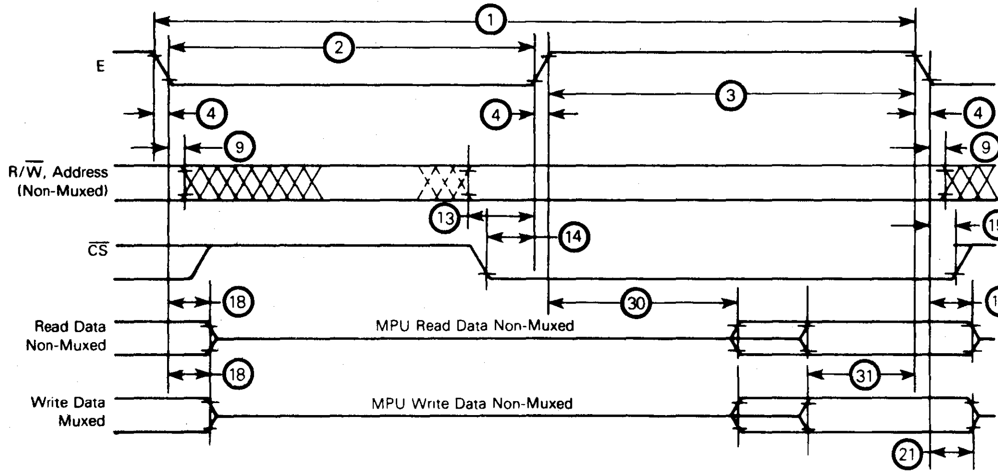


FIGURE 2 — INPUT PULSE WIDTH LOW

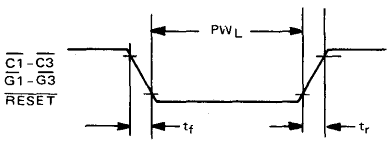


FIGURE 3 — INPUT PULSE WIDTH HIGH

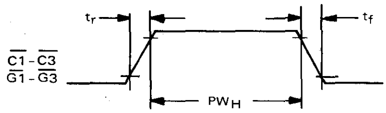


FIGURE 4 — INPUT SETUP AND HOLD TIMES

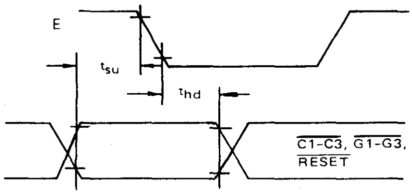


FIGURE 5 — OUTPUT DELAY

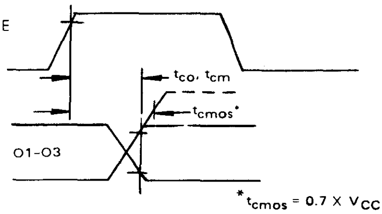


FIGURE 6 — IRQ RELEASE TIME

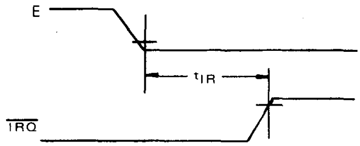


FIGURE 7 — $\overline{C3}$ INPUT SYNCHRONIZATION TIME (÷8 PRESCALER MODE ONLY)

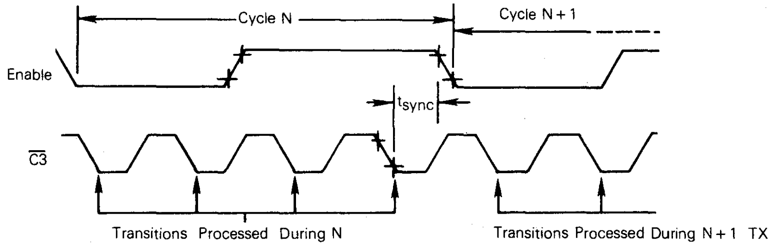
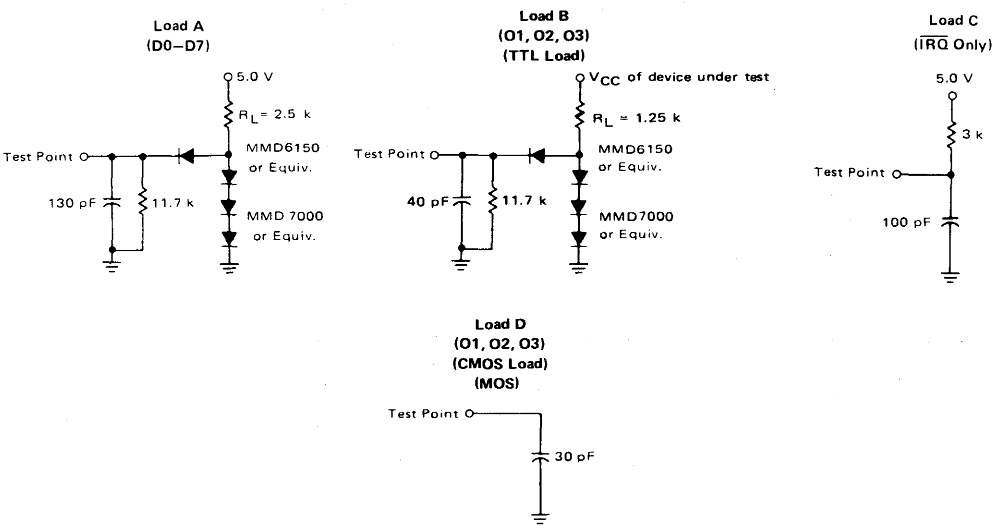


FIGURE 8 — BUS TIMING TEST LOADS



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. 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/$\overline{\text{W}}$ 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.

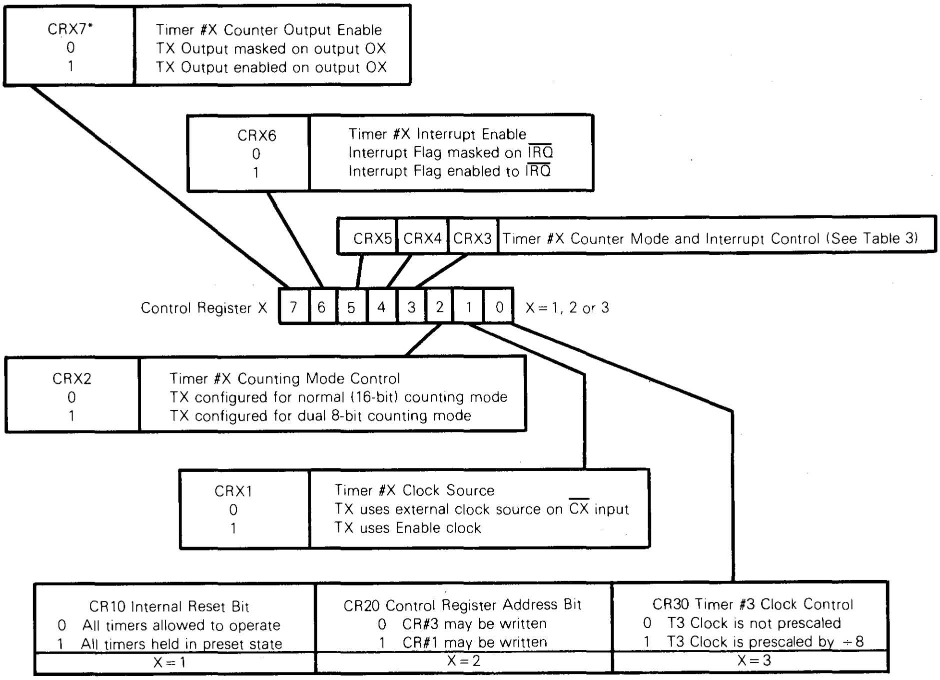
3

NOTE

When initializing Timer 3 into the divide-by-eight mode on consecutive E-cycles (i.e., with DMA), Control Register 3 must be initialized before Timer Latch #3 to insure proper timer initialization.

CR30 — The functions depicted in the foregoing discussions are tabulated in Table 2 for ease of reference.

TABLE 2 — CONTROL REGISTER BITS



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 $\overline{RESET}$ = 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 Figure 10 regarding the binary number 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 $\overline{RESET}$ 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
1	0	0	Frequency Comparison Mode: Interrupt If Gate ↑ ↓ is< Counter Time Out
0	1	0	Continuous Operating Mode: Gate ↓ or Reset Causes Counter Initialization
1	1	0	Pulse Width Comparison Mode: Interrupt if Gate ↑ ↑ is< Counter Time Out
0	0	1	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
0	1	1	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 ($\overline{RESET}=0$ 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

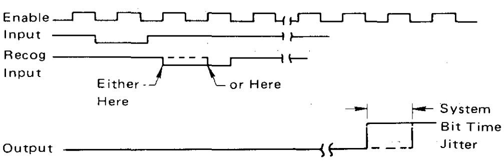
Each of the three timers within the PTM has external clock and gate inputs as well as a counter output line. The inputs are high-impedance, TTL-compatible lines and ouputs are capable of driving two standard TTL loads.

CLOCK INPUTS ($\overline{C1}$, $\overline{C2}$, and $\overline{C3}$) — Input pins $\overline{C1}$, $\overline{C2}$, and $\overline{C3}$ 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 9.

FIGURE 9 — INPUT JITTER



CLOCK INPUT $\overline{C3}$ (÷8 PRESCALER MODE) — External clock input $\overline{C3}$ 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 ($\overline{C3}$) 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 $\overline{C3}$ transition and the falling edge of Enable (see Figure 9). If the synchronization time requirement is not met, it is possible that the $\overline{C3}$ 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 ($\overline{G1}$, $\overline{G2}$, $\overline{G3}$) — Input pins $\overline{G1}$, $\overline{G2}$, and $\overline{G3}$ 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 $\overline{G3}$ 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.

3

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 10 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

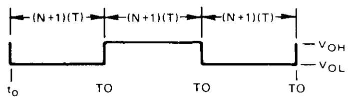
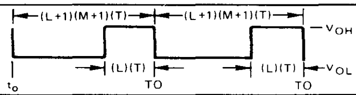
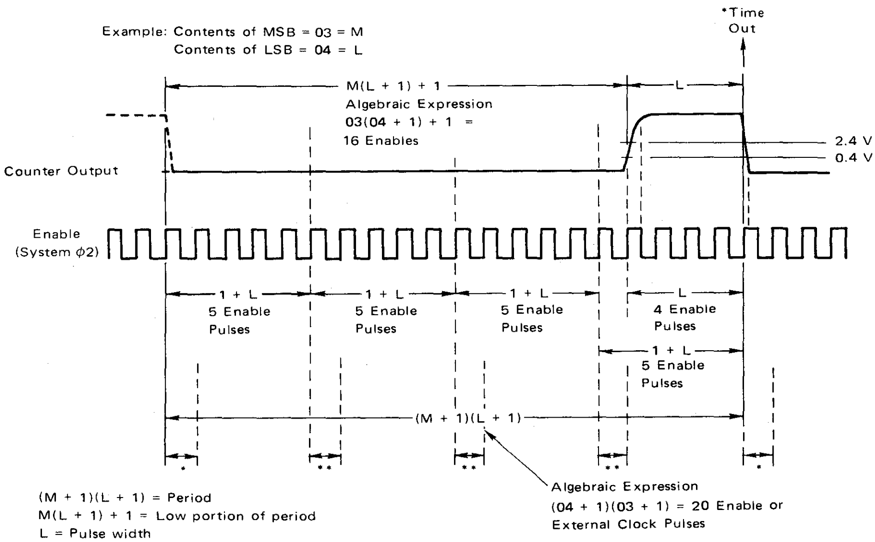
Synthesis Modes		CONTINUOUS MODE (CRX3 = 0, CRX5 = 0)	
Control Register		Initialization/Output Waveforms	
CRX2	CRX4	Counter Initialization	*Timer Output (OX) (CRX7 = 1)
0	0	$\overline{G}\downarrow$+W+R	
0	1	$\overline{G}\downarrow$+R	
1	0	$\overline{G}\downarrow$+W+R	
1	1	$\overline{G}\downarrow$+R	

FIGURE 10 — TIMER OUTPUT WAVEFORM EXAMPLE
(Continuous Dual 8-Bit Mode Using Internal Enable)



*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.

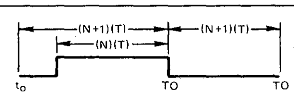
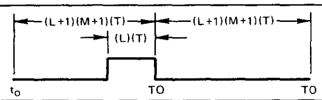
SINGLE-SHOT TIMER MODE — This mode is identical to the Continuous Mode with three exceptions. The first of these is obvious from the name — the output returns to a low level after the initial Time Out and remains low until another Counter Initialization cycle occurs.

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 remaining 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, an 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 $\overline{G}$ 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 terminates 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	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
Comparison	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	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
Comparison	1	1	1	$\overline{G\downarrow}\cdot\overline{I}+\overline{R}$	$\overline{G\downarrow}\cdot\overline{W}\cdot\overline{R}\cdot\overline{I}$	$W+R+I+G$	TO Before $\overline{G\uparrow}$

$\overline{G\downarrow}$ = Negative transition of Gate input.
W = Write Timer Latches Command.
R = Timer Reset (CR10 = 1 or External $\overline{RESET}$ = 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.

3

MOTOROLA

MC6844

MOS

(N-CHANNEL, SILICON-GATE)

DIRECT MEMORY ACCESS CONTROLLER (DMAC)

3

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.

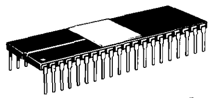
The bus interface of the MC6844 includes select, read/write, interrupt, transfer request/grant, a data port, and an address port which allow data transfer over an 8-bit bidirectional data bus. The funtional configuration of the DMAC is programmed via the data bus. The internal structure provides for 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.

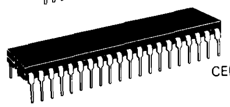
Typical MC6844 applications are a Floppy Disk Controller (FDC) and an Advanced Data Link Controller (ADLC) DMA interface.

MC6844 features include:

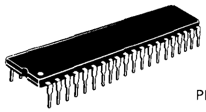
- Four DMA Channels, Each Having a 16-Bit Address Register and a 16-Bit Byte Count Register
- 2 M Byte/Sec Maximum Data Transfer Rate
- Selection of Fixed or Rotating Priority Service Control
- Separate Control Bits for Each Channel
- Data Chain Function
- Address Increment or Decrement Update
- Programmable Interrupts and DMA End to Peripheral Controllers



L SUFFIX
CERAMIC PACKAGE
CASE 715

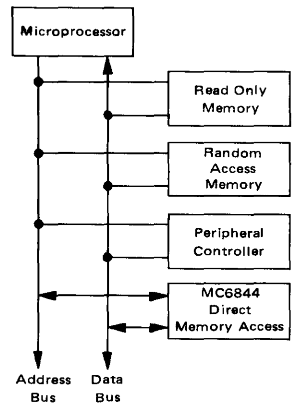


S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711

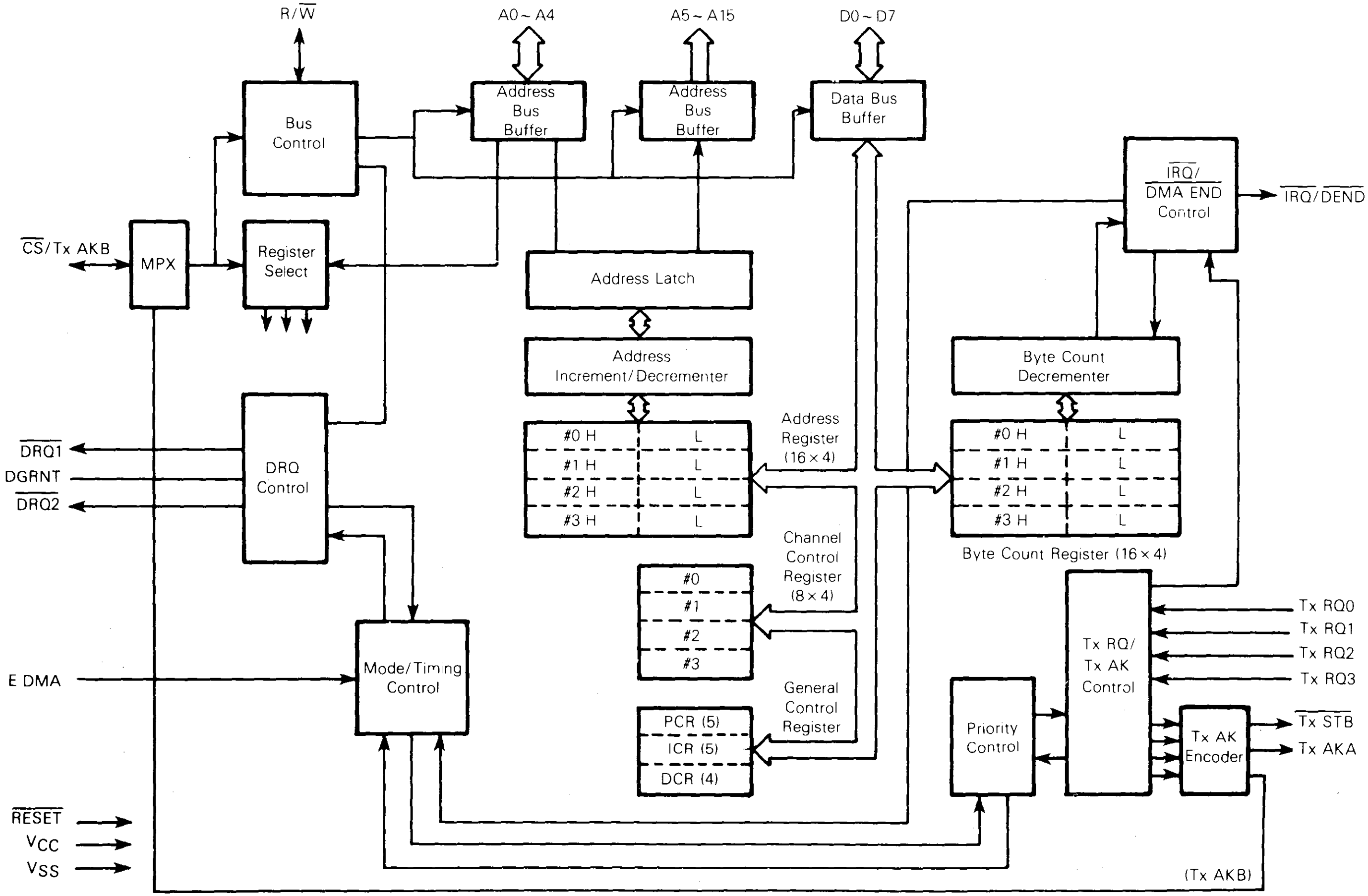
FIGURE 1 — M6800 MICROCOMPUTER FAMILY BLOCK DIAGRAM



PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	E
$\overline{CS}$/Tx AKB	2	39	$\overline{RESET}$
R/$\overline{W}$	3	38	DGRNT
A0	4	37	$\overline{DRQ1}$
A1	5	36	$\overline{DRQ2}$
A2	6	35	Tx AKA
A3	7	34	$\overline{Tx\ STB}$
A4	8	33	$\overline{IRQ}/\overline{DEND}$
A5	9	32	Tx RQ0
A6	10	31	Tx RQ1
A7	11	30	Tx RQ2
A8	12	29	Tx RQ3
A9	13	28	D0
A10	14	27	D1
A11	15	26	D2
A12	16	25	D3
A13	17	24	D4
A14	18	23	D5
A15	19	22	D6
V_{CC}	20	21	D7

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

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

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

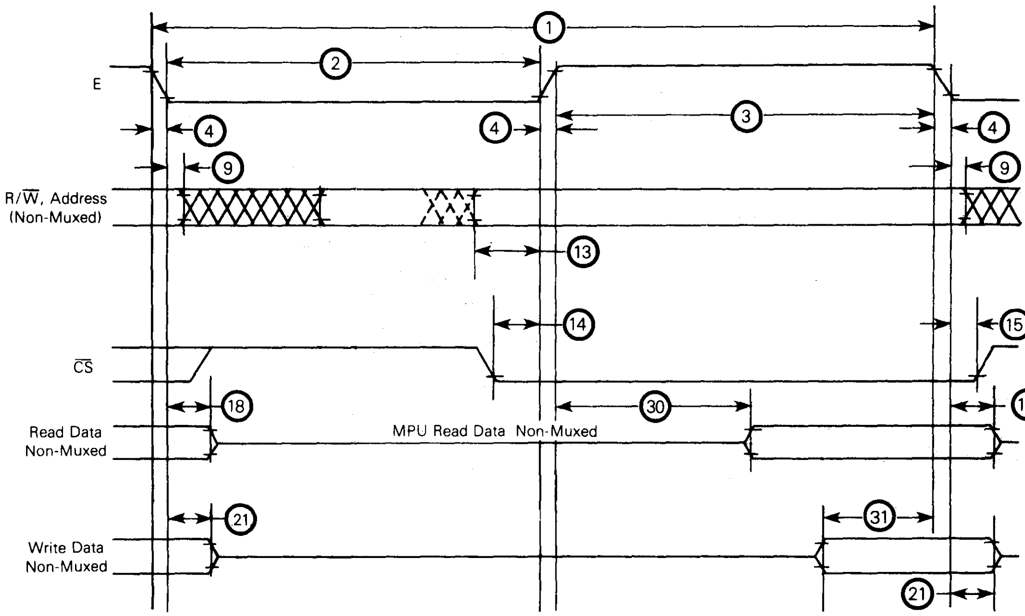
Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	All Inputs	V_{IH}	V_{SS}+2.0	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$/Tx AKB Other Inputs	V_{IL}	V_{SS}−0.3 V_{SS}−0.3	– –	V_{SS}+0.6 V_{SS}+0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	Tx RQ0-3, E, $\overline{RESET}$, DGRNT	I_{in}	–	–	2.5	μA
Hi-Z 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 = 0°C)		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

*For temperatures less than T_A = 0°C, P_{INT} maximum will increase.

MPU MODE TIMING (See Notes 1 and 2)

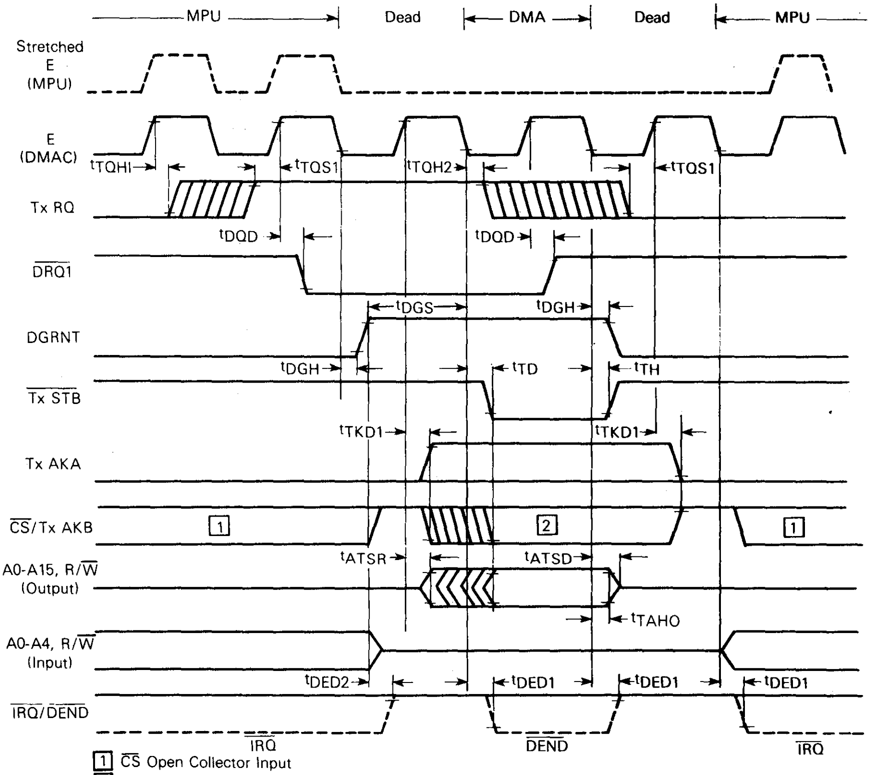
Ident. Number	Characteristic	Symbol	MC6844 Min	MC6844 Max	MC68A44 Min	MC68A44 Max	MC68B44 Min	MC68B44 Max	Unit
1	Cycle Time	t_{cyc}	1.0	10	0.67	10	0.5	10	µs
2	Pulse Width, E Low	PW_{EL}	430	9500	280	9500	210	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	280	9500	220	9500	ns
4	Clock Rise and Fall Time	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)

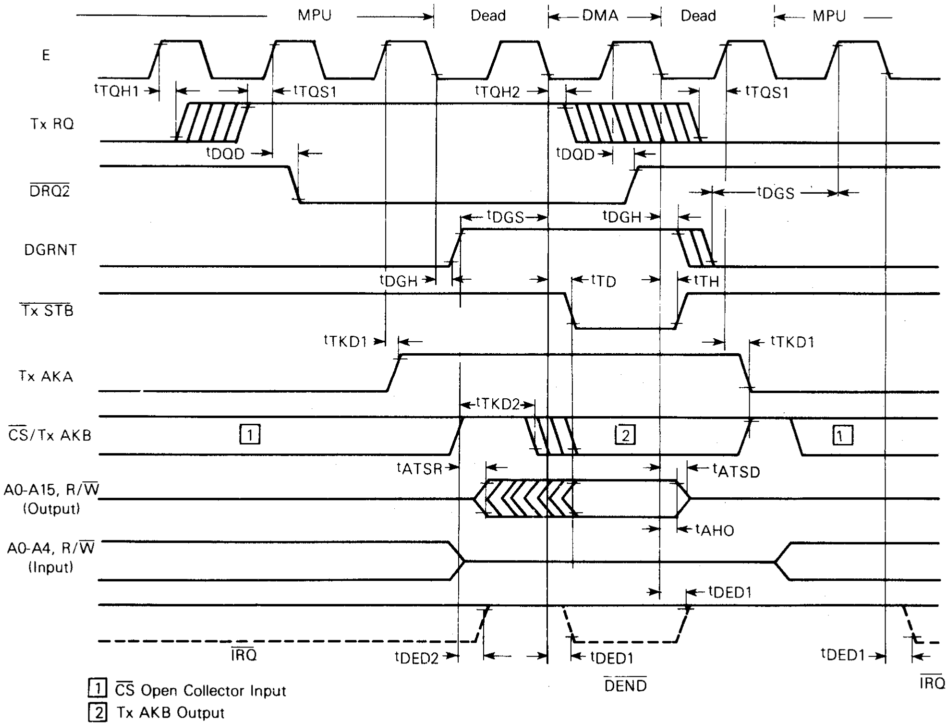
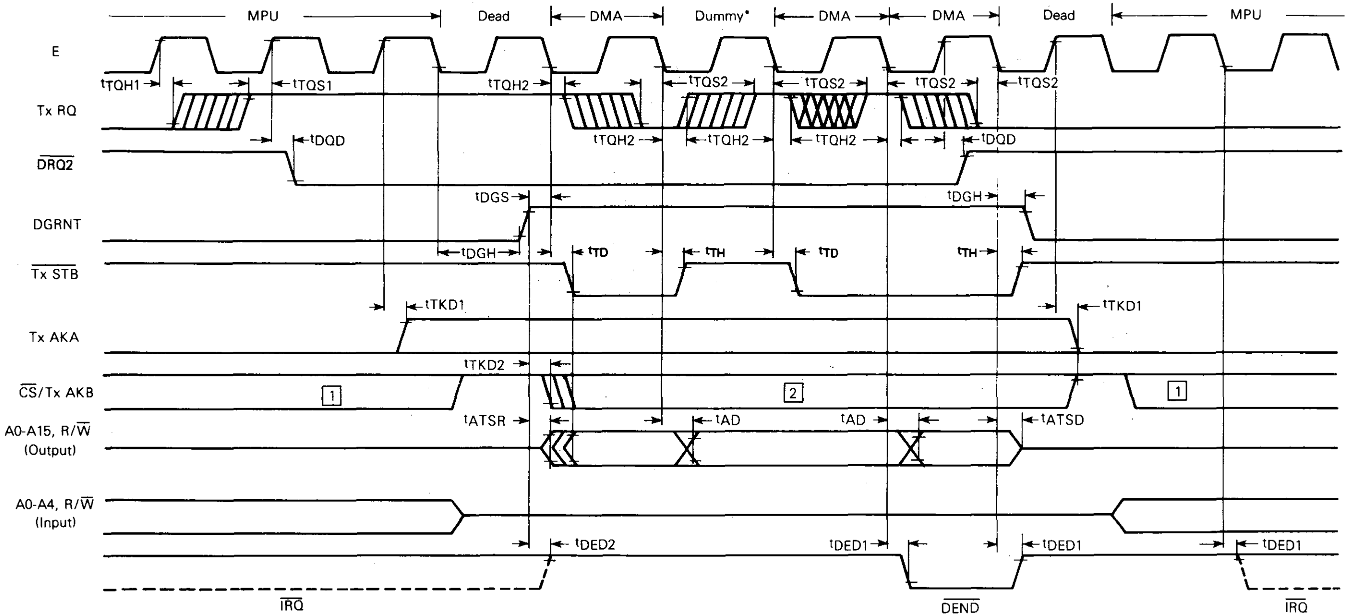


**FIGURE 6 — MODE 3 TIMING
(HALT BURST MODE)**



1 $\overline{CS}$ Open Collector Input

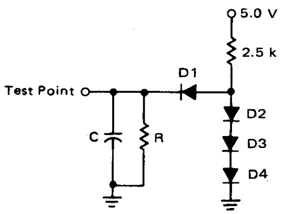
2 Tx AKB Output

*No transfer (dummy cycle) because Tx RQ was negated at start of E cycle.

DMA TIMING (Load Condition Figure 7)

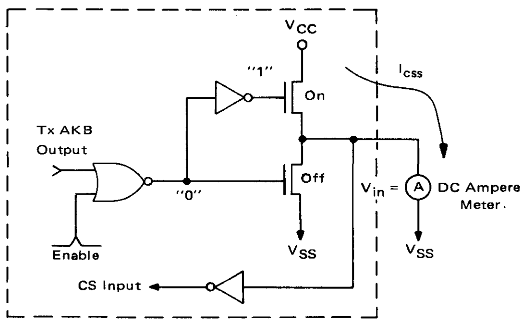
Characteristic		Symbol	MC6844 Min	MC6844 Max	MC68A44 Min	MC68A44 Max	MC68B44 Min	MC68B44 Max	Unit
Tx RQ Setup Time									
E Rising Edge		t_{TQS1}	120	–	120	–	120	–	ns
E Falling Edge		t_{TQS2}	210	–	210	–	170	–	
Tx RQ Hold Time									
E Rising Edge		t_{TQH1}	20	–	10	–	10	–	ns
E Falling Edge		t_{TQH2}	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}	30	–	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



3

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 $\overline{IRQ}$ interrupt by each channel at the end of a data block transfer and sets a flag when $\overline{IRQ}$ 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 $\overline{RESET}$ 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/$\overline{W}$. The DMAC also generates $\overline{Tx\ STB}$, 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; $\overline{DRQ1}$ and $\overline{DRQ2}$, which are used for handshaking with the system MPU; $\overline{DEND}$, which is asserted when the last byte of a data block is being transferred; and $\overline{IRQ}$, 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 ($\overline{DRQ1}$ or $\overline{DRQ2}$). 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.

$\overline{RESET}$

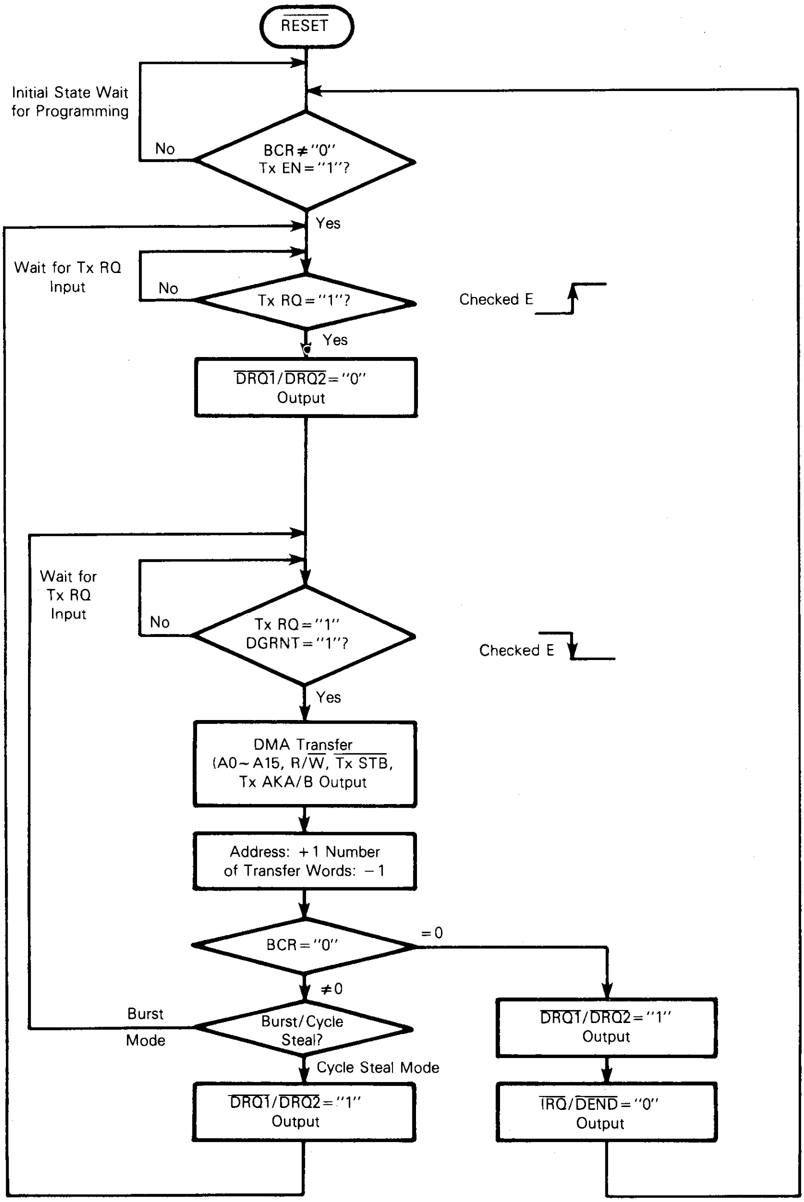
This input is used to place the DMAC into a known state and provide for an orderly startup procedure. Assertion of $\overline{RESET}$ clears all internal registers except the address and the byte count registers (see Hardware Initialization).

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



3

3

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

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 ($\overline{IRQ}$/$\overline{DEND}$)

Interrupt Request/$\overline{DMA\ End}$ 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. $\overline{DEND}$ 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). $\overline{IRQ}$ 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, $\overline{IRQ}$ 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 ($\overline{DRQ1}$, $\overline{DRQ2}$)

These active low TTL-compatible outputs are used by the DMAC to handshake with the MPU in requesting the system bus for DMA operation. $\overline{DRQ1}$ is asserted to indicate that a channel configured for mode 1 operation requires servicing, and $\overline{DRQ2}$ 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 ($\overline{Tx\ STB}$)

$\overline{Tx\ STB}$ is asserted during each DMA transfer cycle and can be used as a transfer acknowledge for peripheral controllers and as a system VMA. $\overline{Tx\ STB}$ 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 ($\overline{CS}$/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

$\overline{CS}$/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/Ready Flag	Not Used	Not Used	Address $\overline{\text{Up}}$/Down	MCA	MCB	Read/$\overline{\text{Write}}$ ($R/\overline{W}$)
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 $R/\overline{W}$ 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 $R/\overline{W}$ 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 decrements with each DMA cycle; if it is clear, the address register increments 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 $\overline{IRQ}$ 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 $\overline{IRQ}$ 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 $\overline{IRQ}$ is asserted. Interrupt control register bits are defined as follows:

Bits 0-3 DIE0-3 — DEND IRQ Enable. These bits enable individual channel $\overline{IRQ}$ 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 $\overline{IRQ}$ 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 $\overline{IRQ}$ is asserted, indicating the end of a channel block transfer (DEND assertion) with interrupt enabled. This flag is cleared and $\overline{IRQ}$ is negated by a read of the channel control register of the channel causing the $\overline{IRQ}$ 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.

$\overline{\text{IRQ}}$, DEND, Tx AK GENERATION

Derivation of $\overline{\text{IRQ}}$ (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 $\overline{\text{IRQ}}$/DEND DMAC output to ensure that the system $\overline{\text{IRQ}}$ is asserted at the proper time, only during MCU operation. Whenever DMA GRANT is high, $\overline{\text{IRQ}}$ is negated.

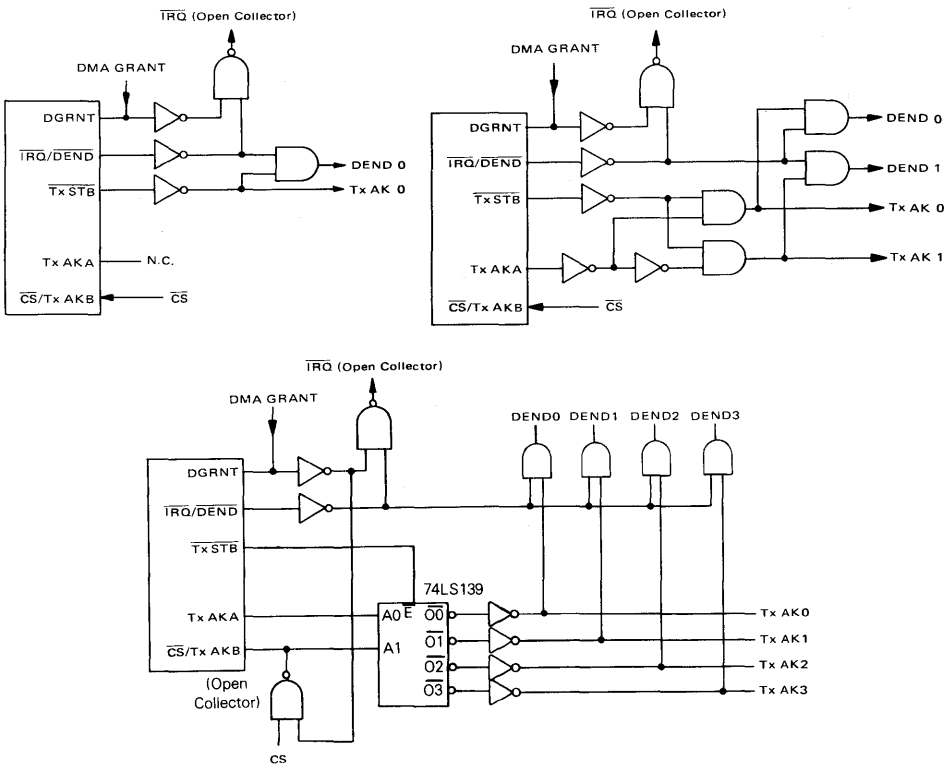
The circuits also generate DEND and Tx AK for the proper channel, gated by Tx STB.

The one-channel DMA mode requires no channel decoding, so for this mode Tx AK is derived from $\overline{\text{Tx STB}}$ directly, and $\overline{\text{Tx STB}}$ is used to demultiplex the $\overline{\text{IRQ}}$/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 $\overline{\text{CS}}$/Tx AKB pin is bidirectional during four-channel operation, so an open collector gate must be used to drive $\overline{\text{CS}}$ in order to avoid drive contention.

FIGURE 10 — $\overline{\text{IRQ}}$, $\overline{\text{DEND}}$, 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, $\overline{\text{BGACK}}$ output is high, and $\overline{\text{Tx STB}}$ is high. The MC6844 responds to a Tx RQ by asserting $\overline{\text{DRQH}}$. Assertion of Tx RQ also asserts MC68000 $\overline{\text{BR}}$. For DMA transfer, two conditions must be met: 1) DMAC $\overline{\text{DRQH}}$ must be asserted and 2) all bus masters must relinquish the system bus. Once $\overline{\text{DRQH}}$ 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 $\overline{\text{BGACK}}$, $\overline{\text{AS}}$, and $\overline{\text{DTACK}}$ after the MC68000 asserts $\overline{\text{BG}}$ in response to a bus request.

When both conditions are met, the NAND flip-flop is set by assertion of LS138 $\overline{\text{O3}}$, asserting DGRNT and $\overline{\text{BGACK}}$. 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 $\overline{\text{Tx STB}}$ 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 $\overline{\text{BR}}$ to the MC68000 is negated when $\overline{\text{BGACK}}$ 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 $\overline{\text{Tx STB}}$ and the MPU VMA line.

MC6844/MC6809 BUS ARBITRATION INTERFACE

An MC6844/MC6809 interface is presented in Figure 13. This circuit ensures that MC6809 $\overline{\text{DMA/BREQ}}$ 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 $\overline{\text{VMA}}$.

The MC6809 does not generate a $\overline{\text{VMA}}$ output since the only invalid address asserted by the MPU is $FFFF with R/$\overline{\text{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 $\overline{\text{VMA}}$ 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, $\overline{\text{DRQ1/2}}$ 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 $\overline{\text{DRQ1/2}}$ 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 $\overline{\text{DMA/BREQ}}$. The output of U4 at this time is a low, since both of the U4 inputs are low.

FIGURE 11 — MC68000/MC6844 INTERFACE

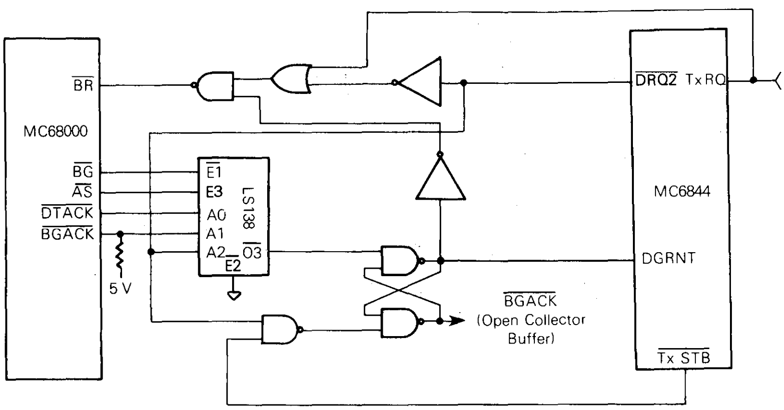


FIGURE 12 — MC6800/MC6844 INTERFACE

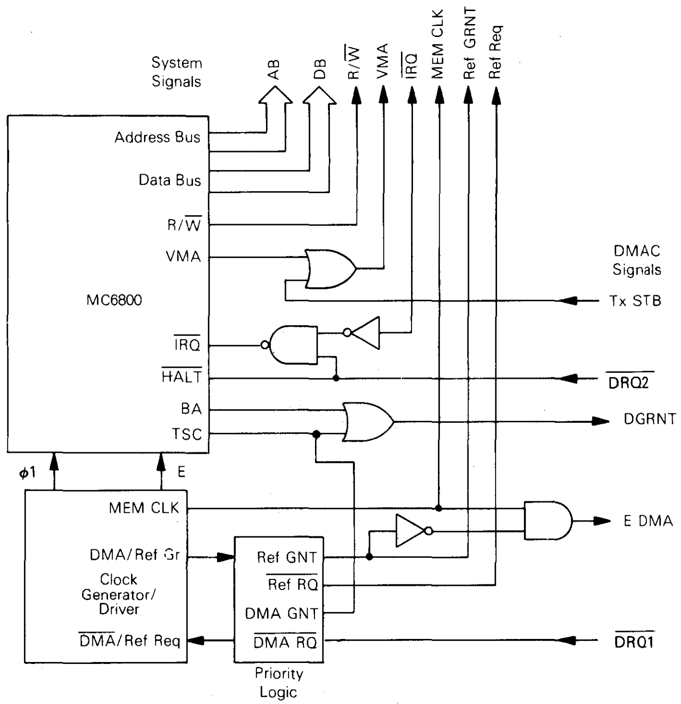
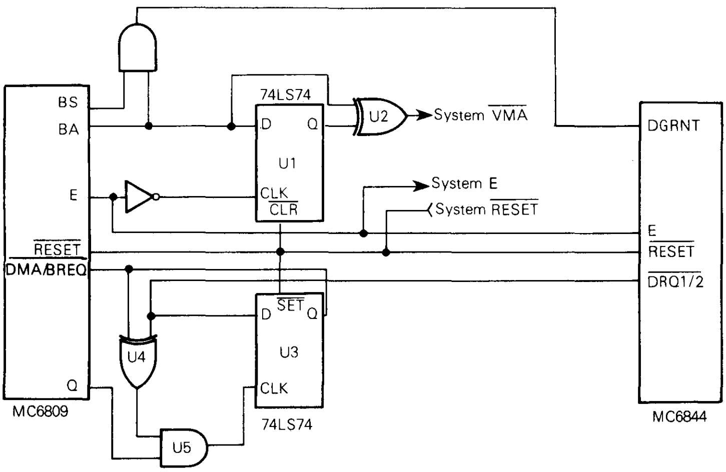


FIGURE 13 — MC6844/MC6809 INTERFACE



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 ($\overline{VMA}$) is low ($\overline{VMA}$ 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.

ORDERING INFORMATION

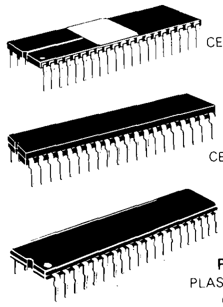
Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic	1.0	0°C to 70°C	MC6844L
L Suffix	1.0	−40°C to 85°C	MC6844CL
	1.5	0°C to 70°C	MC68A44L
	1.5	−40°C to 85°C	MC68A44CL
	2.0	0°C to 70°C	MC68B44L
Cerdip	1.0	0°C to 70°C	MC6844S
S Suffix	1.0	−40°C to 85°C	MC6844CS
	1.5	0°C to 70°C	MC68A44S
	1.5	−40°C to 85°C	MC68A44CS
	2.0	0°C to 70°C	MC68B44S
Plastic	1.0	0°C to 70°C	MC6844P
P Suffix	1.0	−40°C to 85°C	MC6844CP
	1.5	0°C to 70°C	MC68A44P
	1.5	−40°C to 85°C	MC68A44CP
	2.0	0°C to 70°C	MC68B44P

MOTOROLA

MC6845

MOS
(N-CHANNEL, SILICON-GATE)

CRT CONTROLLER
(CRTC)



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

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 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
- Pin Compatible with the MC6835

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6845L
	1.0	−40°C to 85°C	MC6845CL
	1.5	0°C to 70°C	MC68A45L
	1.5	−40°C to 85°C	MC68A45CL
	2.0	0°C to 70°C	MC68B45L
Cerdip S Suffix	1.0	0°C to 70°C	MC6845S
	1.0	−40°C to 85°C	MC6845CS
	1.5	0°C to 70°C	MC68A45S
	1.5	−40°C to 85°C	MC68A45CS
	2.0	0°C to 70°C	MC68B45S
Plastic P Suffix	1.0	0°C to 70°C	MC6845P
	1.0	−40°C to 85°C	MC6845CP
	1.5	0°C to 70°C	MC68A45P
	1.5	−40°C to 85°C	MC68A45CP
	2.0	0°C to 70°C	MC68B45P

PIN ASSIGNMENT

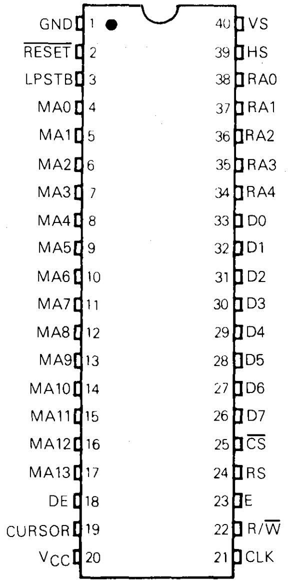
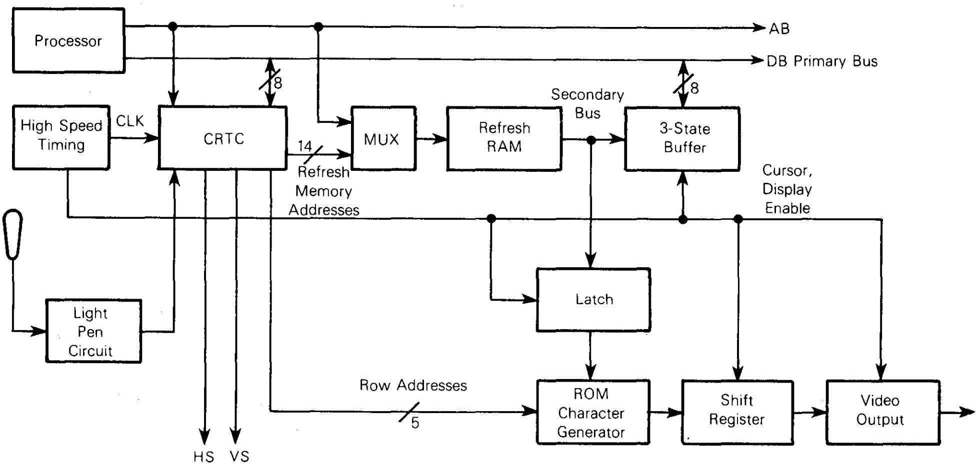


FIGURE 1 — TYPICAL CRT CONTROLLER APPLICATION



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	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 Cerdip Package Ceramic Package	θ_{JA}	 100 60 50	°C/W

The 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 $V_{out}) \leq V_{CC}$.

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:

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

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.

3

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ± 5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted, see Figures 2-4)

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
Hi-Z State Input Current (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)	D0-D7	V_{OH}	2.4	3.0	–	V
(I_{Load} = –100 μA)	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 = 0°C)		P_{INT}	–	600	750	mW
Input Capacitance	D0-D7	C_{in}	–	–	12.5	pF
	All Others		–	–	10	
Output Capacitance	All Outputs	C_{out}	–	–	10	pF

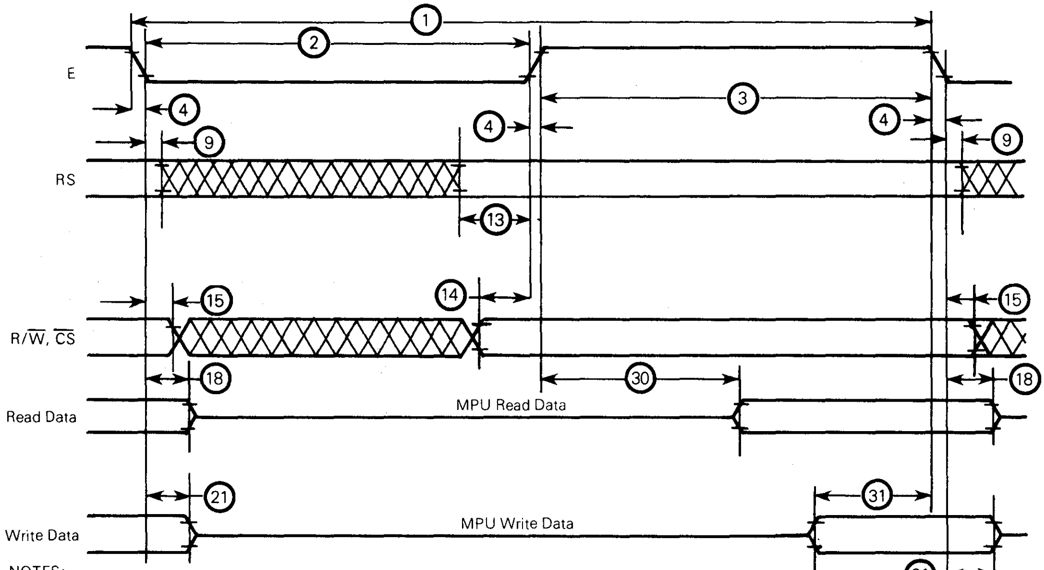
BUS TIMING CHARACTERISTICS (See Notes 1 and 2) (Reference Figures 2 and 3)

Ident. Number	Characteristic	Symbol	MC6845		MC68A45		MC68B45		Unit
			Min	Max	Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	1.0	–	0.67	–	0.5	–	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 (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} maximum (high impedance).

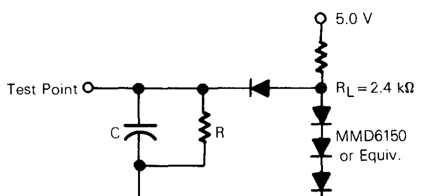
3

FIGURE 2 — 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 3 — BUS TIMING TEST LOAD



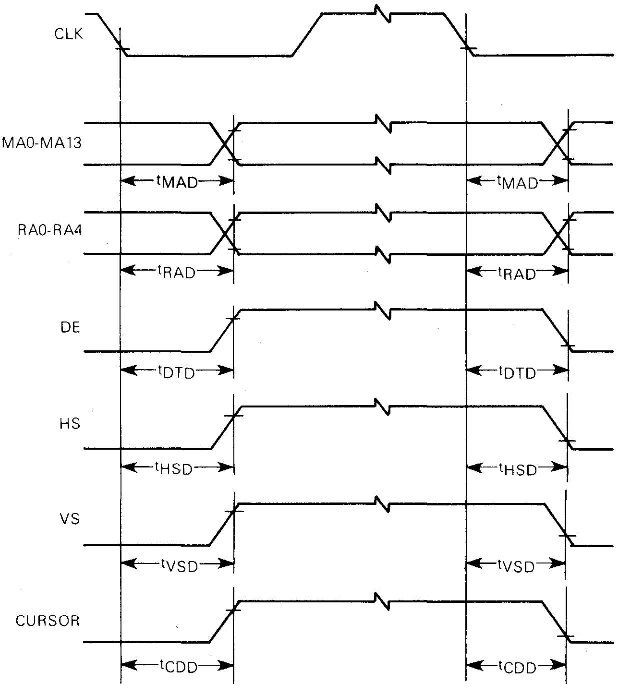
C = 130 pF for D0-D7
= 30 pF for MA0-MA13, RA0-RA4, DE, HS, VS, and CURSOR
R = 11 kΩ for D0-D7
= 24 kΩ for All Other Outputs

CRTC TIMING CHARACTERISTICS (Reference Figures 4 and 5)

Characteristic	Symbol	Min	Max	Unit
Minimum Clock Pulse Width, Low	PW_{CL}	150	–	ns
Minimum Clock Pulse Width, High	PW_{CH}	150	–	ns
Clock Frequency	f_c	–	3.0	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}	–	250	ns
Horizontal Sync Delay Time	t_{HSD}	–	250	ns
Vertical Sync Delay Time	t_{VSD}	–	250	ns
Cursor Display Timing Delay Time	t_{CDD}	–	250	ns
Light Pen Strobe Minimum Pulse Width	PW_{LPH}	80	–	ns
Light Pen Strobe Disable Time	t_{LPD1}	–	80	ns
	t_{LPD2}	–	10	ns

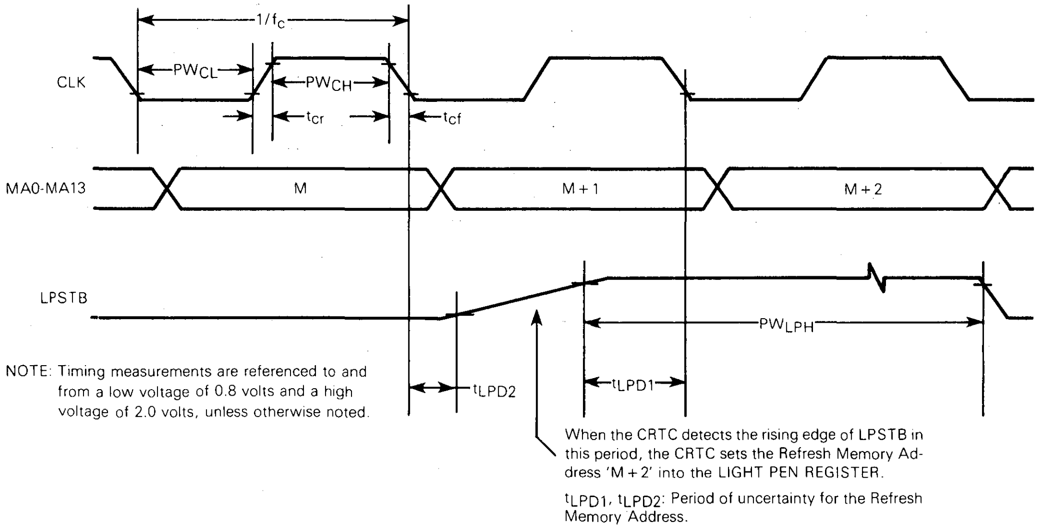
NOTE: The light pen strobe must fall to low level before VS pulse rises.

FIGURE 4 — 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 5 — CRTC-CLK, MA0-MA13, AND LPSTB TIMING DIAGRAM



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 6 and 7. Non-interlace 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 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 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 dots "x" dots (columns) wide and "y" dots (rows) high. Each character is created by selectively filling in

FIGURE 6 — RASTER SCAN SYSTEM (NON-INTERLACE)

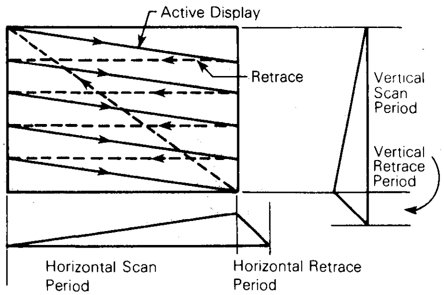
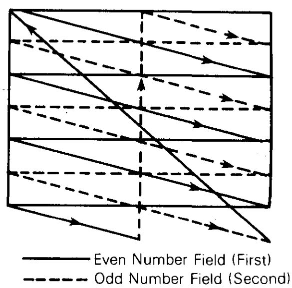


FIGURE 7 — RASTER SCAN SYSTEM (INTERLACE)



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.

Referring to Figure 1, 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. 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 9. The M6800 processor family works 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.

3

FIGURE 8 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL

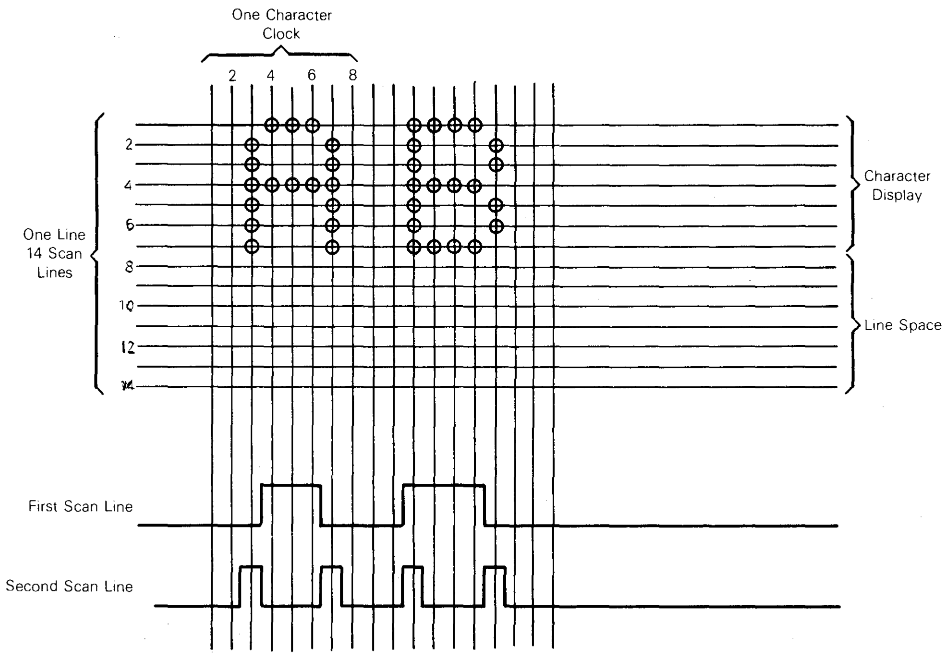
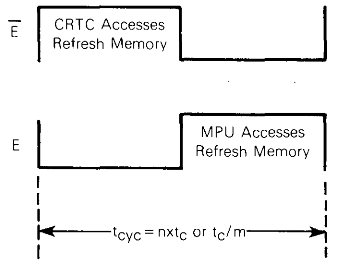


FIGURE 9 — TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING M6800 FAMILY MPU



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 $\overline{CS}$, RS, E, and R/$\overline{W}$ 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 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 ($\overline{CS}$) — The $\overline{CS}$ 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/$\overline{W}$) — The R/$\overline{W}$ 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} and 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.

NOTE

The horizontal sync output is not defined until after R2 is programmed.

(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.

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.

(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 and the CURSOR are not active until after the first frame has been displayed.

CRTC DESCRIPTION

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. A block diagram of the CRTC is shown in Figure 10.

All CRTC timing is derived from the 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 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 the 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, eight pages of 2K characters, etc. 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 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 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 scan line times with respect to the reference as shown in Figure 13.

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 10 — CRTC BLOCK DIAGRAM

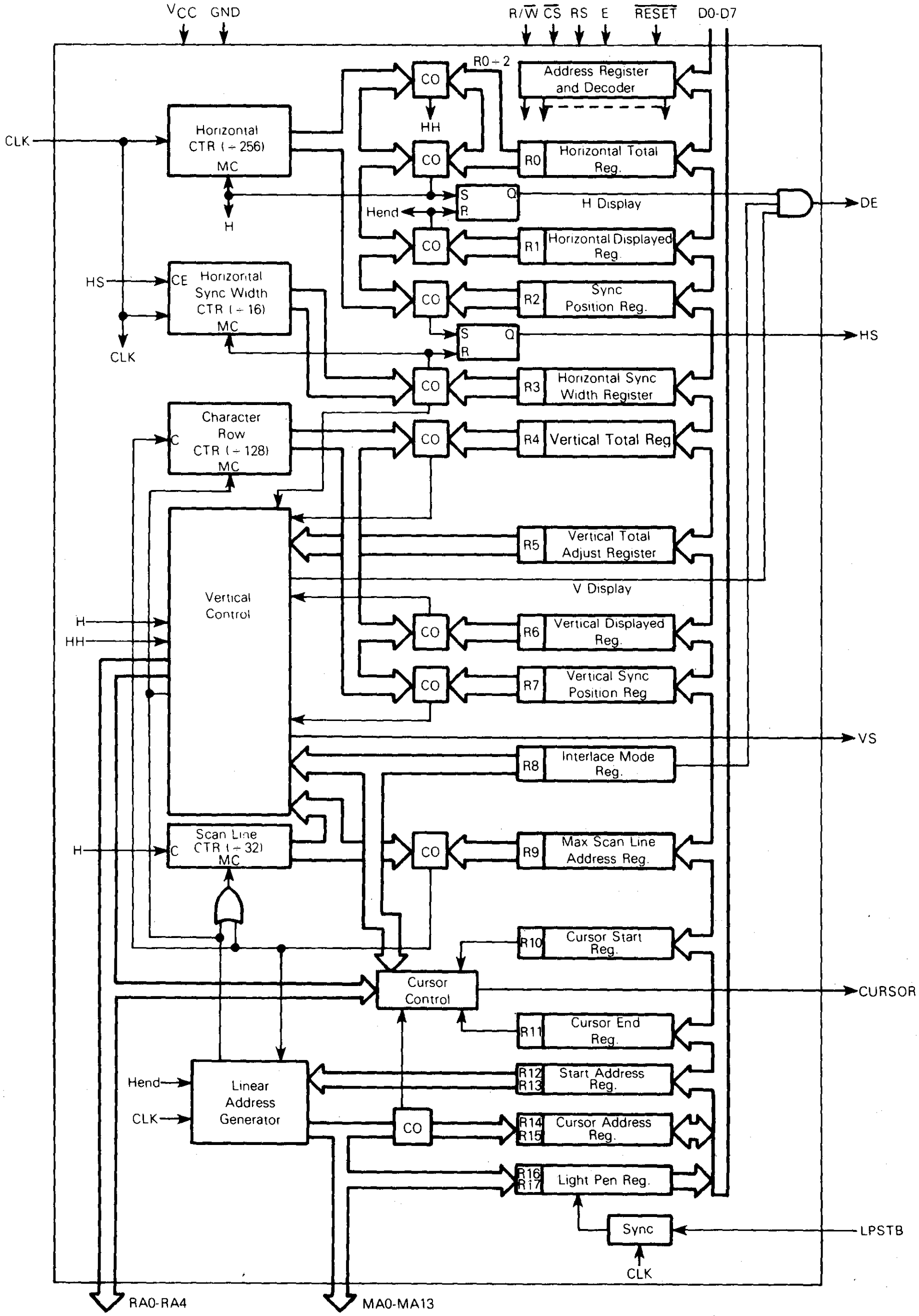
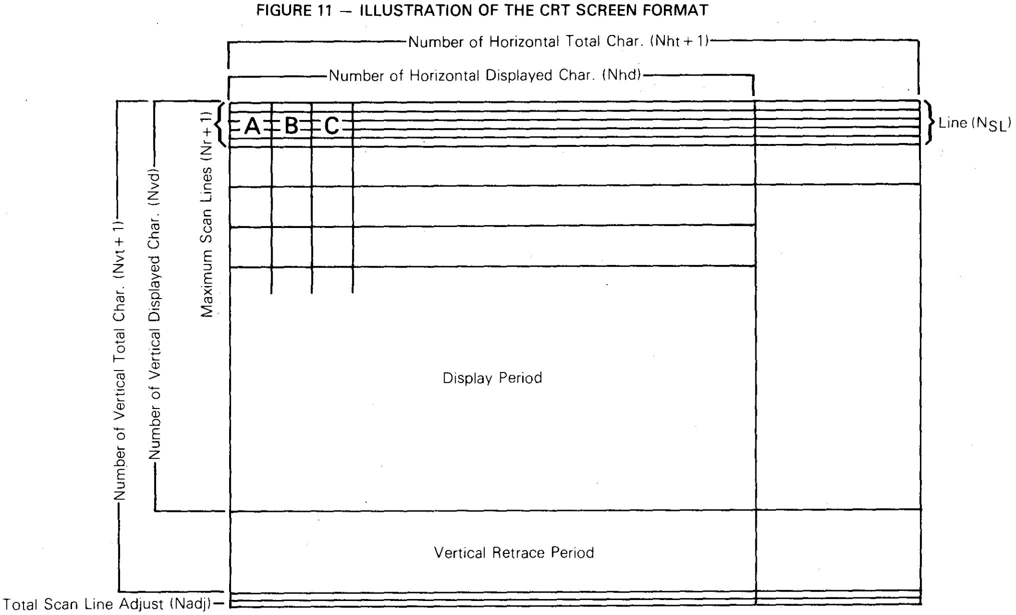


TABLE 2 — CRTC INTERNAL REGISTER ASSIGNMENT

$\overline{CS}$	RS	Address Register 4	3	2	1	0	Register #	Register File	Program Unit	Read	Write	Number of Bits 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					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							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)	—	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)	—	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 interlace is shown in Table 3.
2. Bit 5 of the cursor start raster register is used for blink period control, and bit 6 is used to select blink or no-blink.

FIGURE 11 — ILLUSTRATION OF THE CRT SCREEN FORMAT



NOTE 1: Timing values are described in Table 5.

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 R2 and R3 are less than the contents of R0. R2 must be greater than R1.

Sync Width Register (R3) — This 8-bit write-only register determines the width of the horizontal sync (HS) pulse. The vertical sync pulse width is fixed at 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 zero is written into 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 one third 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 row times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character row 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 the number of scan lines required.

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. 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) and greater than or equal to the vertical displayed (R6) 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. Table 3 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.

TABLE 3 — 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

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 one half 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.

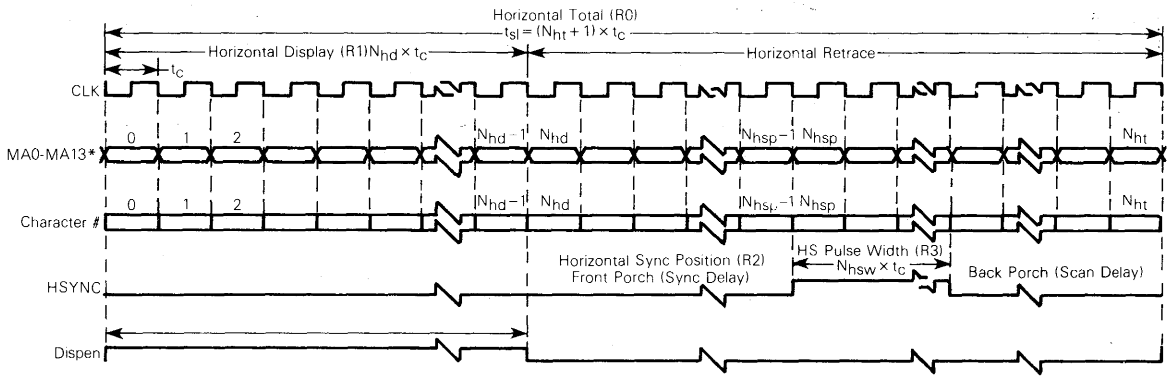
In the interlace sync and video mode, shown in Figure 14c, alternating lines of the character are displayed in the even field and the odd field. This effectively doubles the given bandwidth of the CRT monitor.

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. The horizontal total register value, R0, must be odd (i.e., an even number of character times).
2. For interlace sync and video mode only, the maximum scan-line address, R9, must be odd (i.e., an even number of scan lines).
3. For interlace sync and video mode only, the number (Nvd) programmed into the vertical display register (R6) must be one half 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.
4. 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. A full block cursor will be displayed in both the even and the odd field when the cursor end register (R11) is programmed to a value greater than the value in the maximum scan line address register (R9).

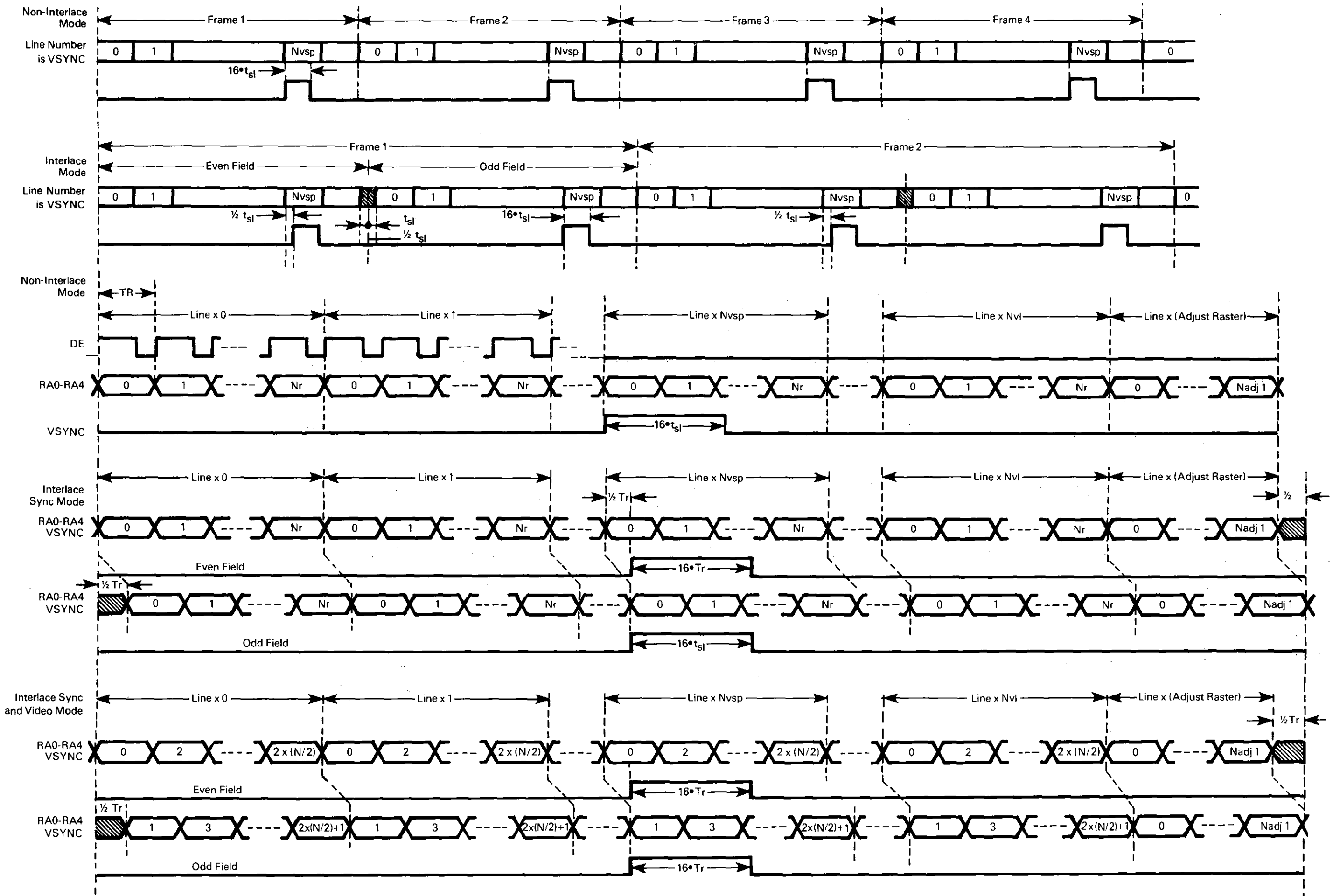
FIGURE 12 — CRTC HORIZONTAL TIMING



*Timing is shown for first displayed scan row only. See chart in Figure 15 for other rows. The initial MA is determined by the contents of start address register, R12/R13. Timing is shown for R12/R13=0.

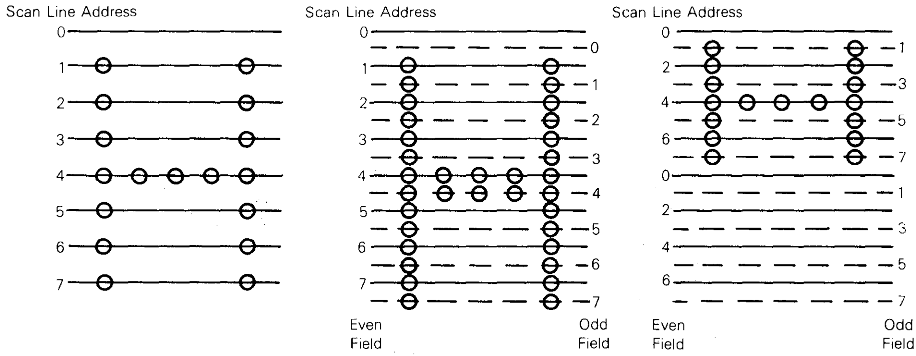
NOTE: Timing values are described in Table 5.

FIGURE 13 — CRTC VERTICAL TIMING



NOTES: 1. In interlace sync and video mode, maximum raster address (Nr) shall be odd.
2. In interlace mode, Nht shall be odd.

FIGURE 14 — 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 15. 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 4. 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.

TABLE 4 — 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

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 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 5) in this circuit, the value of R16 and R17 will need to be corrected in software. Figure 16 shows an interrupt driven approach although a polling routine could be used.

3

FIGURE 15 — CURSOR CONTROL

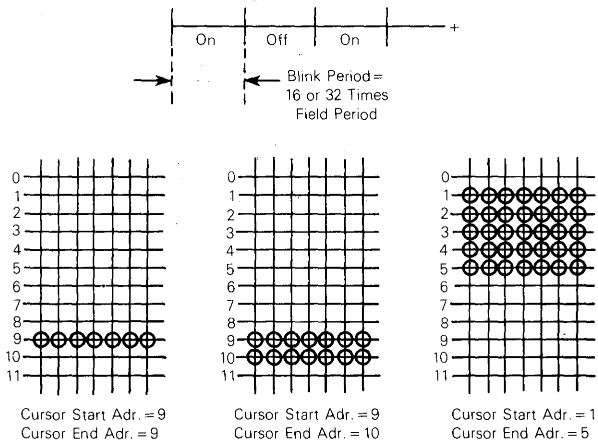
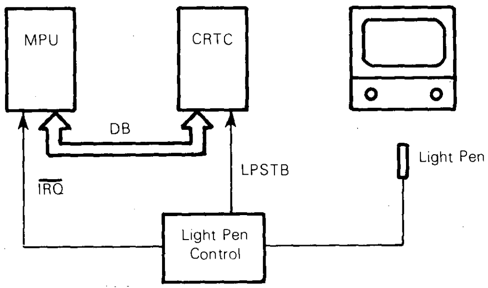


FIGURE 16 — INTERFACING OF LIGHT PEN



OPERATION OF THE CRTC

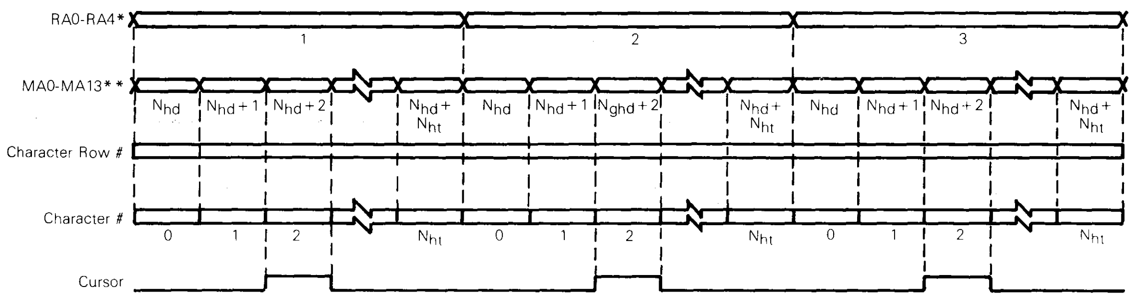
TIMING CHART OF THE CRT INTERFACE SIGNALS

Timing charts of CRT interface signals are illustrated in this section. When values listed in Table 5 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 is shown in Figure 11 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 zero.

TABLE 5 — 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}

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 5.

FIGURE 18 — REFRESH MEMORY ADDRESSING (MA0-MA13) STAGE CHART

	Character Row	Scan Line	Horizontal Display: Character				Horizontal Retrace (Non-Display)		
Vertical Display	0	0	0	1	→	$N_{hd}-1$	N_{hd}	→	N_{ht}
		N_{sl}	0		→	$N_{hd}-1$	N_{hd}	→	N_{ht}
	1	0	N_{hd}	$N_{hd}+1$	→	$2XN_{hd}-1$	$2XN_{hd}$	→	$N_{hd}+N_{ht}$
		N_{sl}	N_{hd}	$N_{hd}+1$	→	$2XN_{hd}-1$	$2XN_{hd}$	→	$N_{hd}+N_{ht}$
	2	0	$2XN_{hd}$	$2XN_{hd}+1$	→	$3XN_{hd}-1$	$3XN_{hd}$	→	$2N_{hd}+N_{ht}$
		N_{sl}	$2XN_{hd}$	$2XN_{hd}+1$	→	$3XN_{hd}-1$	$3XN_{hd}$	→	$2N_{hd}+N_{ht}$
			↓	↓		↓	↓		↓
	$N_{vd}-1$	0	$(N_{vd}-1)\times N_{hd}$	$(N_{vd}-1)\times N_{hd}+1$	→	$N_{vd}\times N_{hd}+1$	$N_{vd}\times N_{hd}$	→	$(N_{vd}-1)\times N_{hd}+N_{ht}$
		N_{sl}	$(N_{vd}-1)\times N_{hd}$	$(N_{vd}-1)\times N_{hd}+1$	→	$N_{vd}\times N_{hd}-1$	$N_{vd}\times N_{hd}$	→	$(N_{vd}-1)\times N_{hd}+N_{ht}$
Vertical Retrace (Non-Display)	N_{vd}	0	$N_{vd}\times N_{hd}$	$N_{vd}\times N_{hd}+1$	→	$(N_{vd}+1)\times N_{hd}-1$	$(N_{vd}+1)\times N_{hd}$	→	$N_{vd}\times N_{hd}+N_{ht}$
		N_{sl}	$N_{vd}\times N_{hd}$	$N_{vd}\times N_{hd}+1$	→	$(N_{vd}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$	→	$N_{vd}+N_{hd}+N_{ht}$
	N_{vt}	0	$N_{vt}\times N_{hd}$	$N_{vt}\times N_{hd}+1$		$(N_{vt}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$		$N_{vt}\times N_{hd}+N_{ht}$
		N_{sl}	$N_{vt}\times N_{hd}$			$(N_{vt}+1)\times N_{hd}-1$	$(N_{vt}+1)\times N_{hd}$		$N_{vt}\times N_{hd}+N_{ht}$
	$N_{vt}+1$	0	$(N_{vt}+1)\times N_{hd}$	$(N_{vt}+1)\times N_{hd}+1$		$(N_{vt}+2)\times N_{hd}-1$	$(N_{vt}+2)\times N_{hd}$		$(N_v+1)N_{hd}+N_{ht}$
		$N_{adj}-1$	$(N_{vt}+1)\times N_{hd}$	$(N_{vt}+1)\times N_d+1$		$(N_{vt}+2)\times N_{hd}-1$	$(N_{vt}+2)\times N_{hd}$		$(N_{vt}+1)N_{hd}+N_{ht}$

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.

DETERMINING REGISTER CONTENTS

Some of the register contents are determined rather easily. They are:

Register	Name	Contents
R8	Interlace Mode Register	See Table 3
R10	Cursor Start	See Figure 15 and Table 4
R11	Cursor End	See Figure 15
R12	Start Address (H)	User programs first memory location to be displayed
R13	Start Address (L)	
R14	Cursor (H)	User programs desired cursor location
R15	Cursor (L)	
R16	Light Pen (H)	Can be loaded via light-pen strobe only
R17	Light Pen (L)	

The remaining register contents must be determined from some basic data related to the CRT monitor and from the user-desired display format. The CRTC reference sheet (see Figure 19) gives a set of formulas for calculating the register contents as well as other useful characteristics of the display. This type of data is summarized under basic parameters in Figures 20 and 21; most or all of this data must be supplied by the user before he can determine the contents for registers R0-R7 and R9. All variables B_1-B_{10} are equal to basic parameters 1 through 10.

FIGURE 19 — CRTC REFERENCE SHEET

Register Function

R0	Horizontal Total
R1	Horizontal Displayed
R2	Horizontal Sync Position
R3	Horizontal Sync Width
R4	Vertical Total
R5	Vertical Total Adjust
R6	Vertical Displayed
R7	Vertical Sync Position
R8	Interlace Mode
R9	Maximum Scan Line Address
R10	Cursor Start
R11	Cursor End
R12	Start Address (H)
R13	Start Address (L)
R14	Cursor (H)
R15	Cursor (L)
R16	Light Pen (H)
R17	Light Pen (L)

Intermediate Calculations

Symbol	Description	Calculation
f'	Dot frequency (1st approx.)	$\frac{B_5 \bullet (B_7+B_9)}{(1/B_1)-B_3}$
t_c	Character Time	$\frac{1}{[(R0)+1]\bullet B_1}$
f	Dot frequency	$\frac{B_7+B_9}{t_c}$
t_{sl}	Scan line time	$[(R0)+1]\bullet t_c$
n	Total # of scan lines	$\frac{1}{B_2 \bullet t_{sl}}$
N	Integer and	$\frac{n}{B_8+B_{10}} = N + \frac{R}{B_8+B_{10}}$
R	Integer remainder	
t_{cr}	Character row time	$(B_8+B_{10})\bullet t_{sl}$
t_{hr}	Horizontal retrace time	$\le \frac{[(R0)+1-B_5]\bullet(B_7+B_9)}{f}$
t_{vr}	Vertical retrace time	$\le \frac{B1}{B2} - B_6(B_8+B_{10})\bullet t_{sl}$

Register Calculations

Register	Calculation
R0	$\frac{f'}{B_1\bullet(B_7+B_9)} - 1$
R1	B_5
R2	$(R1) + \frac{(R3)}{2}$
R3	$\frac{(R0)-(R1)}{3}$
R4	$N-1$
R5	R
R6	B_6
R7	$[(R4)+1] - \frac{16-(R5)}{B_8+B_{10}} \ge (R7) \ge (R6)$
R9	$(B_8+B_{10})-1$

In Figures 20 and 21, worksheet example calculations are shown for 32×16 and 80×24 display formats respectively. The following items are keyed to the figures. Basic parameters (1) through (10) have been provided; items (1) through (4) are data about the CRT monitor and items (5) through (10) are data about the user's desired display.

(1) Calculate the approximate dot frequency. The user should verify that the bandwidth of his CRT monitor will accomodate this frequency.
(2) Calculate R0. The resultant answer will usually be an integer plus a fraction. Assume the next high integer.
(3) Fill in value for R1.
(4) Calculate R3. Use the next highest integer. In these examples the sync width was chosen to be one third of the horizontal blanking interval.
(5) Calculate R2. Again, use the next highest integer.
(6) Calculate t_C, character tie. This is the time required for one scan line of one character block to be written.
(7) Calculate the exact dot frequency.
(8) Calculate t_{sl}, scan line time. This is the time required for one scan line of one character row to be written including retrace time.
(9) Calculate n. This is the total number of scan lines for each frame. Discard any fraction.
(10) Calculate N and R.
(11) Calculate R4.
(12) Fill in R5.
(13) Fill in R6.
(14) Calculate R7. If there is no possible value for R7, then the display demands for the CRT monitor exceed its capability. A compromise adjustment must be made in basic parameter 6, 8, or 10.
(15) Calculate R9.
(16) Calculate t_{cr}. This the time required for one character row to be written.
(17) Calculate t_{hr}. $t_{hr} > B_3$.
(18) Calculate t_{vr}. $t_{vr} > B_4$.

In Figure 20, calculation (B) verifies that the vertical period is 16.667 milliseconds or 60 hertz. The expression used is:

$$t_{cr} \times [(R4)+1] + [t_{sl} \times (R5)] = V_P.$$

Another check is calculation of horizontal sync pulse width R3. $t_C = PW_{HS}$ (typically approximately equals 4 microseconds).

For convenience, a blank worksheet is provided in Figure 22.

FIGURE 20 — CRTC WORKSHEET EXAMPLE CALCULATION (32×16)

Basic Parameters (B1-B10)		Symbol	Intermediate Calculations	Value
1. Horizontal frequency	$= 15750 \pm 500$	(1) f′	$\dfrac{32 \times (5+2)}{\frac{1}{15750} - 11 \times 10^{-6}}$	$= 4.27 \times 10^6$
2. Vertical frequency	= 60	(6) t_C	$\dfrac{1}{39 \times 15750}$	$= 1.63 \times 10^{-6}$
3. Minimum Horizontal retrace time	$= 11 \times 10^{-6}$	(7) f	$\dfrac{5+2}{1.63 \times 10^{-6}}$	$= 4.29 \times 10^6$
4. Minimum vertical retrace time	$= 10^{-3}$	(8) t_{sl}	$39 \times 1.63 \times 10^{-6}$	$= 63.6 \times 10^{-6}$
5. # of displayed characters per row	= 32	(9) n	$\dfrac{1}{60 \times 63.6 \times 10^{-6}}$	= 262
6. # of displayed charactor rows	= 16	(10) N		17
7. # of dots in character dot matrix row	= 5	R	$\dfrac{262}{7+8}$	= 7
8. # of scan lines in character • matrix column	= 7	(16) t_{cr}	$(7+8) \times 63.6 \times 10^{-6}$	$= 954 \times 10^{-6}$
9. Number of dots between horizontal adjacents	= 2	(17) t_{hr}	$\dfrac{(38+1-32) \bullet (5+2)}{4.29 \times 10^6}$	$= 11.42 \times 10^{-6}$
10. Number of scan lines between vertical adjacents	= 8	(18) t_{vr}	$[\frac{15750}{60} - 16(7+8)] \times 63.6 \times 10^{-6}$	$= 1.431 \times 10^{-3}$

Register Calculations

Register		Decimal	Hex
(2) R0	$\dfrac{4.27 \times 10^6}{15{,}750 \times (5+2)} - 1 =$	38	26
(3) R1	$B_5 = 32$	32	20
(5) R2	$32 + \frac{3}{2}$	33	21
(4) R3	$\frac{38-32}{3} =$	2	2
(11) R4	$17 - 1 =$	16	10
(12) R5	R = 7	7	7
(13) R6	$B_6 = 16$	16	10
(14) R7	(A)	16	10
R8			
(15) R9	$7+8-1=$	14	0E
R10			
R11			
R12			
R13			
R14			
R15			

(A) $16 + 1 - \dfrac{16-7}{7+8} \geq (R7) \geq 16$

$17 - .6 \geq (R7) \geq 16$

(B) $954 \times 10^{-6} \times 17 = 16.218 \times 10^{-3}$

$+ \; 63.6 \times 10^{-6} \times 7 = .445 \times 10^{-3}$

$\overline{16.663 \text{ ms}} = \dfrac{1}{f}$

$f \approx 60$ Hz

FIGURE 21 — CRTC WORKSHEET EXAMPLE CALCULATION (80 × 24)

Basic Parameters (B1-B10)		Symbol	Intermediate Calculations	Value
1. Horizontal frequency	= 18,600	(1) f'	$\dfrac{80\bullet(7+2)}{\frac{1}{18600} - 11\times10^{-6}}$	$16\bullet836\times10^{6}$
2. Vertical frequency	= 60	(6) t_c	$\dfrac{1}{(100+1)\bullet18600}$	532.31×10^{-9}
3. Minimum Horizontal retrace time	= 11×10^{-6}	(7) f	$\dfrac{7+2}{532.31\times10^{-9}}$	16.907×10^{6}
4. Minimum vertical retrace time	= 1×10^{-3}	(8) t_{sl}	$(100+1)(532.31\times10^{-9})$	53.76×10^{-6}
5. # of displayed characters per row	= 80	(9) n	$\dfrac{1}{(60)(53.76\times10^{-6})}$	310
6. # of displayed charactor rows	= 24	(10) N		28
7. # of dots in character dot matrix row	= 7	R	$\dfrac{310}{11}$	2
8. # of scan lines in charactor • matrix column	= 9	(16) t_{cr}	$(9+2)(53.76\times10^{-6})$	591.39×10^{-6}
9. Number of dots between horizontal adjacents	= 2	(17) t_{hr}	$\leq \dfrac{(101-80)(7+2)}{16.907\times10^{6}}$	11.17×10^{-6}
10. Number of scan lines between vertical adjacents	= 2	(18) t_{vr}	$\leq \left[\dfrac{18600}{60} - 24(11)\right]53.76\times10^{-6}$	2.47×10^{-3}

Register Calculations

Register		Decimal	Hex
(2) R0	$\dfrac{16.836\times10^{6}}{(18,600)(9)} - 1$	100	64
(3) R1	B5 = 80	80	50
(5) R2	$80 + \dfrac{7}{2}$	84	54
(4) R3	$\dfrac{R0-R1}{3}$	7	07
(11) R4	28 − 1	27	1B
(12) R5	R = 2	02	02
(13) R6	B6 = 24	24	18
(14) R7	(A)	25	19
R8		0	0
(15) R9	(9 + 2) − 1	10	0A
R10		00	00
R11		11	0B
R12			00
R13		128	80
R14		128	00
R15			80

(A) $(27+1) - \dfrac{(16-2)}{11} \geq R7 \geq 24$

$26.72 \geq R7 \geq 24$

(B) $B2 = 1/[(t_{cr})(R4+1) + (t_{sl})(R5)]$

$= 1/[(591.39\times10^{6})(28) + (53.76\times10^{-6})(2)]$

$= 1/16.667\times10^{-3}$

$= 60$

FIGURE 22 — CRTC WORKSHEET

Basic Parameters			Symbol	Intermediate Calculations	Value
1. Horizontal frequency	=		f′		
2. Vertical frequency	=		t_c		
3. Minimum Horizontal retrace time	=		f		
4. Minimum vertical retrace time	=		t_{sl}		
5. # of displayed characters per row	=		n		
6. # of displayed charactor rows	=		N		
7. # of dots in character dot matrix row	=		R		
8. # of scan lines in character • matrix column	=		t_{cr}		
9. Number of dots between horizontal adjacents	=		t_{hr}		
10. Number of scan lines between vertical adjacents	=		t_{vr}		

Register Calculations

Register	Decimal	Hex
R0		
R1		
R2		
R3		
R4		
R5		
R6		
R7		
R8		
R9		
R10		
R11		
R12		
R13		
R14		
R15		
R16		
R17		
R18		
R19		

CRTC INITIALIZATION

Register 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 program required to initialize the CRTC for a 80 × 24 format (example calculation #2) is shown in Figure 23.

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.

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.

FIGURE 23 — MC6800 PROGRAM FOR CRTC INITIALIZATION

```
PAGE 001   CRTCINIT.SA:0    MC6845 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 CB 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                  * 80 x 24 non-interlaced format
00026A 1020                    ORG    $1020      start of table
00027A 1020     65   A CRTTAB  FCB    $64,$50    R0, R1 - H total & H displayed
     A 1021     50   A
00028A 1022     56   A         FCB    $54,$07    R2, R3 - HS pos. & HS width
     A 1023     09   A
00029A 1024     18   A         FCB    $1B,$02    R4, R5 - V total & V total adj.
     A 1025     0A   A
00030A 1026     18   A         FCB    $18,$19    R6, R7 - V displayed $ VS pos.
     A 1027     18   A
00031A 1028     00   A         FCB    $00,$0A    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
```

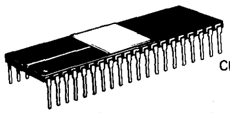
MOTOROLA

MC6846

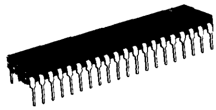
MOS

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

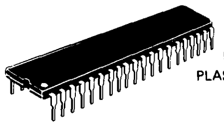
ROM—I/O—TIMER



L SUFFIX
CERAMIC PACKAGE
CASE 715



S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711

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.

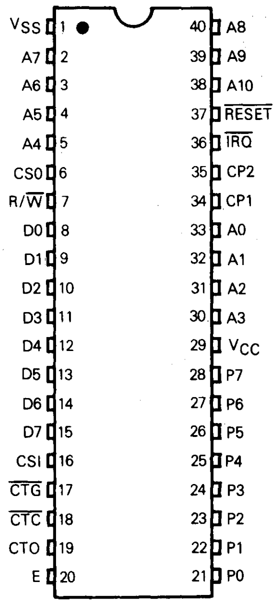
This device is capable of interfacing with the MC6802 (basic MC6800, clock, and 128 bytes of RAM) as well as the entire M6800 family if desired. No external logic is required to interface with most peripheral devices.

- 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

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic	1.0	0°C to 70°C	MC6846L
L Suffix	1.0	−40°C to 85°C	MC6846CL
	1.5	0°C to 70°C	MC68A46L
Cerdip	1.0	0°C to 70°C	MC6846S
S Suffix	1.0	−40°C to 85°C	MC6846CS
	1.5	0°C to 70°C	MC68A46S
Plastic	1.0	0°C to 70°C	MC6846P
P Suffix	1.0	−40°C to 85°C	MC6846CP
	1.5	0°C to 70°C	MC68A46P

PIN ASSIGNMENT



MAXIMUM RATINGS

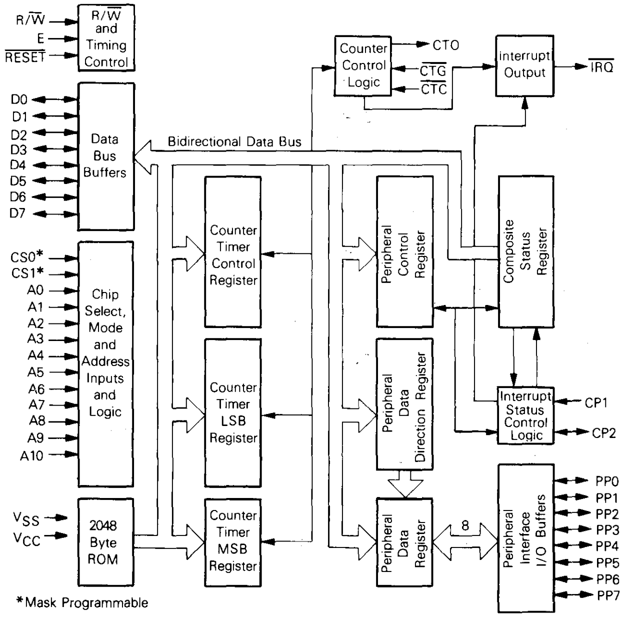
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6846, MC68A46 MC6846C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

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

MC6846 BLOCK DIAGRAM



*Mask Programmable

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 ≡ 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:

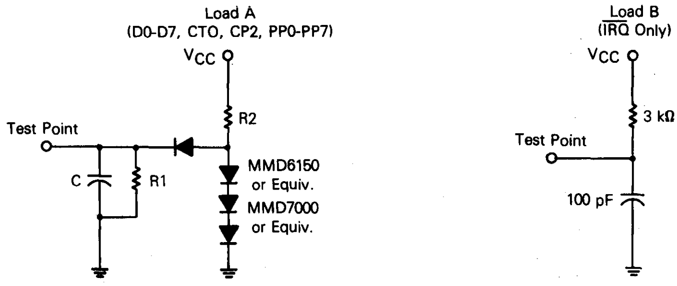
$$P_D = K \div (T_J + 273°C) \tag{2}$$

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.

FIGURE 1 — BUS TIMING TEST LOADS



C = 130 pF for D0-D7
= 30 pF for CTO, CP2, PP0-PP7
R1 = 11.7 kΩ for D0-D7, CTO, CP2, PP0-PP7
R2 = 2.5 kΩ for D0-D7
= 1.2 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)	$R/\overline{W}$, $\overline{RESET}$, CS0, CS1 CP1, $\overline{CTG}$, $\overline{CTC}$, E, A0-A10	I_{in}	–	1.0	2.5	μA
Hi-Z (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)	 D0-D7 Other Outputs	I_{OH}	 −205 −200	 – –	 – –	 μA μA
(V_O = 1.5 V, the current for driving other than TTL, e.g., Darlington Base)	CP2, PP0-PP7		−1.0	–	−10	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)	$\overline{IRQ}$	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, $R/\overline{W}$, $\overline{RESET}$, CS0, CS1, CP1, $\overline{CTC}$, $\overline{CTG}$ $\overline{IRQ}$	C_{in}	 – – – –	 – – – –	 20 12.5 10 7.5	pF
	PP0-PP7, C2, CTO	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

Characteristic	Symbol	Min	Max	Unit
Input Rise and Fall Time $\overline{CTC}$ and $\overline{CTG}$	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 Min	MC6846 Max	MC68A46 Min	MC68A46 Max	Unit
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	50*	20	50*	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.

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high impedance).

FIGURE 2 — BUS TIMING

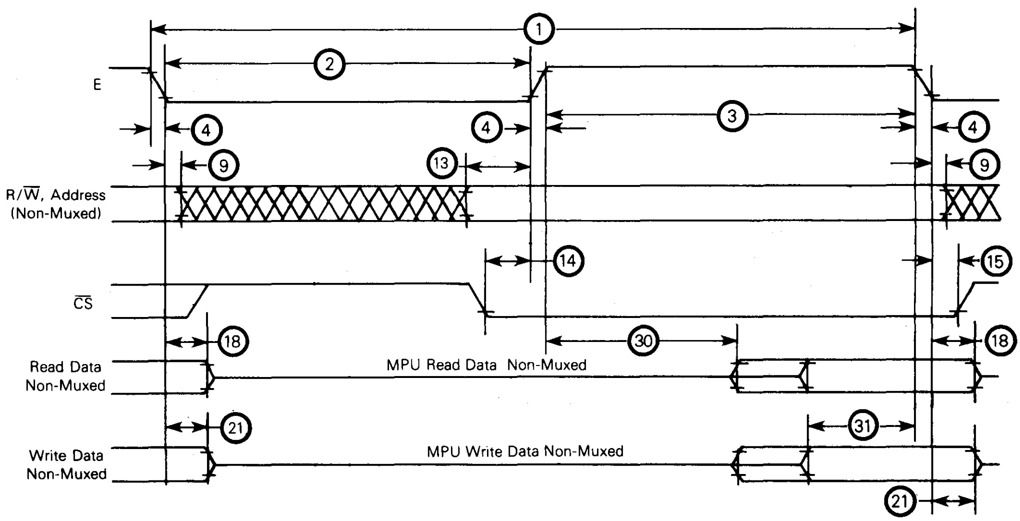


FIGURE 3 — PERIPHERAL PORT LATCH SETUP AND HOLD TIME

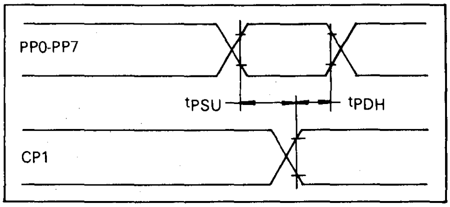
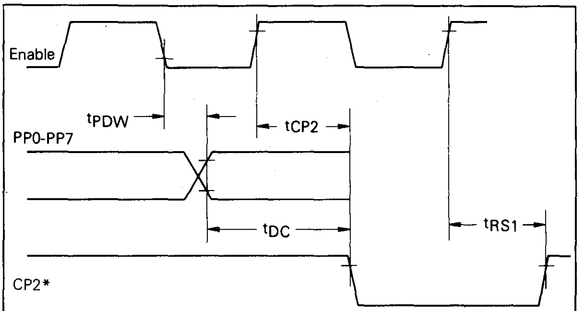


FIGURE 4 — PERIPHERAL DATA AND CP2 DELAY (Control Mode PCR5 = 1, PCR4 = 0, PCR3 = 1)



*CP2 goes low as the result of positive transition of the second E pulse.

3

FIGURE 5 — $\overline{IRQ}$ RELEASE TIME

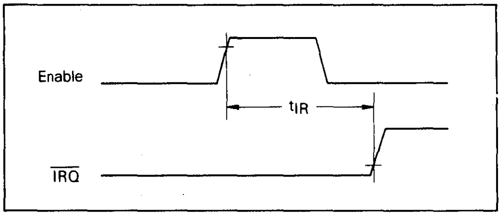


FIGURE 6 — PERIPHERAL PORT SETUP TIME

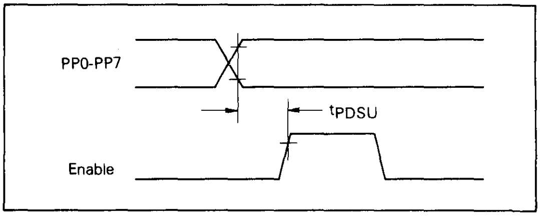


FIGURE 7 — CP2 DELAY TIME (PCR5 = 1, PCR4 = 0, PCR3 = 0)

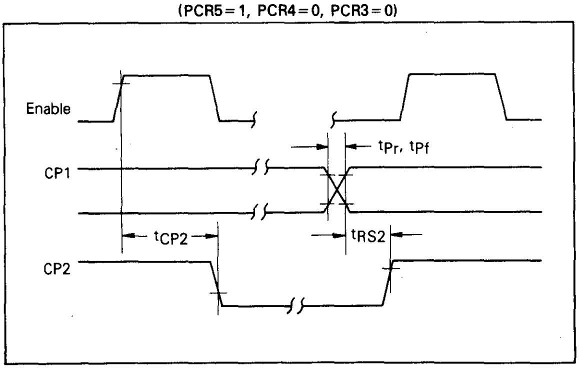


FIGURE 8 — INPUT PULSE WIDTHS

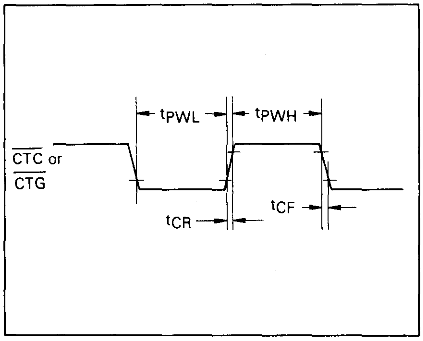
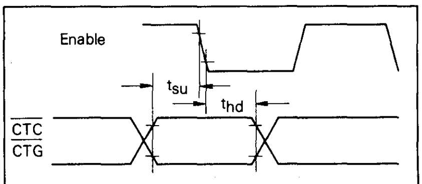
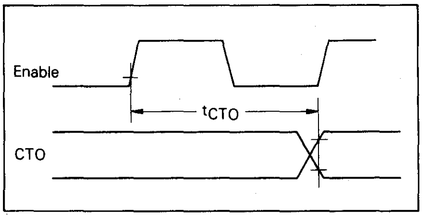


FIGURE 9 — INPUT SETUP AND HOLD TIMES*



*This mode is valid only for synchronous operation.

FIGURE 10 — 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 11 — TYPICAL MICROCOMPUTER

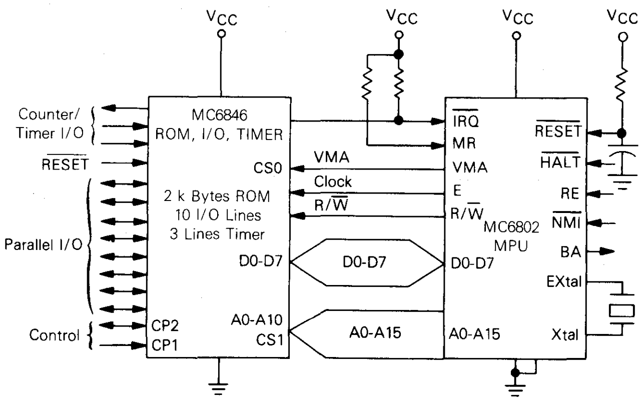


Figure 11 is a block diagram of a typical cost-effective microcomputer. The MPU is the center of the microcomputer system and is shown in a minimum system interfacing with a ROM combination chip. It is not intended that this system be limited to this function but that it be expandable with other parts 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 $\overline{CTC}$ (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 ($\overline{CTG}$) 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.

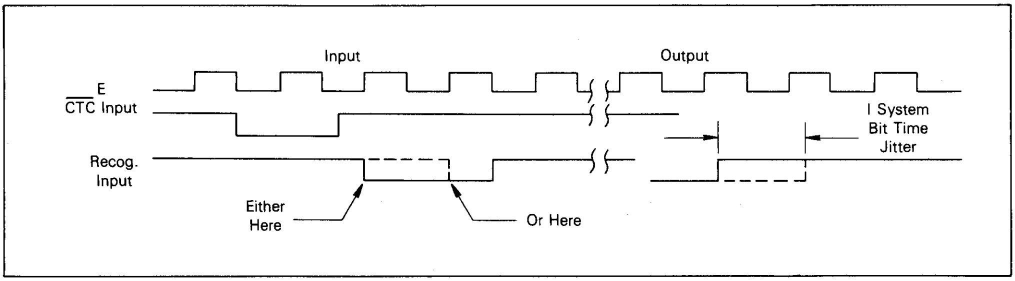
When observing recurring events, a lack of synchronization will result in either "System jitter" or "Input jitter" being observed on the output of the MC6846 when using an asynchronous clock and gate input signal. "System jitter" is the result of the input signals being out of synchronization with E 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 12.

GATE INPUTS ($\overline{CTG}$)

The input pin $\overline{CTG}$ 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, $\overline{CTG}$ transition is recognized on the fourth Enable pulse (provided setup and hold time requirements are met), and the high or low levels of the $\overline{CTG}$ input must be stable for at least one system clock period plus the sum of setup and hold times. All references to $\overline{CTG}$ transition in this document relate to internal recognition of the input transition.

The $\overline{CTG}$ input of the timer directly affects the internal 16-bit counter. The operation of $\overline{CTG}$ is therefore independent of the divide-by-8 prescaler selection.

FIGURE 12 — RECOGNITION OF $\overline{CTC}$



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 13 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.

TABLE 1 — INTERNAL REGISTER ADDRESSES

Register Selected	A2	A1	A0
Composite Status Register	X	0	0
Peripheral Control Register	0	0	1
Data Direction Register	0	1	0
Peripheral Data Register	0	1	1
Timer Control Register	1	0	1
Timer MSB Register	1	1	0
Timer LSB Register	1	1	1
ROM Address	X	X	X

INITIALIZATION

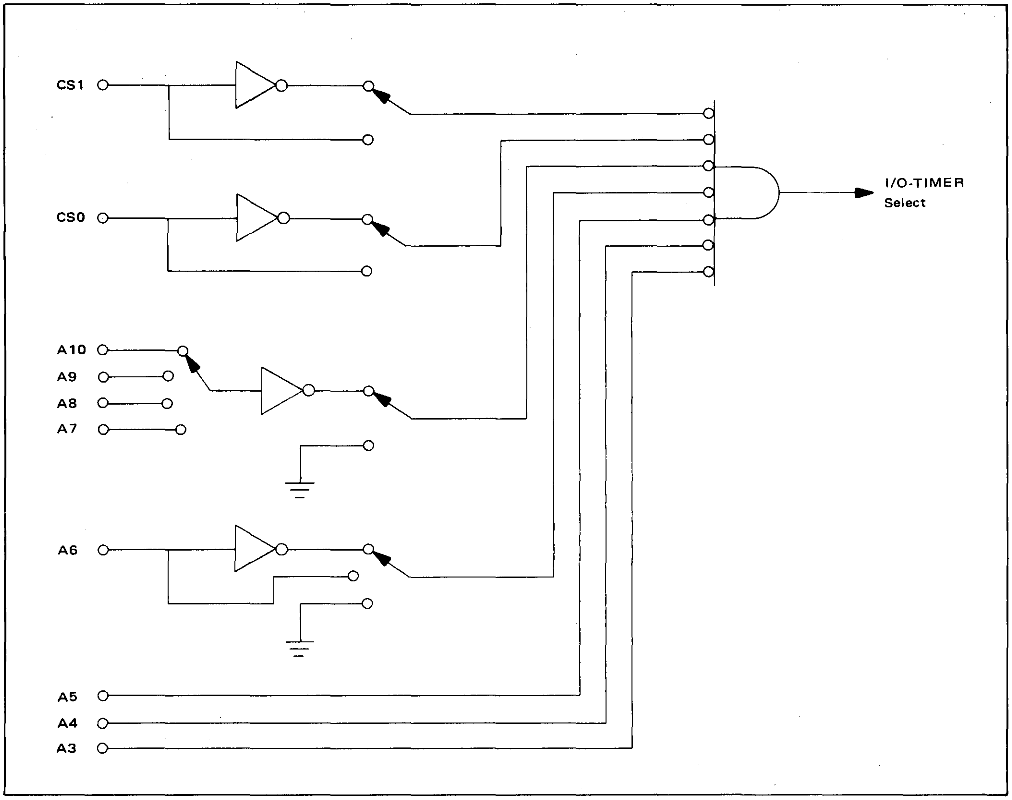
When the $\overline{RESET}$ input has accepted a low signal, all registers are initialized to the reset state. The data direction and peripheral data registers are cleared. The Peripheral Control Register is cleared except for bit 7 (the $\overline{RESET}$ bit). This forces the parallel port to the input 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.

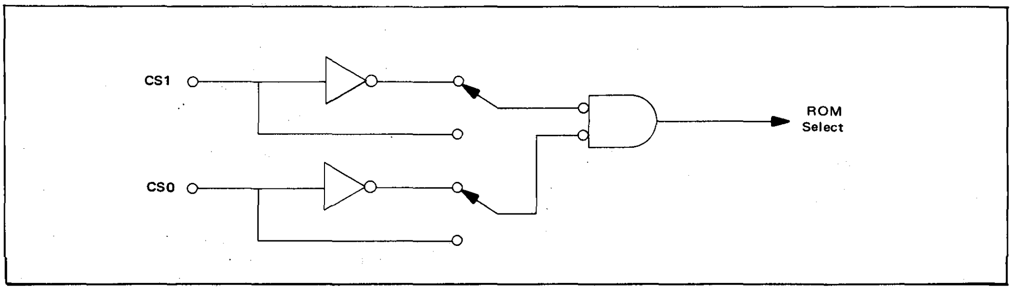
FIGURE 13 — I/O-TIMER SELECT CIRCUITRY



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 14.

FIGURE 14 — 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 ($\overline{CTC}$)
	1		Timer uses System Clock (E)
TCR2	0	÷ 8 Prescaler Enabler	Clock is not Prescaled
	1		Clock is prescaled by ÷ 8 Counter
TCR3 TCR4 TCR5	X X X	Operating Mode Selection	See Table 3
TCR6	0	Timer Interrupt Enable	$\overline{IRQ}$ Masked from Timer
	1		$\overline{IRQ}$ 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 $\overline{RESET}$ = "0") condition or internal recognition of a negative transition of the $\overline{CTG}$ 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	$\overline{CTG}\downarrow + W + R$	T.O.
0	0	1	Cascaded Single Shot	$\overline{CTG}\downarrow + R$	T.O.
0	1	0	Continuous	$\overline{CTG}\downarrow + R$	T.O.
0	1	1	Normal Single Shot	$\overline{CTG}\downarrow + R$	T.O.
1	0	0	Frequency Comparison	$\overline{CTG}\downarrow \cdot \overline{I} \cdot (W + T.O.) + R$	$\overline{CTG}\downarrow$ Before T.O.
1	0	1		$\overline{CTG}\downarrow \cdot \overline{I} + R$	T.O. Before $\overline{CTG}\downarrow$
1	1	0	Pulse Width Comparison	$\overline{CTG}\downarrow \cdot \overline{I} + R$	$\overline{CTG}\uparrow$ Before T.O.
1	1	1			T.O. Before $\overline{CTG}\uparrow$

R = Reset Condition
W = Write Timer Latches
T.O. = Counter Time Out

$\overline{CTG}\downarrow$ = Negative Transition of Pin 17
$\overline{CTG}\uparrow$ = Positive Transition of Pin 17
$\overline{I}$ = 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 Initialization	Timer Output (2X)
0	0	$\overline{CTG}\downarrow + \overline{W} + R$	←(N + 1) (T) → ←(N + 1) (T) → ←(N + 1) (T) → V_{OH} / V_{OL}; to, T.O., T.O., T.O.
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).

3

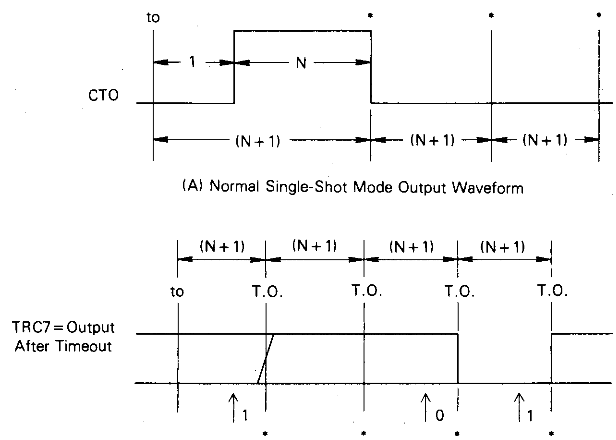
NORMAL SINGLE-SHOT TIMER MODE (TCR3=0, TCR4=1, TCR5=1)

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 15.

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 15 — 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 15.)

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	$\overline{G}\downarrow$ 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 $\overline{G}\downarrow$

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 16.) 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"

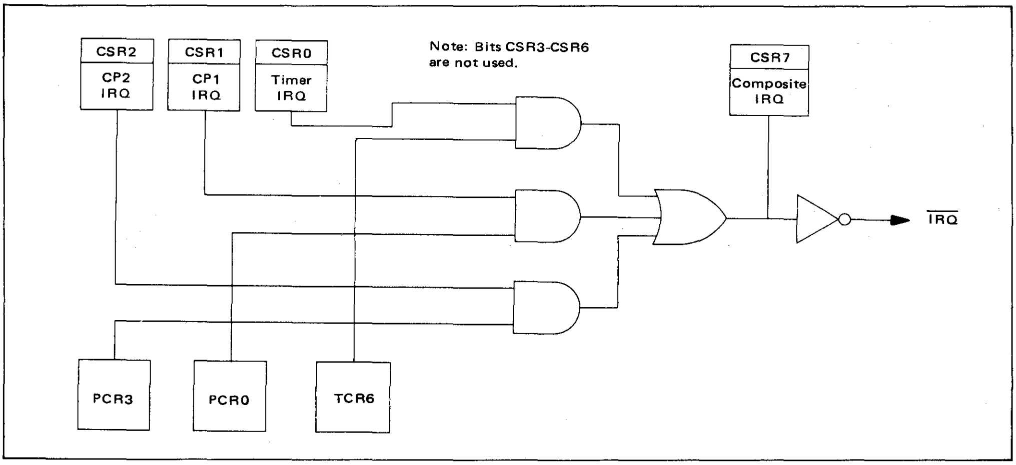
2) Any Counter Initialization condition.

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 16 — 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.

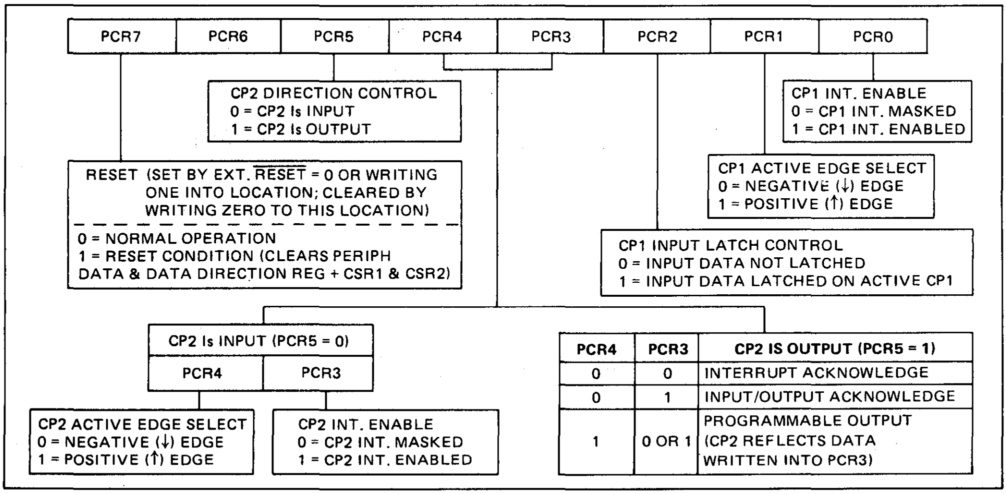
PERIPHERAL DATA REGISTER

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)



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 $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{RESET}}$ 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
X	0	0	Composite Status Register
0	0	1	Peripheral Control Register
0	1	0	Data Direction Register
0	1	1	Peripheral Data Register
1	0	1	Timer Control Register
1	1	0	Timer MSB Register
1	1	1	Timer LSB Register
X	X	X	ROM Address

3

TABLE 10 — COMPOSITE STATUS REGISTER

CSR7	CSR3-CSR6 NOT USED. DEFAULT TO ZERO WHEN READ	CSR2	CSR1	CSR0

CSR7:
COMPOSITE INTERRUPT FLAG
0 = NO ENABLED INTERRUPT FLAG SET
1 = ONE OR MORE ENABLED INTERRUPT FLAGS SET.*

INVERSE OF THIS BIT APPEARS AT $\overline{IRQ}$ OUTPUT

*STATUS OF THIS BIT CAN BE EXPRESSED AS:
CSR7 = CSR0 • TCR6 + CSR1 • PCR0 + CSR2 • PCR3

CSR2:
CP2 INTERRUPT FLAG
0 = NO INT REQ
1 = INT REQUESTED

CSR0:
TIMER INTERRUPT FLAG
0 = NO INT REQ.
1 = INT REQUESTED

CSR1:
CP1 INTERRUPT
0 = NO INT REQ.
1 = INT REQUESTED

TABLE 11 — TIMER CONTROL REGISTER

TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

TCR6:
INTERRUPT ENABLE
0 = $\overline{IRQ}$ MASKED
1 = $\overline{IRQ}$ ENABLED

TCR0:
INTERNAL RESET
0 = TIMER ENABLED
1 = RESET STATE

TCR7:
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

TCR1:
CLOCK SOURCE
0 = EXTERNAL CLOCK (CTC)
1 = INTERNAL CLOCK ($\phi 2$)

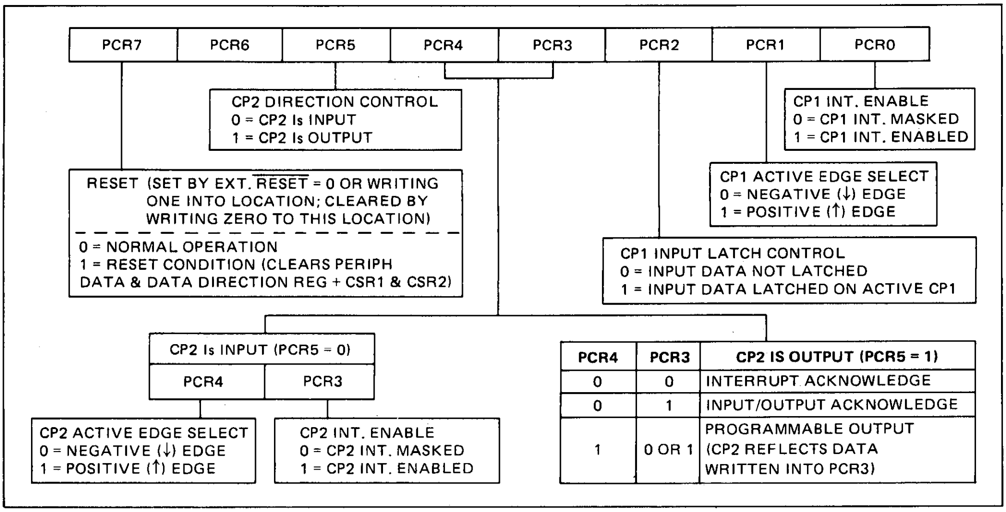
TCR2:
÷ 8 PRESCALE ENABLE
0 = CLOCK NOT PRESCALED
1 = CLOCK PRESCALED (÷ 8)

TCR3	TCR4	TCR5	TIMER OPERATING MODE	COUNTER INITIALIZATION	INTERRUPT FLAG SET
0	0	0	CONTINUOUS	$\overline{CTG}\downarrow$ + W + R	T.O.
0	0	1	CASCADED SINGLE SHOT	$\overline{CTG}\downarrow$ + R	T.O.
0	1	0	CONTINUOUS	$\overline{CTG}\downarrow$ + R	T.O.
0	1	1	NORMAL SINGLE SHOT	$\overline{CTG}\downarrow$ + R	T.O.
1	0	0	FREQUENCY COMPARISON	$\overline{CTG}\downarrow \cdot \overline{T} \cdot$ (W + T.O.) + R	$\overline{CTG}\downarrow$ BEFORE T.O.
1	0	1		$\overline{CTG}\downarrow \cdot \overline{T}$ + R	T.O. BEFORE $\overline{CTG}\downarrow$'
1	1	0	PULSE WIDTH COMPARISON	$\overline{CTG}\downarrow \cdot \overline{T}$ + R	$\overline{CTG}\uparrow$ BEFORE T.O.
1	1	1			T.O. BEFORE $\overline{CTG}\uparrow$

R = RESET CONDITION
W = WRITE TIMER LATCHES
T.O. = COUNTER TIME OUT

$\overline{CTG}\downarrow$ = NEG TRANSITION OF PIN 17
$\overline{CTG}\uparrow$ = POS TRANSITION OF PIN 17
$\overline{T}$ = 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 17.

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

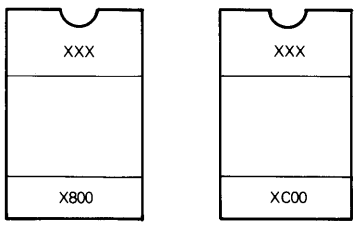
A copy of the cover letter should also be mailed separately.

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 17 — 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)

	ROM SECTION 1	ROM SECTION 0	I/O-TIMER SECTION 1	I/O-TIMER SECTION 0
CS0	☐	☐	☐	☐
CS1	☐	☐	☐	☐

I/O-TIMER SELECT		CHECK ONE COLUMN ONLY				
A6: 1 ☐ 0 ☐ X ☐		☐	☐	☐	☐	☐
	A10	X	1	X	X	X
	A9	X	X	1	X	X
	A8	X	X	X	1	X
	A7	X	X	X	X	1

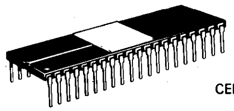
1 ≥ 2.0V.
0 ≤ 0.8V.
X = NOT USED

MOTOROLA

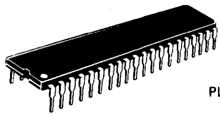
MC6847
Non-Interlace
MC6847Y
Interlace

MOS
(N-CHANNEL, SILICON-GATE)

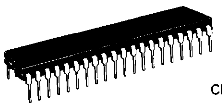
VIDEO DISPLAY GENERATOR



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711



S SUFFIX
CERDIP PACKAGE
CASE 734

3

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.

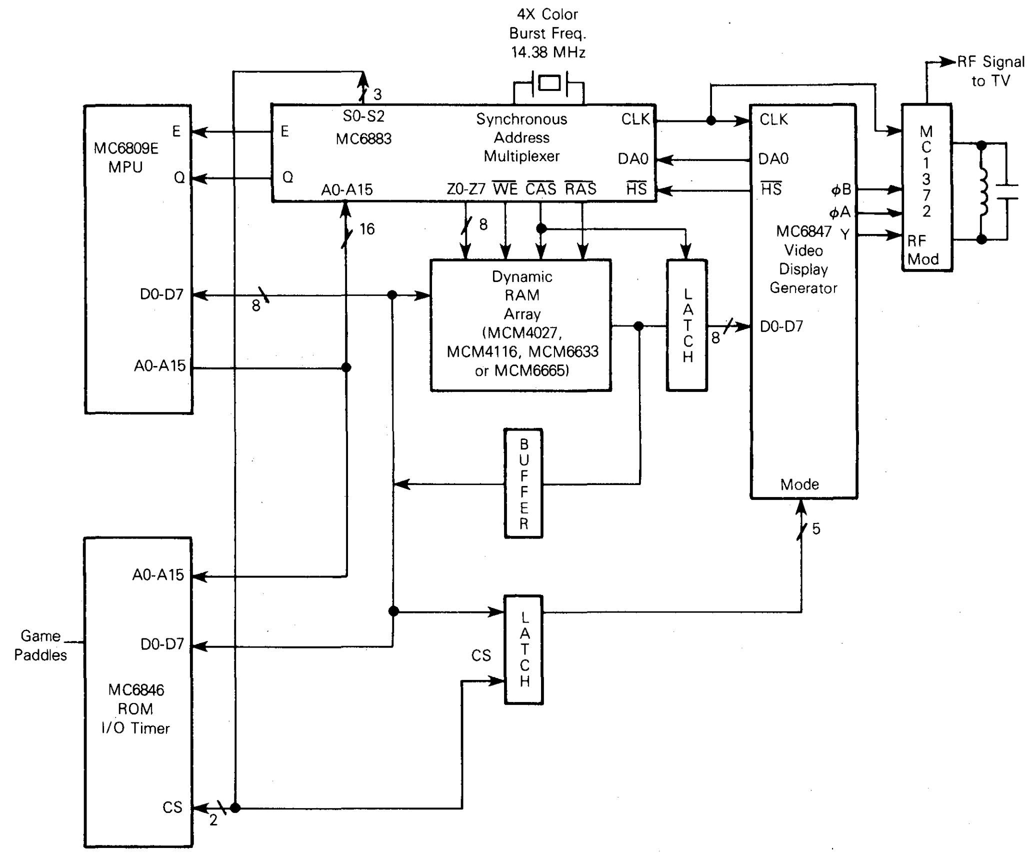
- Compatible with the M6800 Family, the M68000 Family, and Other Microprocessor Families
- Generates Four Different Alphanumeric Display Modes, Two Semigraphic Modes, and Eight Graphic Display Modes
- The Alphanumeric Modes Display 32 Characters Per Line by 16 Lines Using Either the Internal ROM or an 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

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	40	DD7
DD6	2	39	CSS
DD0	3	38	$\overline{HS}$
DD1	4	37	$\overline{FS}$
DD2	5	36	$\overline{RP}$
DD3	6	35	$\overline{A}$/G
DD4	7	34	$\overline{A}$/S
DD5	8	33	CLK
CHB	9	32	INV
ϕB	10	31	$\overline{INT}$/EXT
ϕA	11	30	GM0
$\overline{MS}$	12	29	GM1
DA5	13	28	Y
DA6	14	27	GM2
DA7	15	26	DA4
DA8	16	25	DA3
V_{CC}	17	24	DA2
DA9	18	23	DA1
DA10	19	22	DA0
DA11	20	21	DA12

3

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}	 $V_{SS}-0.3$ $V_{SS}-0.3$	 – –	 $V_{SS}+0.6$ $V_{SS}+0.8$	V
Input Leakage Current, Force 5.25 V on Pin Under Test, V_{CC} = 5.5 V CLK, GM0-GM2, INV, $\overline{INT}$/EXT, $\overline{MS}$, V_{SS}, DD0-DD7, $\overline{A}$/S, $\overline{A}$/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) $\overline{RP}$, $\overline{HS}$, $\overline{FS}$	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.

3

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

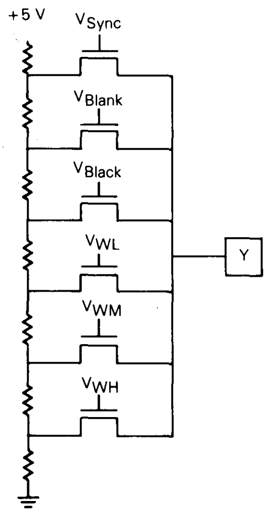
Solving equations 1 and 2 for K gives:

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

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

3

FIGURE 2 — PSEUDO ANALOG LUMINANCE RESISTOR CHAIN

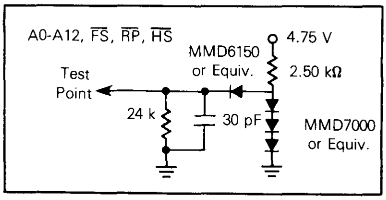


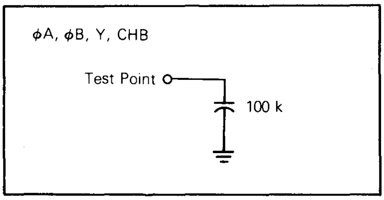
NOTE: The chrominance output chain is similar in design to the luminance chain.

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

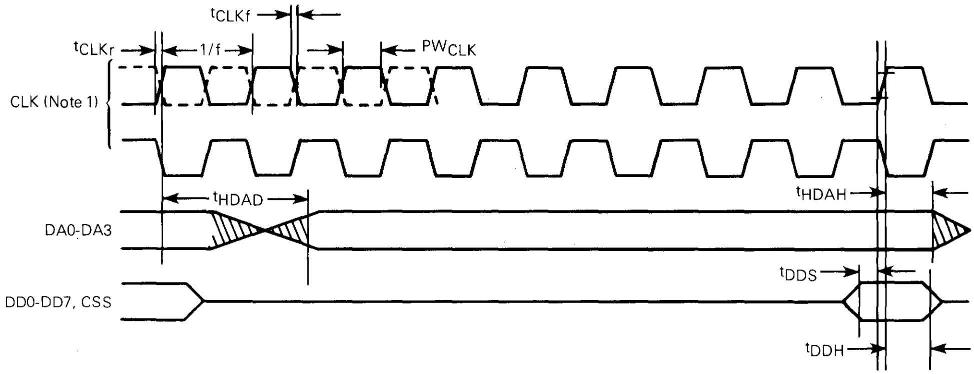
FIGURE 3 — TEST LOADS





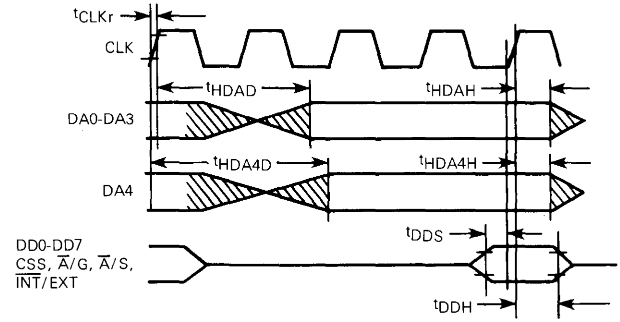
3

FIGURE 4 — CLOCK AND LONG CYCLE HORIZONTAL ACCESS TIMING



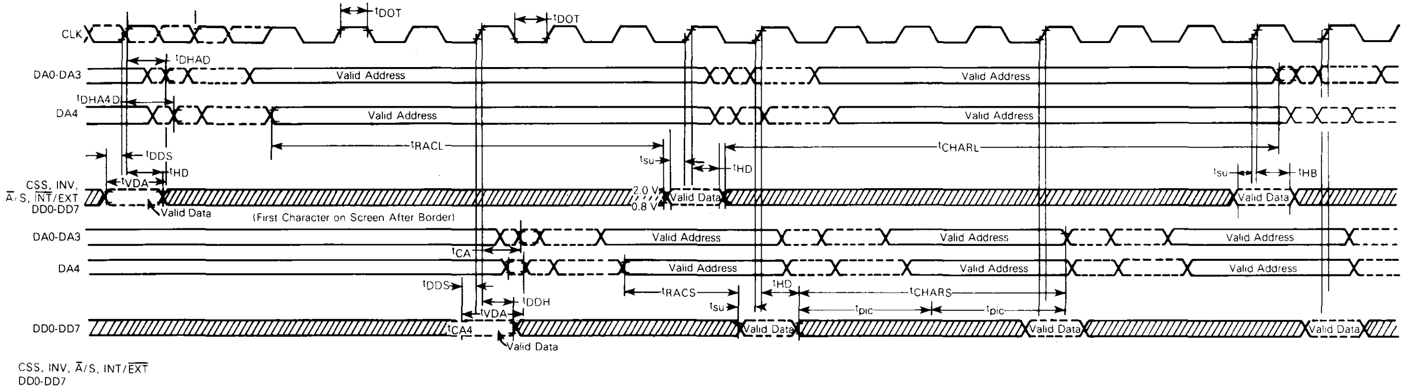
NOTES:
1. The VDG may power-up using either the rising or falling edge of the clock (dotted line).
2. Transitions of DA4-DA12 occur outside the display area. DA0-DA3 access the 16 bytes of data displayed during each scan line in the display area.
3. Long cycle timing applies to CG1, RG1, RG2, and RG3 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:
   $t_{RACL} = 8 \bullet 1/f_{max} - t_{HDAD_{max}} - t_{DDS_{min}} - t_{CLKr}$
   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 5 — SHORT CYCLE HORIZONTAL ACCESS TIMING



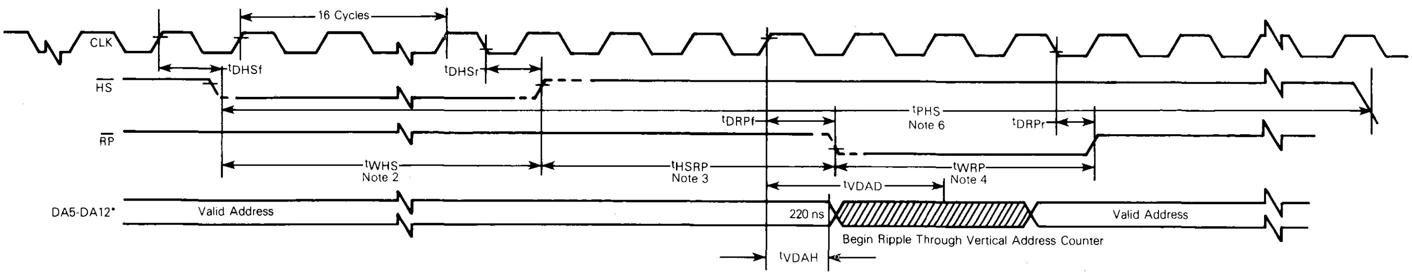
NOTES:
1. The VDG may power-up using either the rising or falling edge of the clock as shown in Figure 4.
2. Transitions of DA5-DA12 occur outside the display area. DA0-DA4 access the 32 bytes of data displayed during each scan line in the display area.
3. Short cycle timing applies to the four alphanumeric 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:
   $t_{RACS} = 4 \bullet 1/f_{max} - t_{HDA4D_{max}} - t_{DDS_{min}} - t_{CLKr}$
   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

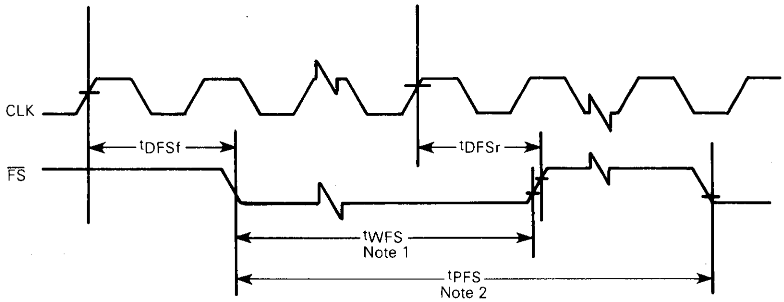
FIGURE 7 — VERTICAL ADDRESS, ROW PRESET AND HORIZONTAL SYNCHRONIZATION TIMING



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. $t_{DOT} = ½ f$.

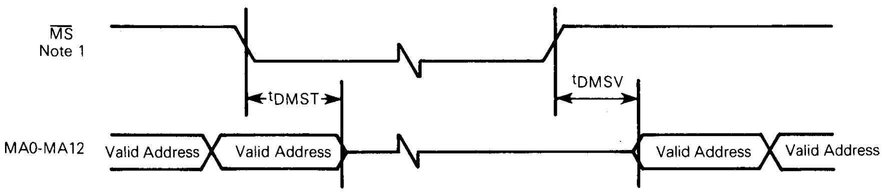
3

FIGURE 8 — FIELD SYNC ($\overline{FS}$) TIMING



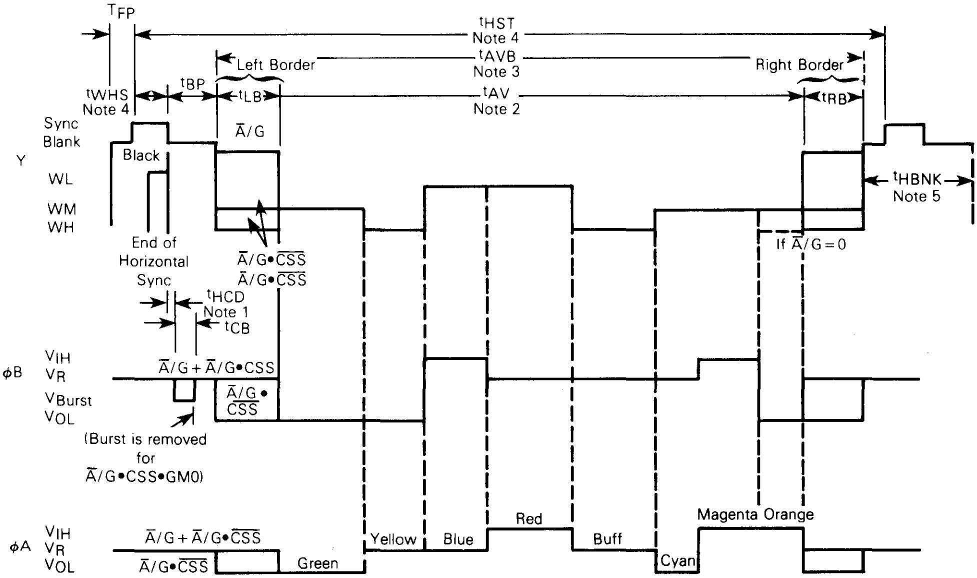
NOTES:
1. $t_{WFS} = 32 \cdot t_{PHST} = 32 \cdot (277.5 \cdot 1/f)$
2. $t_{PFS} = 262 \cdot t_{PHST} = 262 \cdot (227.5 \cdot 1/f)$ for MC6847
   $t_{PFS} = 262.5 \cdot t_{PHST} = 262.5 \cdot (227.5 \cdot 1/f)$ for MC6847Y

FIGURE 9 — MEMORY SELECT ($\overline{MS}$) TIMING



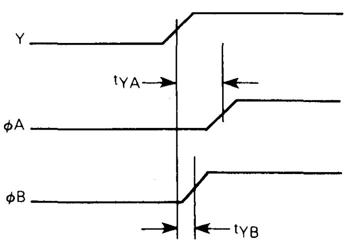
NOTES:
1. MS is asserted asynchronously with respect to CLK.

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



3

FIGURE 12 — TIMING DIAGRAMS
VIDEO RISE AND FALL TIMES (Illustrates Beginning of One Horizontal Line)

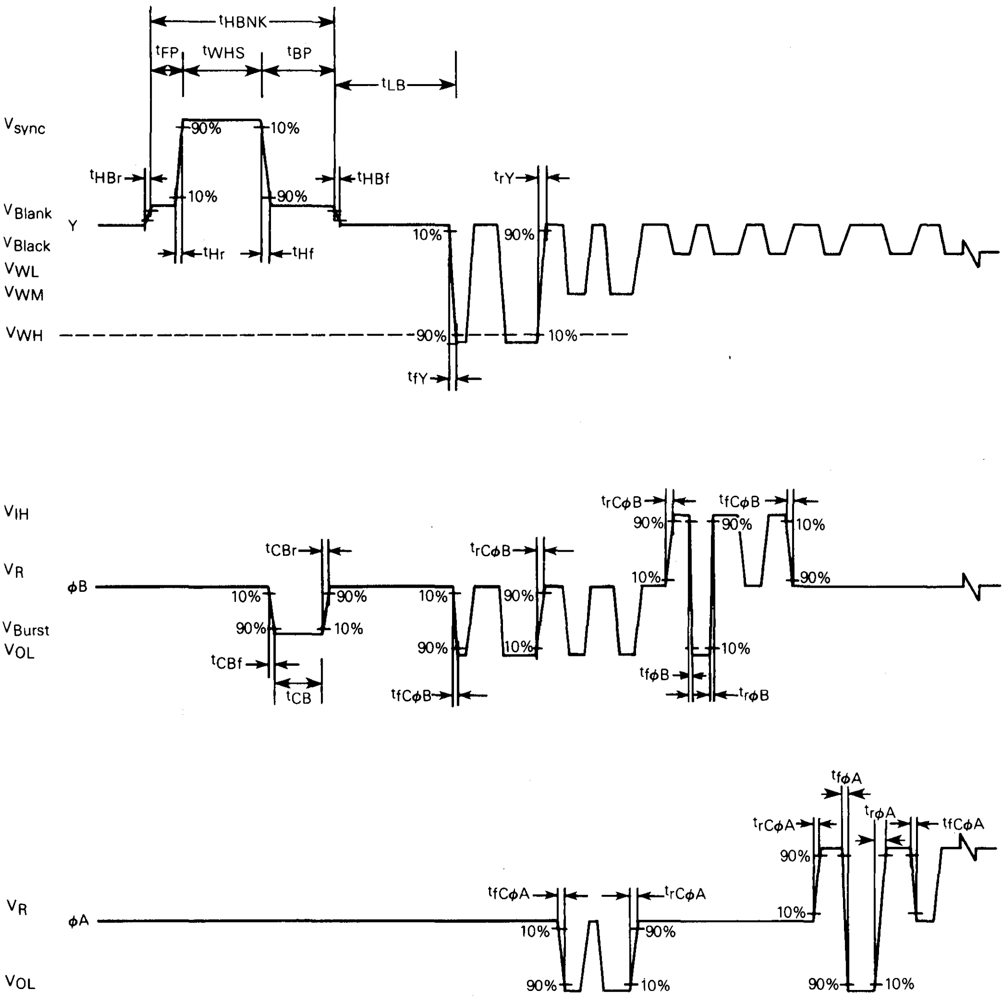
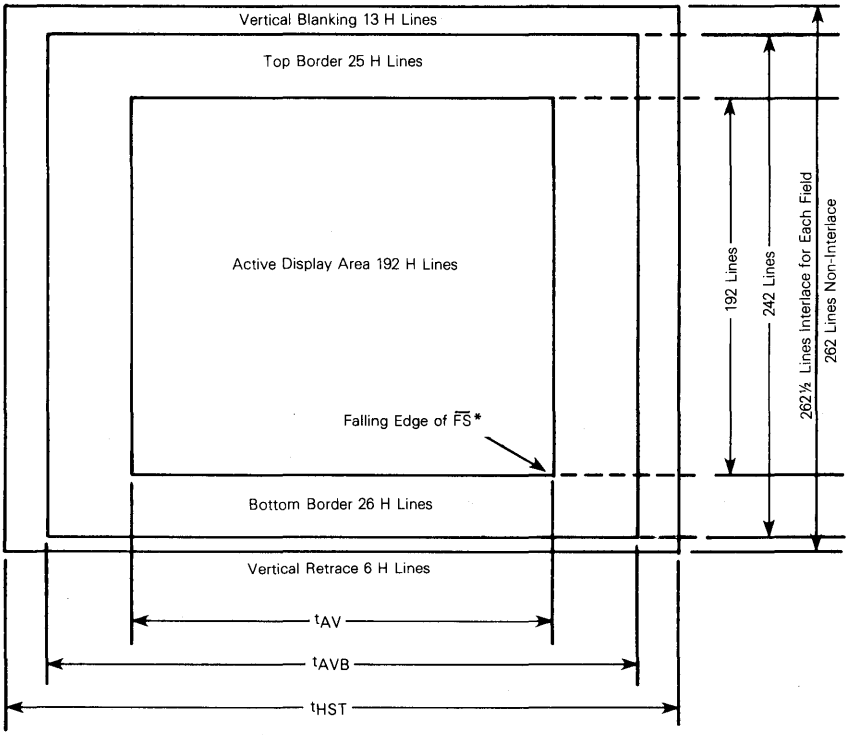


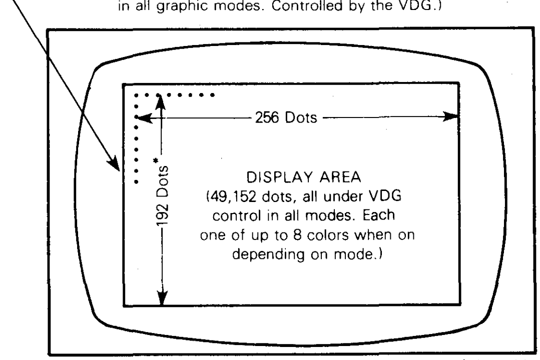
FIGURE 13 — DISPLAY AREA TIMING



*Typically 2.4 μs after start of vertical blank.

FIGURE 14 — TYPICAL FORMAT OF THE TELEVISION SCREEN

BORDER
(Black in all alpha/semigraphic modes. Green or buff (off-white) in all graphic modes. Controlled by the VDG.)



*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.

3

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).

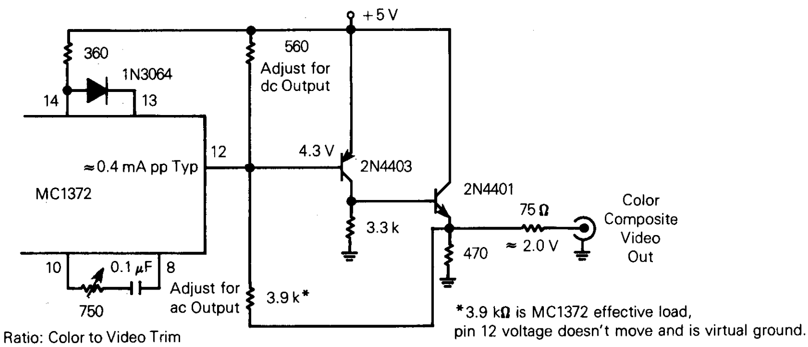
Luminance (Y) — This six level analog output contains composite sync, blanking and four levels of video luminance.

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 — COLOR COMPOSITE VIDEO TO COLOR MONITOR



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.

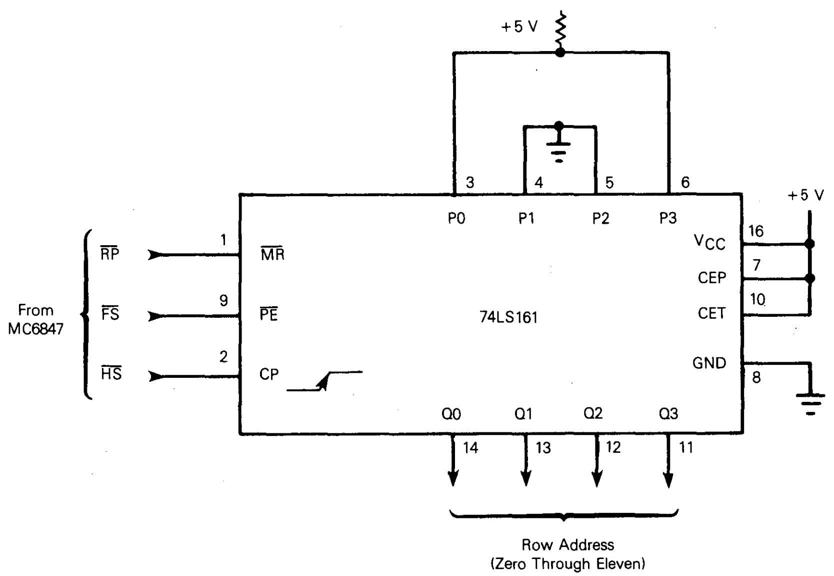
HORIZONTAL SYNC ($\overline{HS}$) — The $\overline{HS}$ pulse coincides with the horizontal synchronization pulse furnished to the television receiver by the VDG (see Figure 7). The high-to-low transition of the $\overline{HS}$ output coincides with the leading edge of the horizontal synchronization 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



3

TABLE 1 — MODE CONTROL LINES (INPUTS)

$\overline{A}$/G	$\overline{A}$/S	$\overline{INT}$/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

3

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 ($\approx$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 ana 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.

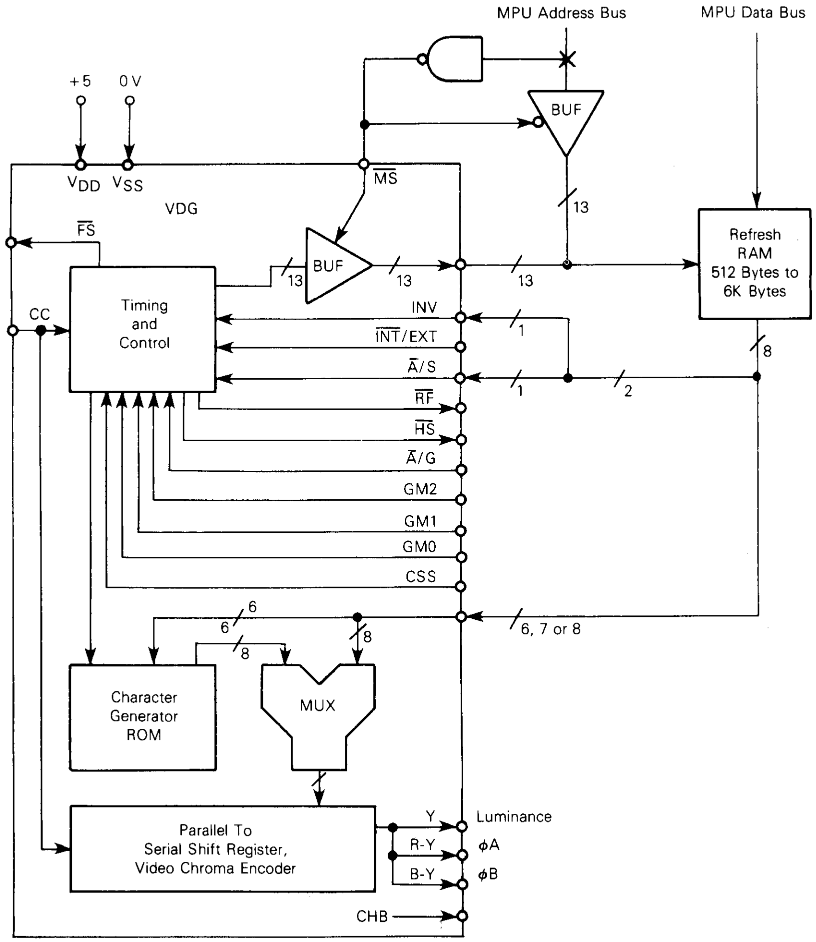
INTERNAL CHARACTER GENERATOR ROM

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

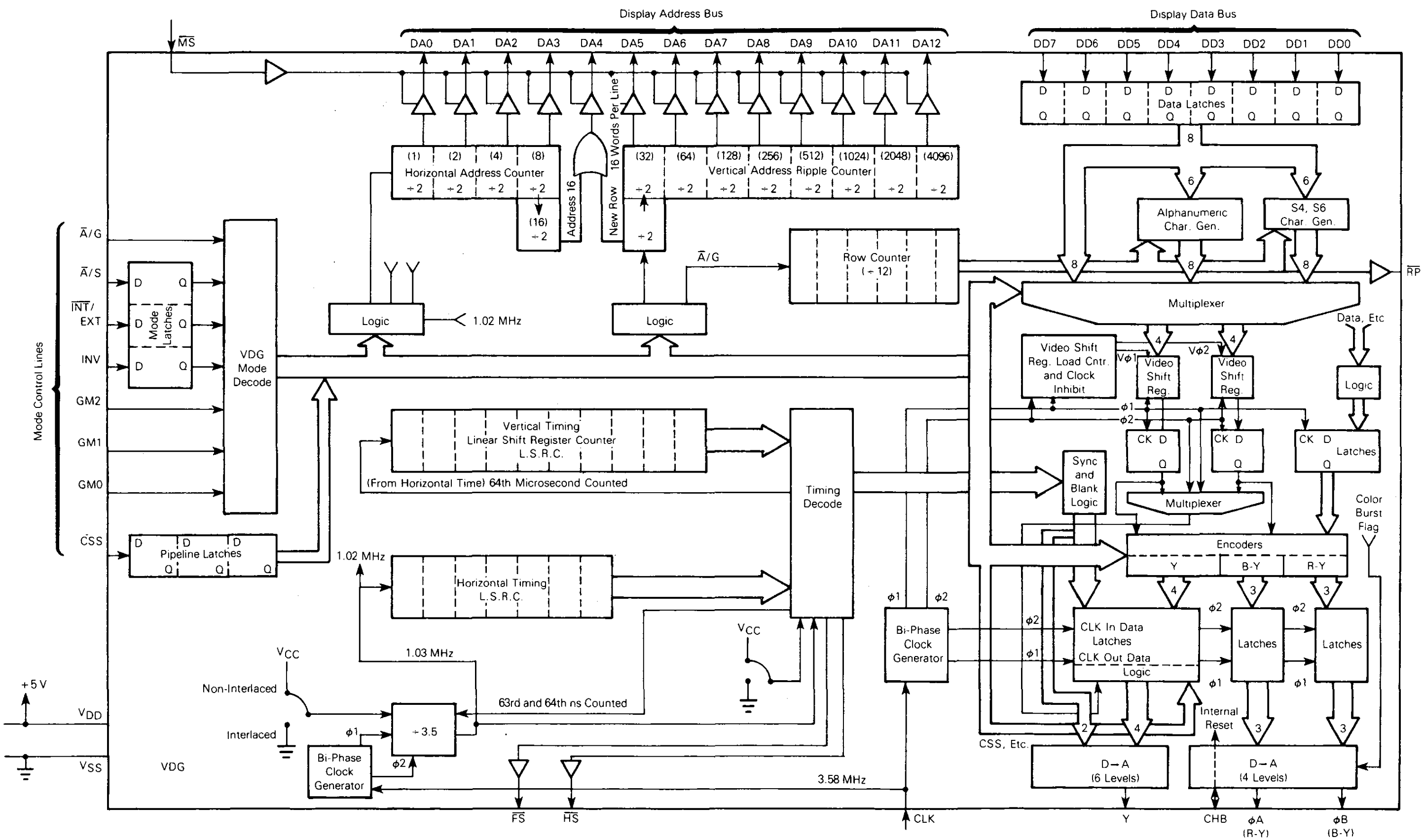
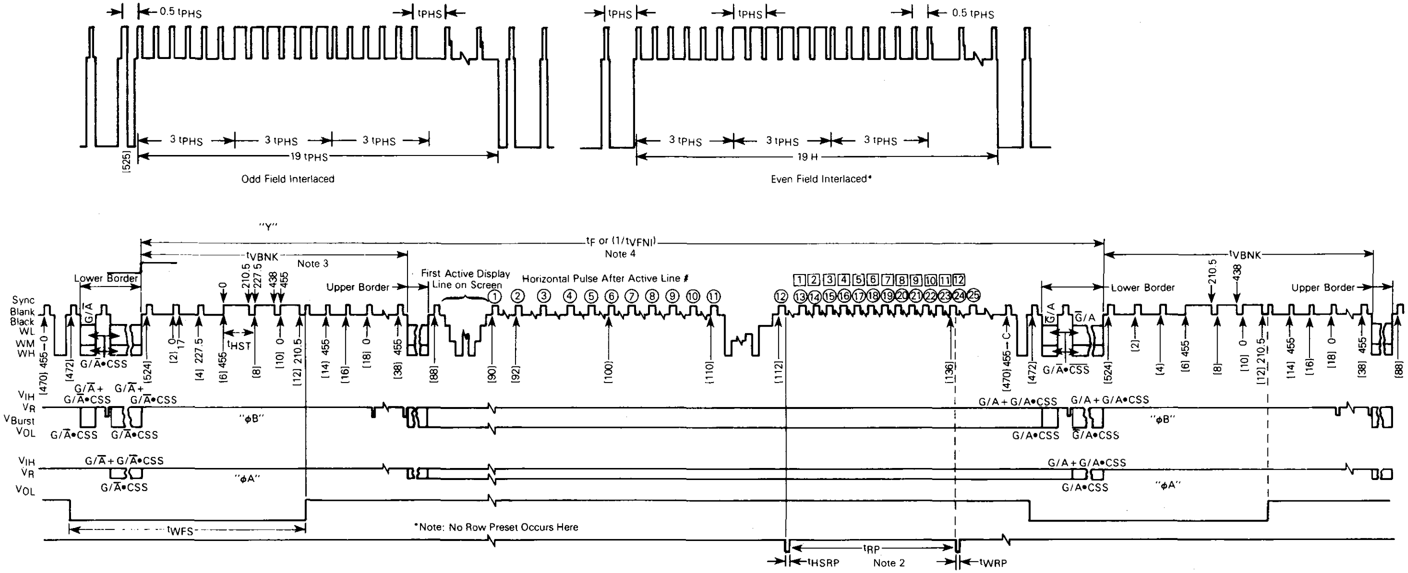


FIGURE 18 — NON-INTERLACE VERTICAL TIMING
(For Interlace Vertical Timing Use Inserts)



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:
   $t_{HST} = (227.5 - 0) \times 279.366\ ns = 63.5\ \mu s$
   $= (455 - 227.5) \times 279.366\ ns = 63.5\ \mu s$
   $t_{WHS} = (455 - 438) \times 279.366\ ns = 4.75\ \mu s$
   Lower Border $= (524 - 472) \times 31.7783\ \mu s - t_{HBNK}$
   $= 1.6525\ ms - 11.6\ ns = 1.64\ ms$

   Upper Border $= (88 - 38) \times 31.7783\ \mu s - t_{HBNK} =$
   $= 1.5889\ \mu s - 11.6\ \mu s = 1.58\ ms$
2. $t_{RP} = 12$ horizontal scan lines.
3. $t_{VBNK} = 20 \cdot t_{PHS} = 20 \cdot (227.5 \cdot 1/f)$
4. $t_F = 262 \cdot t_{PHS} = 262 \cdot (227.5 \cdot 1/f)$ for Non-Interlace.
   $t_F = 262.5 \cdot T_{PHS} = 262.5 \cdot (227.5 \cdot 1/f)$ 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.

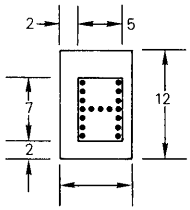
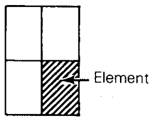
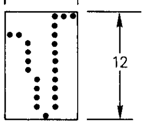
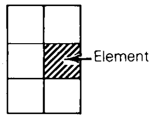
DISPLAY MODES

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 MAJOR MODES
Major Mode 1 — Alpha Modes

Title	Memory	Display Elements	Colors	Title	Memory	Display Elements	Colors
Alphanumerics (Internal)	512 × 8		2	Semigraphic 4	512 × 8		8
Alphanumerics (External)	512 × 8		2	Semigraphic 6	512 × 8		4

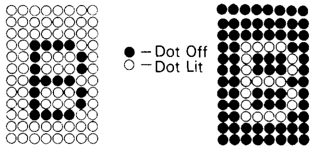
Major Mode 2 — Graphics Modes

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.

FIGURE 19 — ALPHANUMERIC MODE (INTERNAL)

512 Characters (32 × 16)
Typical Character



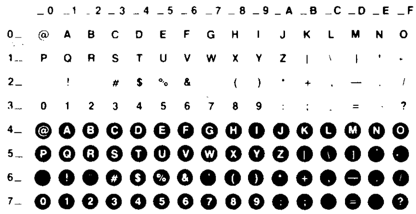
Inverted	Normal
Black Character	Black Background
Orange or Green	Orange or Green
Background (Selectable)	Character (Selectable)

Character Source:
Internal — 6 Bit ASCII Generator ROM On Chip or User Definable
External — Users ROM

FIGURE 20 — AVAILABLE ALPHANUMERICS

MD4 = INV = D4
MD7 = AS = D7

O = Inverted Character — Illuminated Background, Dark Character



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.

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.

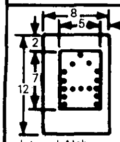
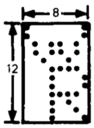
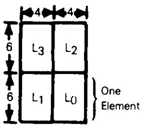
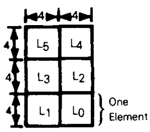
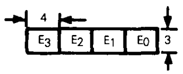
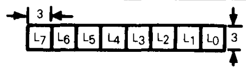
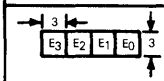
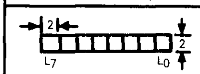
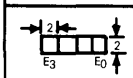
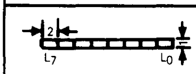
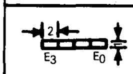
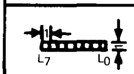
THE 128 × 192 COLOR GRAPHICS SIX (CG6) MODE — The 128 × 192 color graphics mode generates a display 128 elements wide by 192 elements high. Each element may be one of four colors. A 6k × 8 display memory is required. The display RAM is accessed 32 times per horizontal line. Each pixel equals two half-clocks 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.

TABLE 3 — DETAILED DESCRIPTION OF VDG MODES

VDG Pins									Color			
$\overline{MS}$	G/$\overline{A}$	S/$\overline{A}$	EXT/$\overline{INT}$	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

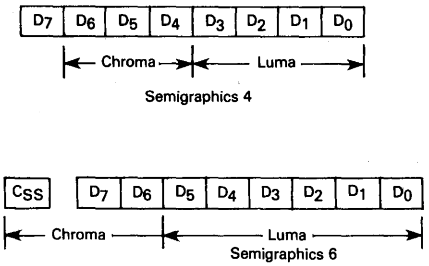
TABLE 3 — DETAILED DESCRIPTION OF VDG MODES
(Continued)

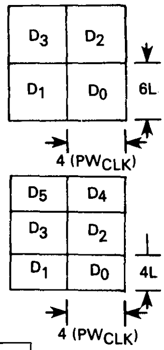
TV Screen Detail	VDG Data Bus	Comments
 Internal Alphanumerics	extra · ASCII Code	The ALPHANUMERIC INTERNAL mode uses an internal character generator (which contains the following five dot by seven dot characters: ,@ABCDEFGHIJKLMNOPQRSTUVWXYZ [\]↑←SP !"#$%&'()*+,–./0123456789:;<=>?. The six bit ASCII code leaves two bits free and these may be externally connected to the mode pins (G/$\overline{A}$, S/$\overline{A}$, EXT/$\overline{INT}$, GM2, GM1, GM0, CSS or INV).
	One Row of Custom Characters	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.
	extra · C2 C1 C0 L3 L2 L1 L0	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.
	C1 C0 L5 L4 L3 L2 L1 L0	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.
	C1 C0 C1 C0 C1 C0 C1 C0	The COLOR GRAPHICS ONE mode uses a maximum of 1024 bytes of display RAM in which one pair of bits specifies one picture element.
	L7 L6 L5 L4 L3 L2 L1 L0	The RESOLUTION GRAPHICS ONE mode uses a maximum of 1024 bytes of display RAM in which one bit specifies one picture element.
	C1 C0 C1 C0 C1 C0 C1 C0	The COLOR GRAPHICS TWO mode uses a maximum of 2048 bytes of display RAM in which one pair of bits specifies one picture element.
	L7 L6 L5 L4 L3 L2 L1 L0	The RESOLUTION GRAPHICS TWO mode uses a maximum of 1536 bytes of display RAM in which one bit specifies one picture element.
	C1 C0 C1 C0 C1 C0 C1 C0	The COLOR GRAPHICS THREE mode uses a maximum of 3072 bytes of display RAM in which one pair of bytes specifies one picture element.
	L7 L6 L5 L4 L3 L2 L1 L0	The RESOLUTION GRAPHICS THREE mode uses a maximum of 3072 bytes of display RAM in which one bit specifies one picture element.
	C1 C0 C1 C0 C1 C0 C1 C0	The COLOR GRAPHICS SIX mode uses a maximum of 6144 bytes of display RAM in which one pair of bits specifies one picture element.
	L7 L6 L5 L4 L3 L2 L1 L0	The RESOLUTION GRAPHICS SIX mode uses a maximum of 6144 bytes of display RAM in which one bit specifies one picture element.

3

FIGURE 21 — SEMIGRAPHIC MODE ENCODING

(a) Data and Display Formats





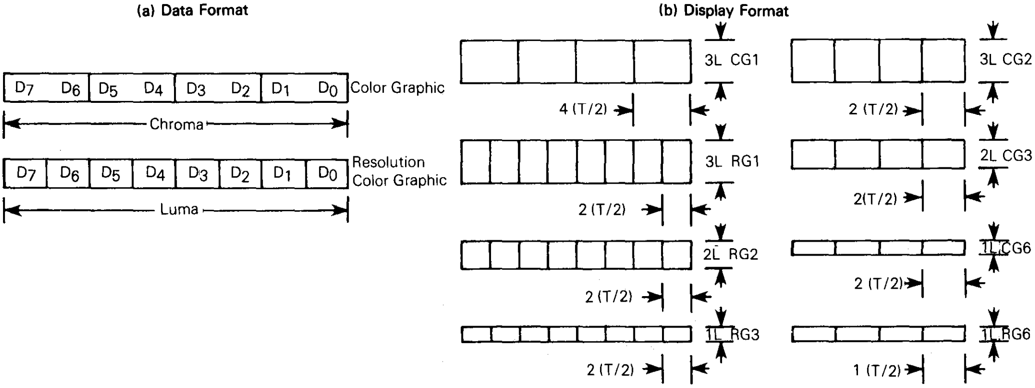
(b) Color Selection

Luma DN	SG4 D6	SG4 D5	SG4 D4	SG6 CSS	SG6 D7	SG6 D6	Color
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



Long Element Modes

Short Element Modes

(c) Color Selection

CSS	Border	Resolution Color Mode DN	Resolution Color Mode Color	Color Mode D_{N+1}	Color Mode D_N	Color Mode 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 ($\overline{RP}$) generated by the VDG initializes the row scan counter for the external character generator once every twelve scan lines, while the negative horizontal sync ($\overline{HS}$) acts as clock to this counter. The negative field sync ($\overline{FS}$) 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).

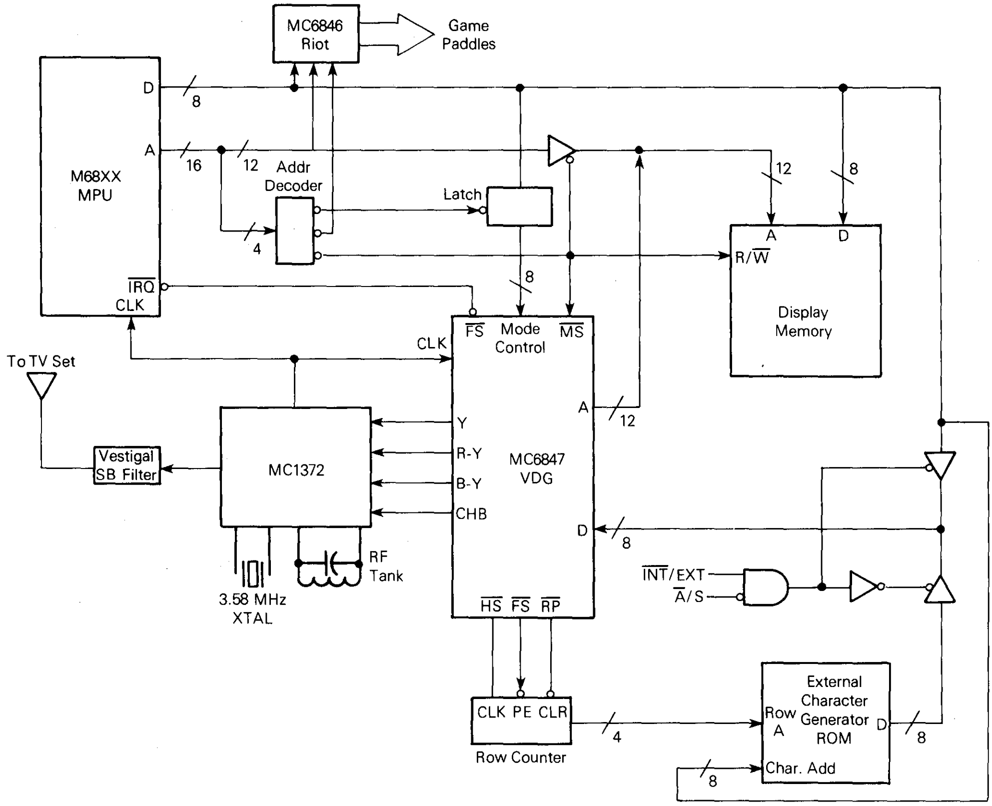
In addition to being a transparent memory access and dynamic memory controller, the SAM also functions as an external clock generator for the MC6800/6809E (slight additional circuitry is required for the MC6800).

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

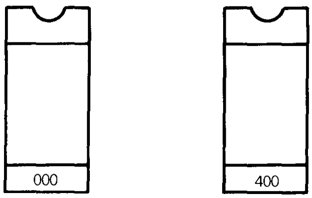
MDOS disk file

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.

3

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

MOS
(N-CHANNEL, SILICON-GATE)

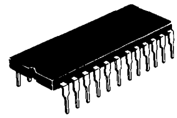
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

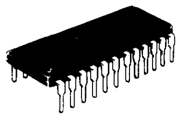
The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

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.

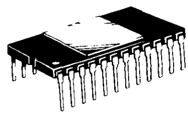
- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Deletion
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation



S SUFFIX
CERDIP PACKAGE
CASE 623

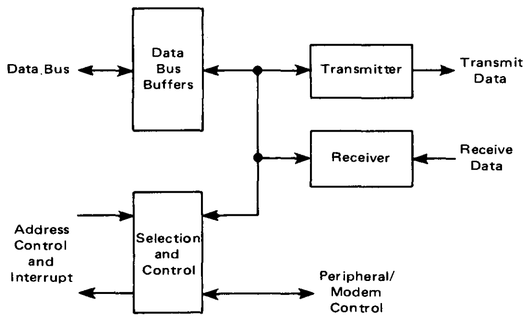


P SUFFIX
PLASTIC PACKAGE
CASE 709

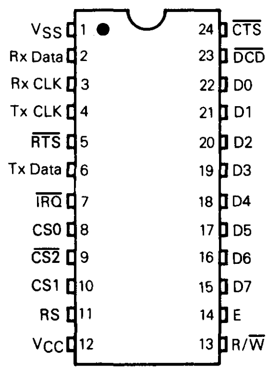


L SUFFIX
CERAMIC PACKAGE
CASE 716

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	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:

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

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
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
Hi-Z (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 = 0°C)		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/$\overline{W}$, 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

*For temperatures less than T_A = 0°C, P_{INT} maximum will increase.

SERIAL DATA TIMING CHARACTERISTICS

Characteristic		Symbol	MC6850		MC68A50		MC68B50		Unit
			Min	Max	Min	Max	Min	Max	
Data Clock Pulse Width, Low	÷16, ÷64 Modes	PW_{CL}	600	–	450	–	280	–	ns
(See Figure 1)	÷1 Mode		900	–	650	–	500	–	
Data Clock Pulse Width, High	÷16, ÷64 Modes	PW_{CH}	600	–	450	–	280	–	ns
(See Figure 2)	÷1 Mode		900	–	650	–	500	–	
Data Clock Frequency	÷16, ÷64 Modes	f_C	–	0.8	–	1.0	–	1.5	MHz
	÷1 Mode		–	500	–	750	–	1000	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

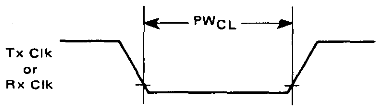


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE

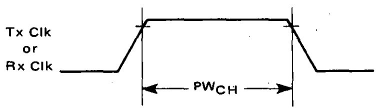


FIGURE 3 — TRANSMIT DATA OUTPUT DELAY

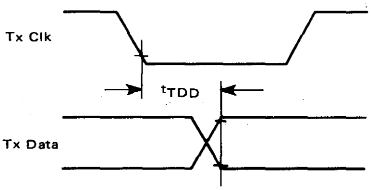


FIGURE 4 — RECEIVE DATA SETUP TIME
(÷1 Mode)

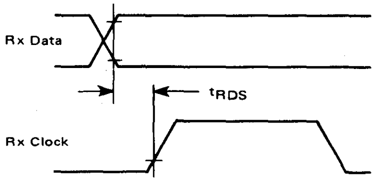


FIGURE 5 — RECEIVE DATA HOLD TIME
(÷1 Mode)

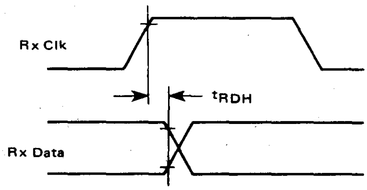
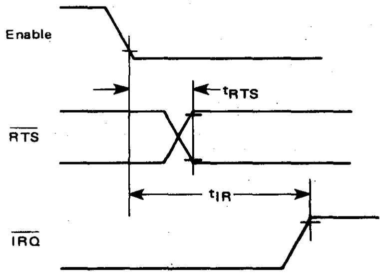


FIGURE 6 — REQUEST-TO-SEND DELAY AND INTERRUPT-REQUEST RELEASE TIMES



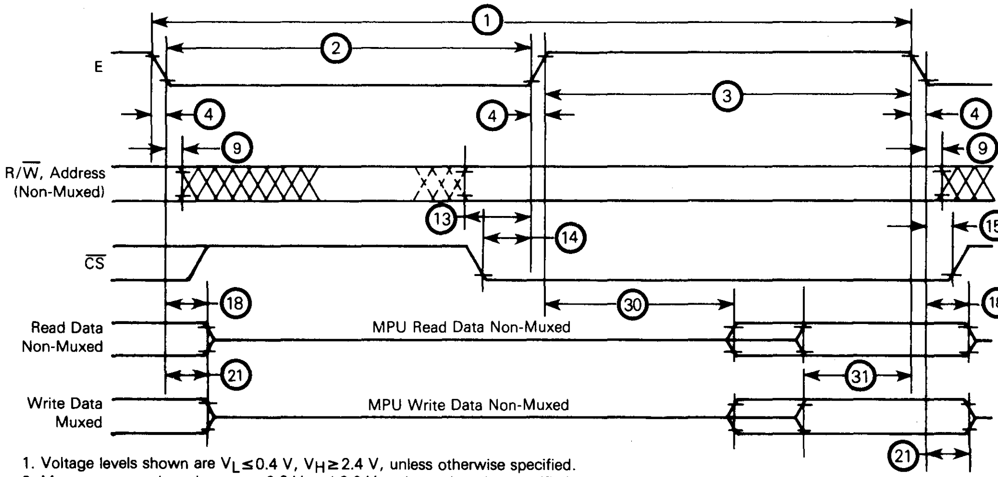
Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

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

Ident. Number	Characteristic	Symbol	MC6850 Min	MC6850 Max	MC68A50 Min	MC68A50 Max	MC68B50 Min	MC68B50 Max	Unit
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

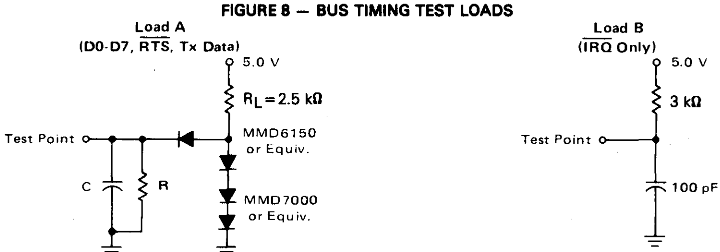
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

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.

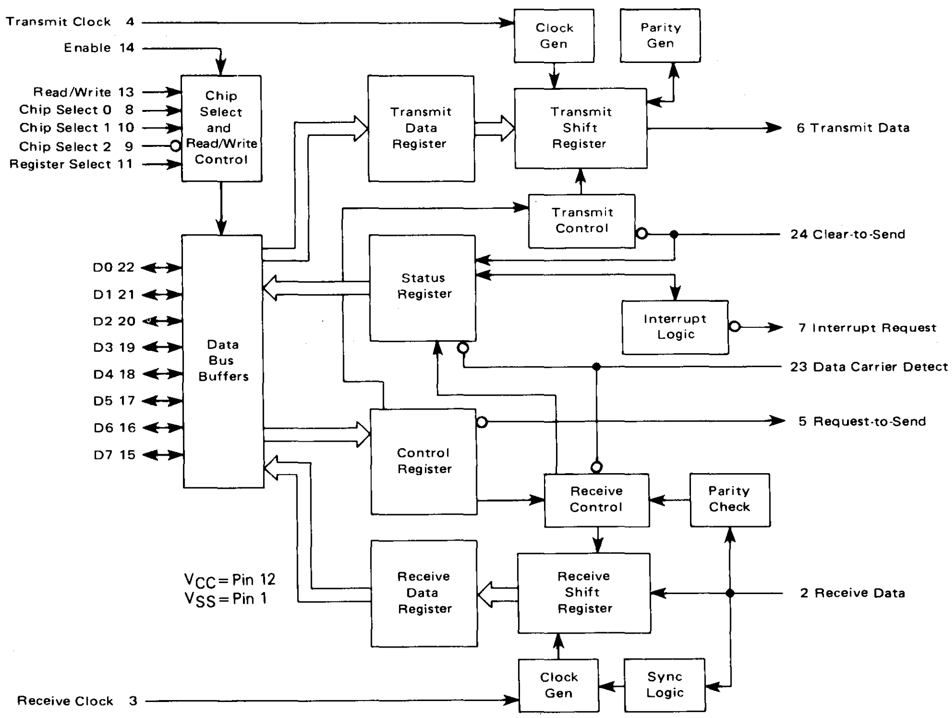
FIGURE 8 — BUS TIMING TEST LOADS



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 ($CR5 \bullet \overline{CR6}$), 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 $\div 1$ 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 ($\overline{CTS}$) — 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 ($\overline{RTS}$) — 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 $\overline{RTS}$ output is low (the active state). This output can also be used for Data Terminal Ready (DTR).

Data Carrier Detect ($\overline{DCD}$) — 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 $\overline{DCD}$ 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 $\overline{DCD}$ 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/$\overline{W}$ 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/$\overline{W}$ 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: RS • R/$\overline{W}$ Transmit Data Register (Write Only)	RS • R/W Receive Data Register (Read Only)	$\overline{RS}$ • R/$\overline{W}$ Control Register (Write Only)	$\overline{RS}$ • R/W Status Register (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 ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear-to-Send ($\overline{CTS}$)
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 ($\overline{IRQ}$)

* 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.

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6850L
	1.0	−40°C to 85°C	MC6850CL
	1.5	0°C to 70°C	MC68A50L
	1.5	−40°C to 85°C	MC68A50CL
	2.0	0°C to 70°C	MC68B50C
Cerdip S Suffix	1.0	0°C to 70°C	MC6850S
	1.0	−40°C to 85°C	MC6850CS
	1.5	0°C to 70°C	MC68A50S
	1.5	−40°C to 85°C	MC68A50CS
	2.0	0°C to 70°C	MC68B50S
Plastic P Suffix	1.0	0°C to 70°C	MC6850P
	1.0	−40°C to 85°C	MC6850CP
	1.5	0°C to 70°C	MC68A50P
	1.5	−40°C to 85°C	MC68A50CP
	2.0	0°C to 70°C	MC68B50P

MOTOROLA

MC68HC51

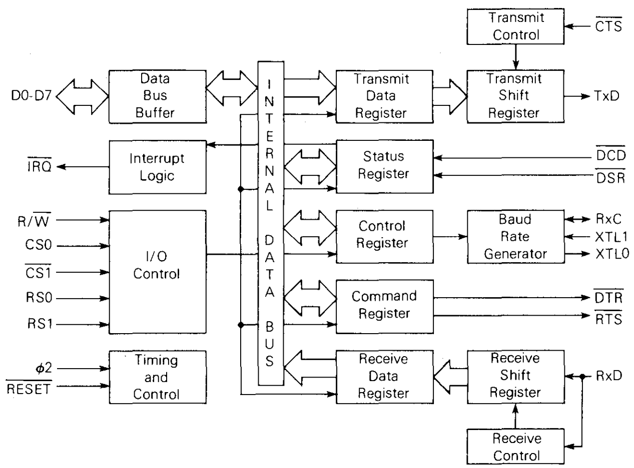
Product Preview

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC68HC51 ACIA provides a program-controlled interface between 8-bit microprocessor-based systems, serial communication data sets, and modems. An on-chip crystal oscillator and a baud-rate generator allow the MC68HC51 to transmit at 15 different program-selected rates, ranging from 50 to 19,200 baud. The MC68HC51 can receive at either the transmit rate or at 16 times an external clock rate.

- Compatible With 8-Bit Microprocessors
- Full-Duplex or Half-Duplex Operation With Buffered Receiver and Transmitter
- Fifteen Programmable Baud Rates (50 to 19,200)
- Receiver Data Rate May Be Identical to Baud Rate or May Be 16 Times the External Clock Input
- Data Set/Modem Control Functions
- Programmable Word Lengths, Number of Stop Bits, and Parity Bit Generation and Detection
- Programmable Interrupt Control
- Software Reset
- Program-Selectable Serial Echo Mode
- Two Chip Selects
- 2 MHz or 1 MHz Clock Rate
- Single +5 Volt ±5% Power Supply
- Full TTL Compatibility

FIGURE 1 — BLOCK DIAGRAM

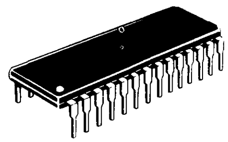


This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

HCMOS

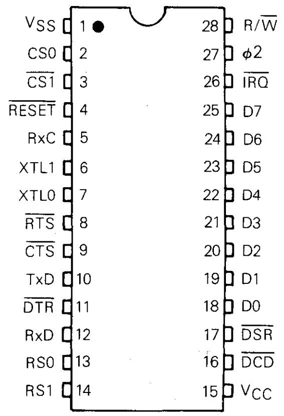
(HIGH DENSITY CMOS SILICON-GATE)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)



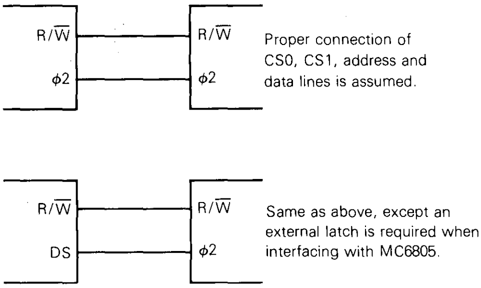
P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENT



3

FIGURE 2 — M6800 SERIES INTERFACE REQUIREMENTS



SIGNAL DESCRIPTIONS

The following paragraphs provide a brief description of the input and output signals for the MC68HC51.

RESET ($\overline{RESET}$)

During system initialization, a low on the $\overline{RESET}$ input causes the internal registers to be cleared.

INPUT CLOCK (ϕ2)

The input clock is the system phase 2 clock and is used to synchronize all data transfers.

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

The R/$\overline{W}$ is generated by the microprocessor and is used to control the direction of data transfers. A high on the R/$\overline{W}$ pin allows the processor to read the data supplied by the ACIA. A low on the R/$\overline{W}$ pin allows a write to the ACIA.

INTERRUPT REQUEST ($\overline{IRQ}$)

The $\overline{IRQ}$ pin is an interrupt output from the interrupt control logic. It is an open-drain output permitting several devices to be connected to the common $\overline{IRQ}$ microprocessor input. Normally a high level, $\overline{IRQ}$ goes low when an interrupt occurs.

DATA BUS (D0-D7)

The D0-D7 pins are the eight data lines used to transfer data. These lines are bidirectional and are normally in the high-impedance state, except during read cycles when the ACIA is selected.

DATA SET READY ($\overline{DSR}$)

The $\overline{DSR}$ input pin is used to indicate to the ACIA the status of the modem. A low indicates the "ready" state and a high "not-ready". $\overline{DSR}$ is a high-impedance input, and must be connected. If unused, it should be driven high or low but not switched.

DATA CARRIER DETECT ($\overline{DCD}$)

The $\overline{DCD}$ input pin is used to indicate to the ACIA the status of the carrier-detect output of the modem. A low indicates that the modem carrier signal is present and a high that it is not. Like $\overline{DSR}$, $\overline{DCD}$ is a high-impedance input and must be connected.

REQUEST TO SEND ($\overline{RTS}$)

The $\overline{RTS}$ output pin is used to control the modem from the processor. The state of the $\overline{RTS}$ pin is determined by the contents of the command register.

CLEAR TO SEND ($\overline{CTS}$)

The $\overline{CTS}$ input pin is used to control the transmitter operation. The enable state is with $\overline{CTS}$ low. The transmitter is automatically disabled if $\overline{CTS}$ is high.

DATA TERMINAL READY ($\overline{DTR}$)

This output pin is used to indicate the status of the ACIA to the modem. A low on $\overline{DTR}$ indicates the ACIA is enabled and a high indicates it is disabled. The processor controls this pin via bit 0 of the command register.

CHIP SELECTS (CS0, $\overline{CS1}$)

The two chip-select inputs are normally connected to the processor address lines either directly or through decoders. The ACIA is selected when CS0 is high and $\overline{CS1}$ is low.

REGISTER SELECTS (RS0, RS1)

The two register-select lines are normally connected to the processor address lines to allow the processor to select the various ACIA internal registers. The following table indicates the internal register-select coding:

RS1	RS0	Write	Read
0	0	Transmit Data Register	Received Data Register
0	0	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

Note that only the command and control registers are read/write. The programmed reset operation does not cause any data transfer, but is used to clear bits 0 through 4 in the command register and bit 2 in the status register.

CRYSTAL PINS (XTL1, XTL0)

These pins are normally directly connnected to the external crystal (1.8432 megahertz) used to derive the various baud rates. Alternatively, an externally generated clock may be used to drive the XTL1 pin in which case the XTL0 pin must float. XTL1 is the input pin for the transmit clock.

TRANSMIT DATA (TxD)

The TxD output line is used to transfer serial non-return-to-zero (NRZ) data to the modem. The least significant bit

(LSB) of the transmit data register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected, or under control of an external clock (as selected by the control register).

RECEIVE DATA (RxD)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or the rate of an externally generated receiver clock (as selected by the control register).

RECEIVE CLOCK (RxC)

The RxC is a bidirectional pin which serves as either the receiver 16x clock input or the receiver 16x clock output. The latter mode results if the internal baud-rate generator is selected for receiver data clocking.

MAIN DATA/CONTROL REGISTERS

A brief description of the main MC68HC51 data and control registers follows.

TRANSMIT DATA REGISTER

This 8-bit register provides temporary storage for the data to be transmitted. Bit 0 is the leading bit to be transmitted. Unused bits are the high-order bits and are "don't care" for transmission.

RECEIVE DATA REGISTER

This 8-bit register provides temporary storage for the data being received. Bit 0 is the leading bit received. Unused bits are the high-order bits and are "zeros" for the receiver. Parity bits are not contained in the receive data register but are stripped off after being used for parity checking. Thus, former parity bits become unused "zero" bits in the receive data register.

COMMAND REGISTER

This 8-bit register contains the command word received from the controlling microprocessor. The command word specifies the specific modes and functions the MC68HC51 is to assume. Included are data terminal ready, transmitter interrupt disabled, receiver echo mode, and parity disabled.

CONTROL REGISTER

This 8-bit register contains, message format information received from the microprocessor, and includes: baud rate, clock source, word length, and number of stop bits. This information is used by the MC68HC51 for synchronization and proper processing of message data.

STATUS REGISTER

This 8-bit register contains the current status of the MC68HC51 and the related modem. This register is continuously accessed by the controlling microprocessor during operation to determine if data processing is being performed properly or if errors have occurred. Status indications include: parity error, framing error, overrun, clear to send, transmit register empty, receive register full, data carrier detect, and interrupt request.

MOTOROLA

MC6852

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.

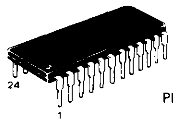
Typical applications include floppy disk controllers, cassette or cartridge tape controllers, data communications terminals, and numerical control systems.

- 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

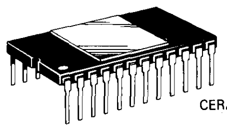
MOS

(N-CHANNEL, SILICON-GATE)

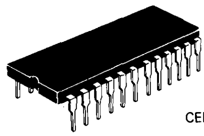
SYNCHRONOUS SERIAL DATA ADAPTER



P SUFFIX
PLASTIC PACKAGE
CASE 709



L SUFFIX
CERAMIC PACKAGE
CASE 716



S SUFFIX
CERDIP PACKAGE
CASE 623

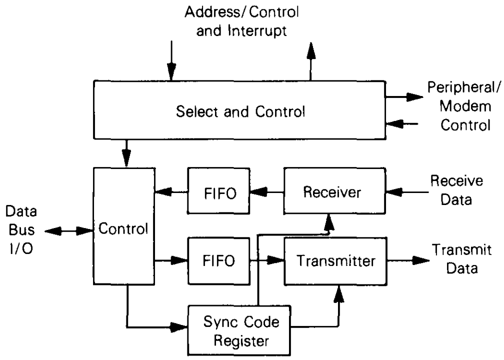
ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6852L
	1.0	−40°C to 85°C	MC6852CL
	1.5	0°C to 70°C	MC68A52L
	1.5	−40°C to 85°C	MC68A52CL
	2.0	0°C to 70°C	MC68B52C
Cerdip S Suffix	1.0	0°C to 70°C	MC6852S
	1.0	−40°C to 85°C	MC6852CS
	1.5	0°C to 70°C	MC68A52S
	1.5	−40°C to 85°C	MC68A52CS
	2.0	0°C to 70°C	MC68B52S
Plastic P Suffix	1.0	0°C to 70°C	MC6852P
	1.0	−40°C to 85°C	MC6852CP
	1.5	0°C to 70°C	MC68A52P
	1.5	−40°C to 85°C	MC68A52CP
	2.0	0°C to 70°C	MC68B52P

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	24	$\overline{CTS}$
Rx Data	2	23	$\overline{DCD}$
Rx CLK	3	22	D0
Tx CLK	4	21	D1
SM/$\overline{DTR}$	5	20	D2
Tx Data	6	19	D3
$\overline{IRQ}$	7	18	D4
TUF	8	17	D5
$\overline{RESET}$	9	16	D6
$\overline{CS}$	10	15	D7
RS	11	14	E
V_{CC}	12	13	R/$\overline{W}$

SYNCHRONOUS SERIAL DATA ADAPTER 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 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

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Package Ceramic Package Cerdip Package	θ_{JA}	 120 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 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}).

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
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
Hi-Z (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)	D0-D7	V_{OH}	V_{SS}+2.4	–	–	V
(I_{Load} = −100 μA, Enable Pulse Width < 25 μs)	TX Data, $\overline{DTR}$, TUF		V_{SS}+2.4	–	–	
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 = 0°C)*		P_{INT}	–	300	525*	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Other Inputs		–	–	7.5	
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	Tx Data, SM/$\overline{DTR}$, TUF	C_{out}	–	–	10	pF
	$\overline{IRQ}$				5.0	

*For temperatures below 0°C, the maximum value of P_{INT} will increase.

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 Min	MC6852 Max	MC68A52 Min	MC68A52 Max	MC68B52 Min	MC68B52 Max	Unit
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
$\overline{DTR}$ 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
$\overline{RESET}$ Pulse Width	t_{RESET}	1.0	–	0.666	–	0.500	–	μs
$\overline{CTS}$ Setup Time (Figure 6)	t_{CTS}	200	–	150	–	120	–	ns
$\overline{DCD}$ 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

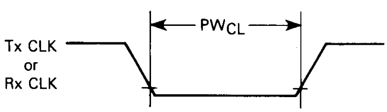
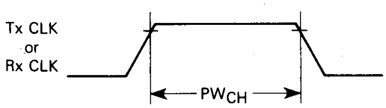


FIGURE 2 — CLOCK PULSE WIDTH, HIGH-STATE



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 — RECEIVE DATA SETUP AND HOLD TIMES AND SYNC MATCH DELAY TIME

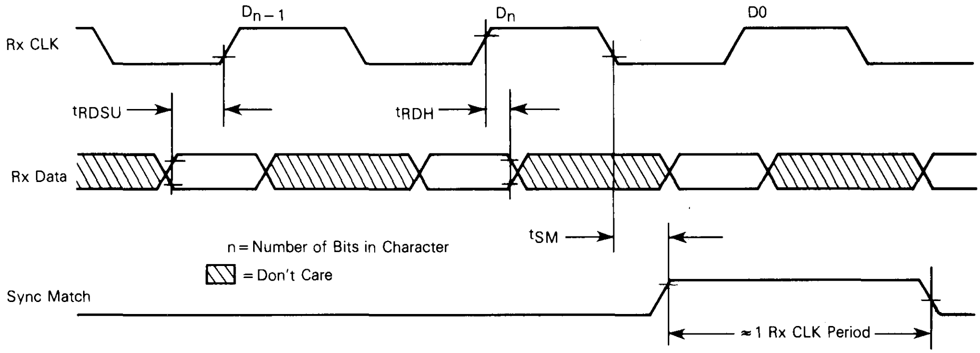
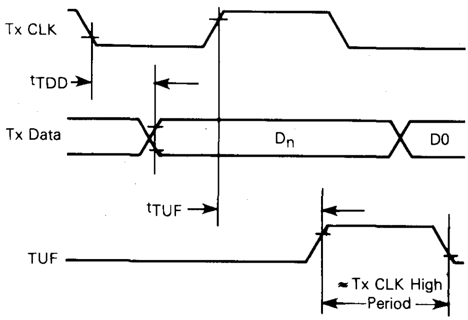


FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND TRANSMITTER UNDERFLOW DELAY TIME



n = Number of bits in character

FIGURE 5 — DATA TERMINAL READY AND INTERRUPT REQUEST RELEASE TIMES

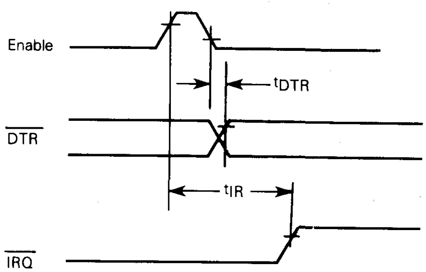


FIGURE 6 — CLEAR-TO-SEND SETUP TIME

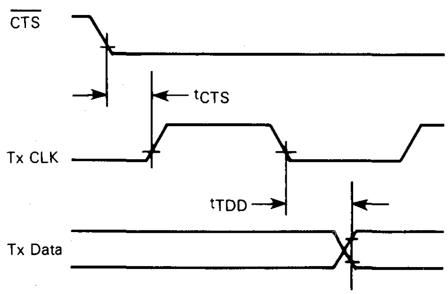
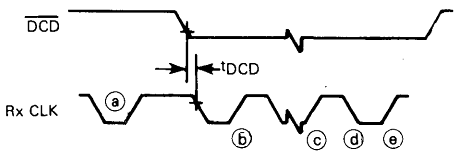


FIGURE 7 — DATA CARRIER DETECT SETUP TIME

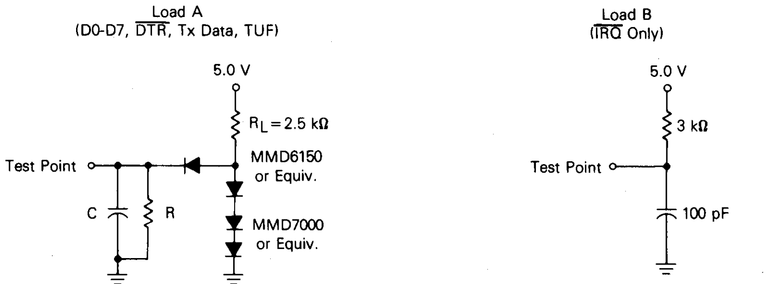


Notes:
a. Must occur before $\overline{DCD}$ goes low.
b. First data bit placed in Rx shift register.
c. Last data bit of byte placed in Rx shift register.
d. Rx data byte transferred from shift register to Rx FIFO.
e. Clock edge required for generation of $\overline{IRQ}$ by RDA status.

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



C = 130 pF for D0-D7
= 30 pF for $\overline{DTR}$, Tx Data, and TUF

R = 11.7 kΩ for D0-D7
= 24 kΩ for $\overline{DTR}$, Tx Data, and TUF

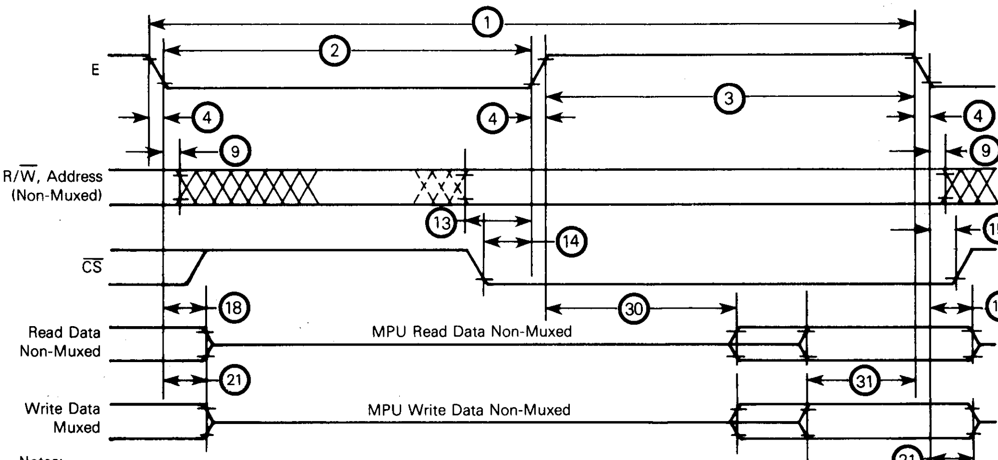
3

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Indent Number	Characteristic	Symbol	MC6852 Min	MC6852 Max	MC68A52 Min	MC68A52 Max	MC68B52 Min	MC68B52 Max	Unit
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

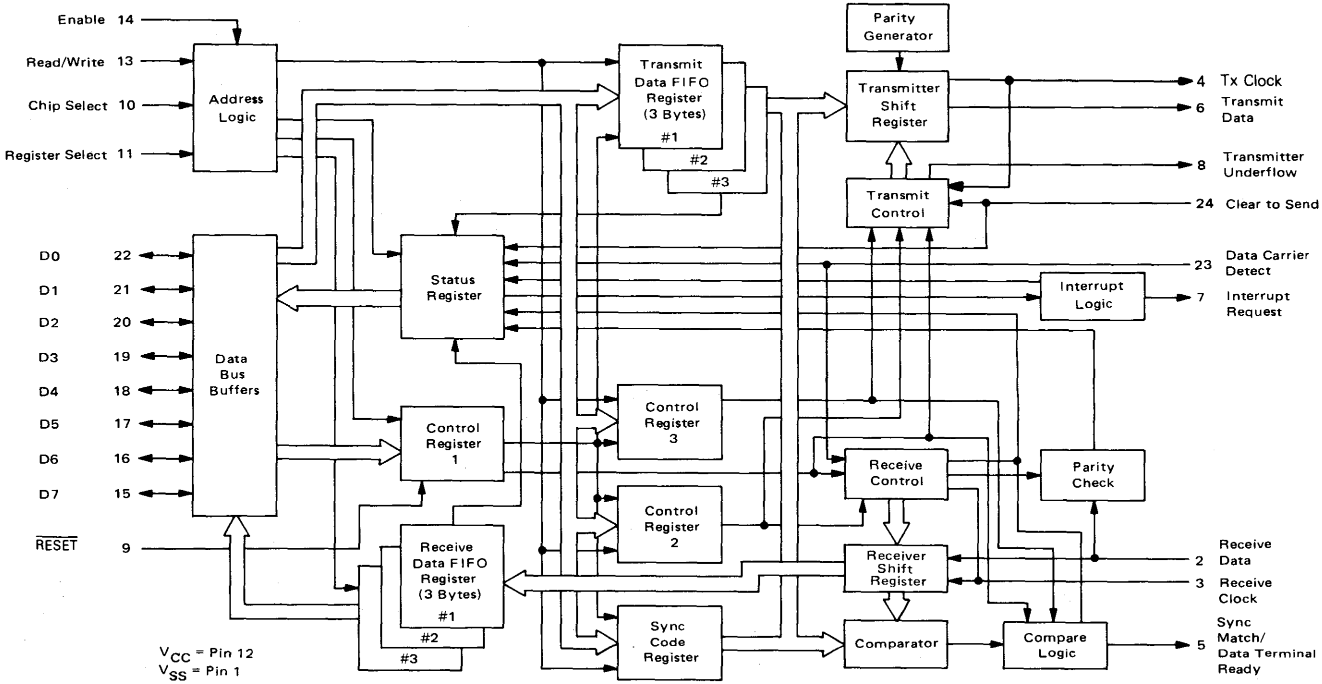
*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

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 $\overline{\text{DCD}}$ (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, $\overline{\text{CS}}$, Read/Write (R/$\overline{\text{W}}$) 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 ($\overline{\text{CTS}}$) and Data Carrier Detect ($\overline{\text{DCD}}$), 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 ($\overline{\text{IRQ}}$).

INITIALIZATION

During a power-on sequence, the SSDA is reset via the $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ 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 ($\overline{\text{CTS}}$) input provides for automatic control of the transmitter by means of external system hardware; e.g., the modem $\overline{\text{CTS}}$ output provides the control in a data communications system. The $\overline{\text{CTS}}$ input resets and inhibits the transmitter section when high, but does not reset the Transmit Data FIFO. The TDRA status bit is inhibited by $\overline{\text{CTS}}$ being high in either the one-sync character or two-sync character mode of operation. In the external sync mode, TDRA is unaffected by $\overline{\text{CTS}}$ in order to provide Transmit Data FIFO status for preloading and operating the transmitter under the control of the $\overline{\text{CTS}}$ 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.)

SYNCHRONIZATION

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 ($\overline{\text{DCD}}$) 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 ($\overline{\text{DCD}}$). 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 $\overline{\text{DCD}}$ input causes an interrupt if the EIE control bit has been set. The interrupt caused by $\overline{\text{DCD}}$ is cleared by reading the Status Register when the $\overline{\text{DCD}}$ status bit is high, followed by a Receive data FIFO read. The $\overline{\text{DCD}}$ status bit will subsequently follow the state of the $\overline{\text{DCD}}$ input when it goes low.

INPUT/OUTPUT FUNCTIONS

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/$\overline{W}$) – 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 ($\overline{CS}$) – This high-impedance TTL-compatible input line is used to address the SSDA. The SSDA is selected when $\overline{CS}$ is low. VMA should be used in generating the $\overline{CS}$ 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 ($\overline{IRQ}$) – 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.

$\overline{RESET}$ Input – The $\overline{RESET}$ input provides a means of resetting the SSDA from an external source. In the low state, the $\overline{RESET}$ 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/$\overline{DTR}$ 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 $\overline{RESET}$ 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 $\overline{RESET}$ (Rx Rs, Tx Rs, PC1, PC2, EIE, and E/I Sync) cannot be changed when $\overline{RESET}$ 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 ($\overline{CTS}$) – The $\overline{CTS}$ input provides a real-time inhibit to the transmitter section (the Tx Data FIFO is not disturbed). A positive $\overline{CTS}$ 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 $\overline{CTS}$ input in the external sync mode.

The positive transition of $\overline{CTS}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{CTS}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by writing a "1" in the Clear $\overline{CTS}$ bit in Control Register 3 or in the Transmitter Reset bit. The $\overline{CTS}$ status bit subsequently follows the $\overline{CTS}$ input when it goes low.

The $\overline{CTS}$ 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 $\overline{CTS}$ (see Figure 6).

Data Carrier Detect ($\overline{DCD}$) – The $\overline{DCD}$ input provides a real-time inhbit to the receiver section (the Rx FIFO is not disturbed). A positive $\overline{DCD}$ transition resets and inhibts the receiver section except for the Receive FIFO and the RDRA status bit and its associated $\overline{IRQ}$.

The positive transition of $\overline{DCD}$ is stored within the SSDA to insure that its occurrence will be acknowledged by the system. The stored $\overline{DCD}$ information and its associated $\overline{IRQ}$ (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 $\overline{DCD}$ status bit subsequently follows the $\overline{DCD}$ input when it goes low. The $\overline{DCD}$ 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 $\overline{DCD}$ (see Figure 7).

Sync Match/Data Terminal Ready (SM/$\overline{DTR}$) – 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 $\overline{DTR}$ mode (PC1="0") provides an output level corresponding to the complement of PC2 ($\overline{DTR}$="0" when PC2="1"). (See Table 1.)

TABLE 1 — SSDA PROGRAMMING MODEL

Register	Control Inputs RS	Control Inputs R/W	Address Control AC2	Address Control AC1	Register Content 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)	Clear-to-Send ($\overline{CTS}$)	Data Carrier Detect ($\overline{DCD}$)	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 $\overline{CTS}$ Status (Clear $\overline{CTS}$)	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).
$\overline{TUF}$	Bit 4	A "1" into CTUF (C3 Bit 3) or into Tx Rs (C1 Bit 1).
$\overline{CTS}$	Bit 3	A "1" into Clear $\overline{CTS}$ (C3 Bit 2) or a "1" into Tx Rs (C1 Bit1)
$\overline{DCD}$	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 $\overline{CTS}$	Bit 2	When "1", clears $\overline{CTS}$ (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, $\overline{CTS}$, and $\overline{DCD}$ 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/$\overline{DTR}$ Output Control

Bit 1 PC2	Bit 0 PC1	SM/$\overline{DTR}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{IRQ}$ status bit will go high and the $\overline{IRQ}$ 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. $\overline{DCD}$ 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 $\overline{CTS}$ input has gone to a "1". The interrupt is cleared by writing a "1" in the Clear $\overline{CTS}$ 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 $\overline{IRQ}$ status bit and the $\overline{IRQ}$ output are disabled for the above error conditions. A low level on the $\overline{RESET}$ 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/$\overline{W}$="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 $\overline{DCD}$ input or by starting Rx CLK at the midpoint of data bit 0 of a cahracter with $\overline{DCD}$ 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 $\overline{RESET}$ 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 $\overline{CTS}$ Status (Clear $\overline{CTS}$), C3 Bit 2 – When a "1" is written into the Clear $\overline{CTS}$ bit, the stored status and interrupt are cleared. Subsequently, the $\overline{CTS}$ status bit reflects the state of the $\overline{CTS}$ input. The Clear $\overline{CTS}$ control bit does not affect the $\overline{CTS}$ input nor its inhibit of the transmitter section. The Clear $\overline{CTS}$ 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/$\overline{W}$="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/$\overline{DTR}$ 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.

3

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 ($\overline{CTS}$), S Bit 3 — A positive transition on the $\overline{CTS}$ input is stored in the SSDA until cleared by writing a "1" into the Clear $\overline{CTS}$ control bit or the Tx Rs bit. The $\overline{CTS}$ status bit, when set, indicates that the $\overline{CTS}$ input has gone high. The Clear $\overline{CTS}$ command (a "1" into C3 Bit 2) allows Bit 3 of subsequent Status reads to indicate the state of the $\overline{CTS}$ 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 $\overline{DCD}$ input does not clear the Parity Error or Rx Data FIFO status bits.

Interrupt Request ($\overline{IRQ}$), S Bit 7 — The Interrupt Request status bit indicates when the $\overline{IRQ}$ output is in the active state ($\overline{IRQ}$ output = "0"). The $\overline{IRQ}$ status bit is subject to the same interrupt enables (RIE, TIE, and EIE) as the $\overline{IRQ}$ output. The $\overline{IRQ}$ status bit simplifies status inquiries for polling systems by providing single bit indication of service requests.

MOTOROLA

MC68HC53

Product Preview

HCMOS

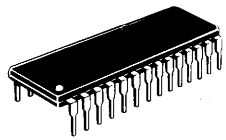
(HIGH DENSITY CMOS SILICON-GATE)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC68HC53 ACIA provides a program-controlled interface between 8-bit, microprocessor-based systems, serial communication data sets, and modems. An on-chip crystal oscillator and a baud-rate generator allow the ACIA to transmit at 15 different program-selected rates, ranging from 50 to 19,200 baud. The MC68HC53 can receive at either the transmit rate or at 16 times an external clock rate. A MOTEL (MOTorola - IntEL) bus compatible circuit, is incorporated in the MC68HC53. This circuit allows the device to directly interface with many types of microprocessors.

- Compatible With 8-Bit Microprocessors
- Full-Duplex or Half-Duplex Operation With Buffered Receiver and Transmitter
- Fifteen Programmable Baud Rates (50 to 19,200)
- Receiver Data Rate May Be Identical to Baud Rate or May Be 16 Times the External Clock Input
- Data Set/Modem Control Functions
- Programmable Word Lengths, Number of Stop Bits, and Parity Bit Generation and Detection
- Programmable Interrupt Control
- Software Reset
- Program-Selectable Serial Echo Mode
- Two Chip Selects
- 2 MHz or 1 MHz Clock Rate
- Single +5 Volt ±5% Power Supply
- Full TTL Compatibility
- MOTEL Read/Write Control Circuit

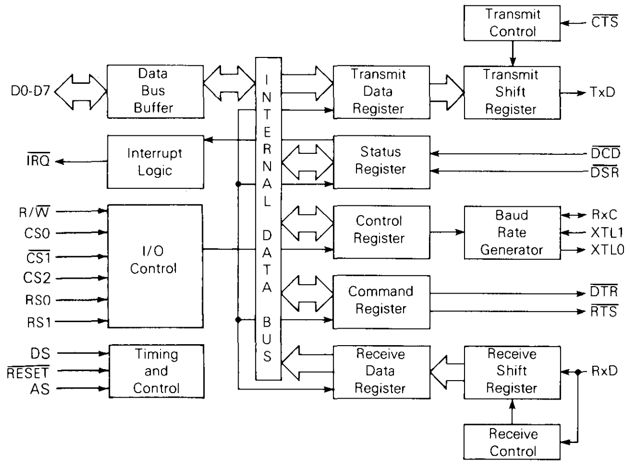


P SUFFIX
PLASTIC PACKAGE
CASE 710

PIN ASSIGNMENT

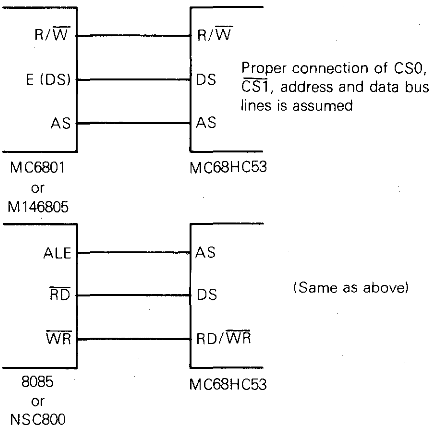
Signal	Pin	Pin	Signal
V_{SS}	1	28	R/$\overline{W}$
CS0	2	27	DS
$\overline{CS1}$	3	26	$\overline{IRQ}$
$\overline{RESET}$	4	25	A/D7
RxC	5	24	A/D6
XTL1	6	23	A/D5
XTL0	7	22	A/D4
$\overline{RTS}$	8	21	A/D3
$\overline{CTS}$	9	20	A/D2
TxD	10	19	A/D1/RS1
$\overline{DTR}$	11	18	A/D0/RS0
RxD	12	17	$\overline{DSR}$
CS2	13	16	$\overline{DCD}$
AS	14	15	V_{CC}

FIGURE 1 — BLOCK DIAGRAM



This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

FIGURE 2 — INTERFACE REQUIREMENTS DIAGRAM



SIGNAL DESCRIPTIONS

The following paragraphs provide a brief description of the input and output signals for the MC68HC53.

RESET ($\overline{RESET}$)

During system initialization, a low on the $\overline{RESET}$ input clears the internal registers.

ADDRESS STROBE (AS)

Address strobe indicates the presence of an address on the multiplexed bus. The negative edge latches address/data lines 0-1 and chip select 2.

DATA STROBE (DS)

This input is used to transfer data to or from the microprocessor.

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

The R/$\overline{W}$ is generated by the microprocessor and is used to control the direction of data transfers. A high on the R/$\overline{W}$ pin allows the processor to read the data supplied by the ACIA. A low on the R/$\overline{W}$ pin allows a write to the ACIA.

INTERRUPT REQUEST ($\overline{IRQ}$)

The $\overline{IRQ}$ pin is an interrupt output from the interrupt control logic. It permits several devices to be connected to the common $\overline{IRQ}$ microprocessor input. Normally a high level, $\overline{IRQ}$ goes low when an interrupt occurs.

ADDRESS/DATA BUS (A/D0-A/D7)

The A/D0-A/D7 pins are the eight data lines used to transfer data and addresses. These lines are bidirectional and are normally in the high-impedance state, except during read cycles when the ACIA is selected. D0 and D1 are dual-purpose register selects and data lines. They are demultiplexed by AS as follows:

D1/RS1	D0/RS0	Write	Read
0	0	Transmit Data Register	Received Data Register
0	0	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

DATA SET READY ($\overline{DSR}$)

The $\overline{DSR}$ input pin is used to indicate to the ACIA the status of the modem. A low indicates the "ready" state and a high "not-ready". $\overline{DSR}$ is a high-impedance input, and must be connected. If unused, it should be driven high or low but not switched.

DATA CARRIER DETECT ($\overline{DCD}$)

The $\overline{DCD}$ input pin is used to indicate to the ACIA the status of the carrier-detect output of the modem. A low indicates that the modem carrier signal is present and a high that it is not. Like $\overline{DSR}$, $\overline{DCD}$ is a high-impedance input and must be connected.

REQUEST TO SEND ($\overline{RTS}$)

The $\overline{RTS}$ output pin is used to control the modem from the processor. The state of the $\overline{RTS}$ pin is determined by the contents of the command register.

CLEAR TO SEND ($\overline{CTS}$)

The $\overline{CTS}$ input pin is used to control the transmitter operation. The enable state is with $\overline{CTS}$ low. The transmitter is automatically disabled if $\overline{CTS}$ is high.

DATA TERMINAL READY ($\overline{DTR}$)

This output pin is used to indicate the status of the ACIA to the modem. A low on $\overline{DTR}$ indicates the ACIA is enabled and a high indicates it is disabled. The processor controls this pin via bit 0 of the command register.

CHIP SELECTS 0, 1, AND 2 (CS0, $\overline{CS1}$, AND CS2)

These three chip-select inputs are normally connected to the processor address lines either directly or through decoders. The ACIA is selected when CS0 is high, $\overline{CS1}$ is low, and CS2 is high. CS2 is latched by AS.

CRYSTAL PINS (XTL1, XTL0)

These pins are normally directly connnected to the external crystal (1.8432 megahertz) used to derive the various baud rates. Alternatively, an externally generated clock may be used to drive the XTL1 pin in which case the XTL0 pin must float. XTL1 is the input pin for the transmit clock.

TRANSMIT DATA (TxD)

The TxD output line is used to transfer serial non-return-to-zero (NRZ) data to the modem. The least significant bit

(LSB) of the transmit data register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected, or under control of an external clock (as selected by the control register).

RECEIVE DATA (RxD)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or the rate of an externally generated receiver clock (as selected by the control register).

RECEIVE CLOCK (RxC)

The RxC is a bidirectional pin which serves as either the receiver 16x clock input or the receiver 16x clock output. The latter mode results if the internal baud-rate generator is selected for receiver data clocking.

MOTEL CIRCUIT

The MOTEL circuit is a new concept that permits the MC68HC53 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.

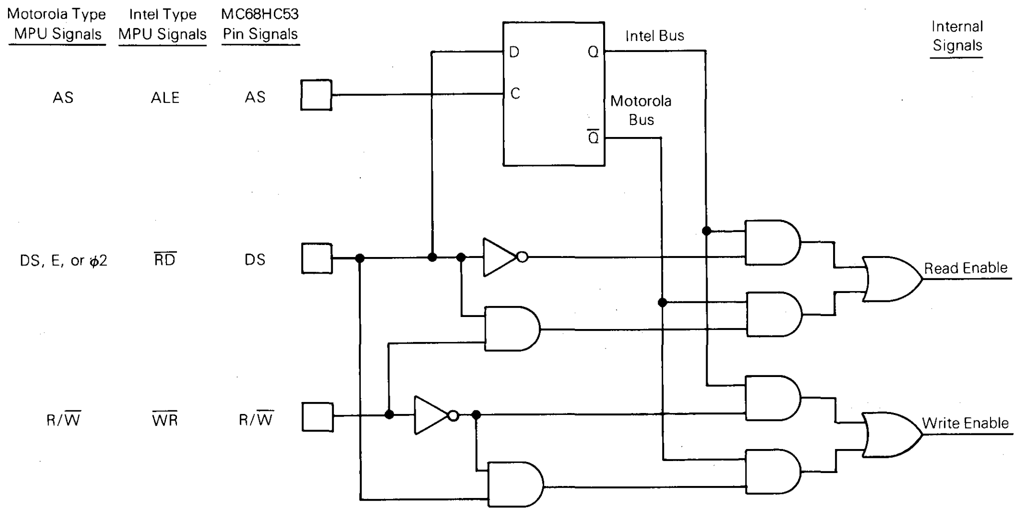
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated for the Motorola MC6800 and the other for the Intel 8080 and its companion part, the 8228.

The MOTEL circuit (for MOTorola and intEL bus compatibility) is built into a peripheral or memory IC 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 3.

MOTEL selects one of the two interpretations of two pins. In the Motorola case, DS and R/$\overline{\text{W}}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of R/$\overline{\text{W}}$. With competitor buses, the inversion of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ create functionally identical internal read and write enable signals.

The MC68HC53 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{\text{RD}}$ pin. Since DS is always low and $\overline{\text{RD}}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

FIGURE 3 — MOTEL CIRCUIT-LOGIC DIAGRAM



MAIN DATA/CONTROL REGISTERS

A brief description of the main MC68HC53 data and control registers follows.

TRANSMIT DATA REGISTER

This 8-bit register provides temporary storage for the data to be transmitted. Bit 0 is the leading bit to be transmitted. Unused bits are the high-order bits and are "don't care" for transmission.

RECEIVE DATA REGISTER

This 8-bit register provides temporary storage for the data being received. Bit 0 is the leading bit received. Unused bits are the high-order bits and are "zeros" for the receiver. Parity bits are not contained in the receive data register but are stripped off after being used for parity checking. Thus, former parity bits become unused "zero" bits in the receive data register.

COMMAND REGISTER

This 8-bit register contains the command word received from the controlling microprocessor. The command word specifies the specific modes and functions the MC68HC53 is to assume. Included are data terminal ready, transmitter interrupt disabled, receiver echo mode, and parity disabled.

CONTROL REGISTER

This 8-bit register contains, message format information received from the microprocessor, and includes: baud rate, clock source, word length, and number of stop bits. This information is used by the MC68HC53 for synchronization and proper processing of message data.

STATUS REGISTER

This 8-bit register contains the current status of the MC68HC53 and the related modem. This register is continuously accessed by the controlling microprocessor during operation to determine if data processing is being performed properly or if errors have occurred. Status indications include: parity error, framing error, overrun, clear to send, transmit register empty, receive register full, data carrier detect, and interrupt request.

MOTOROLA

MC6854

ADVANCED DATA-LINK CONTROLLER (ADLC)

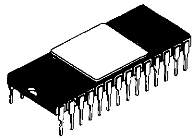
The MC6854 ADLC performs the complex MPU/data communication link function for the "Advanced Data Communication Control Procedure" (ADCCP), High-Level Data-Link Control (HDLC) and Synchronous Data-Link Control (SDLC) standards. The ADLC provides key interface requirements with improved software efficiency. 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

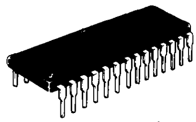
MOS

(N-CHANNEL, SILICON GATE)

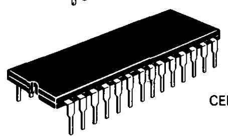
ADVANCED DATA-LINK CONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710

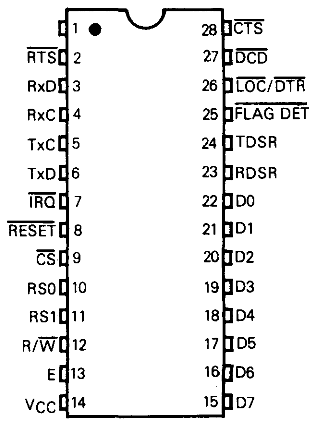


S SUFFIX
CERDIP PACKAGE
CASE 733

ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic L Suffix	1.0	0°C to 70°C	MC6854L
	1.0	−40°C to 85°C	MC6854CL
	1.5	0°C to 70°C	MC68A54L
	1.5	−40°C to 85°C	MC68A54CL
	2.0	0°C to 70°C	MC68B54L
Cerdip S Suffix	1.0	0°C to 70°C	MC6854S
	1.0	−40°C to 85°C	MC6854CS
	1.5	0°C to 70°C	MC68A54S
	1.5	−40°C to 85°C	MC68A54CS
	2.0	0°C to 70°C	MC68B54S
Plastic P Suffix	1.0	0°C to 70°C	MC6854P
	1.0	−40°C to 85°C	MC6854CP
	1.5	0°C to 70°C	MC68A54P
	1.5	−40°C to 85°C	MC68A54CP
	2.0	0°C to 70°C	MC68B54P

PIN ASSIGNMENT



MAXIMUM RATINGS

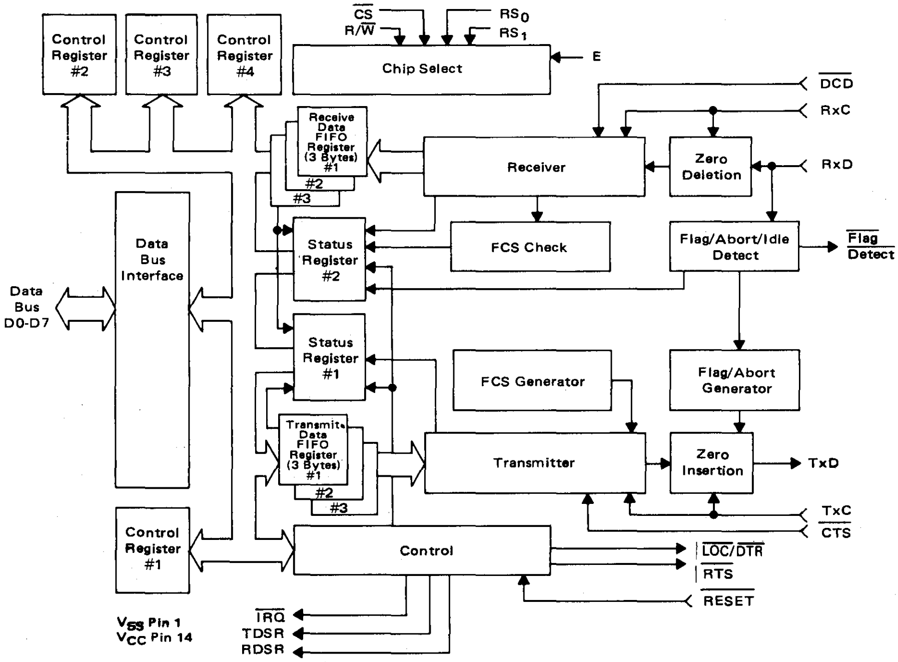
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6854, MC68A54, MC68B54 MC6854C, MC68A54C	T_A	(T_L to T_H) 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

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 CHRACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Ceramic Cerdip	θ_{JA}	 115 60 65	°C/W

FIGURE 1 — ADLC GENERAL BLOCK DIAGRAM



POWER CONSIDERATIONS

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

$T_J = T_A + (P_D \bullet \theta_{JA})$ (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, $T_A = T_L$ 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
Hi-Z (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)	D0-D7	V_{OH}	V_{SS}+2.4	–	–	V
(V_{Load} = −100 μA)	All Others		V_{SS}+2.4	–	–	
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 = 0°C)		P_{INT}	–	–	850*	mW
Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	D0-D7	C_{in}	–	–	12.5	pF
	All Other Inputs		–	–	7.5	
	$\overline{IRQ}$	C_{out}	–	–	5.0	pF
	All Others		–	–	10	

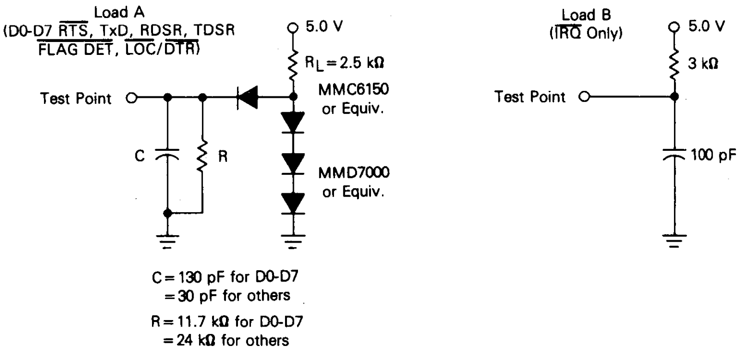
*For temperatures below 0°C, P_{INT} will increase.

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



3

FIGURE 3 — RECEIVER DATA SETUP/HOLD, FLAG DETECT AND LOOP ON-LINE CONTROL DELAY TIMING

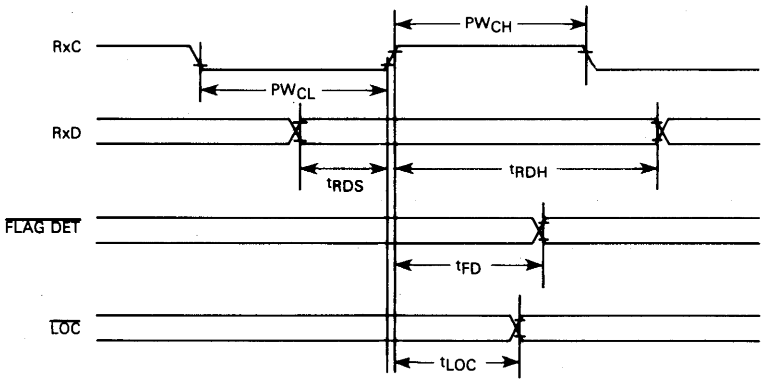
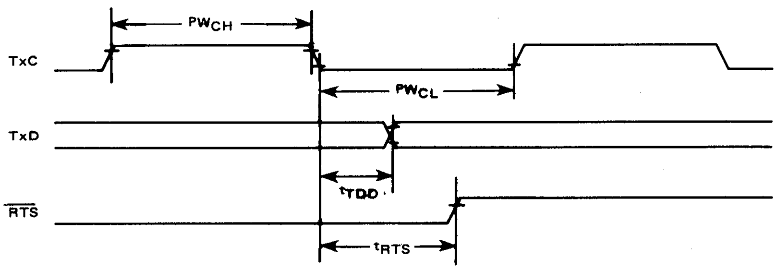
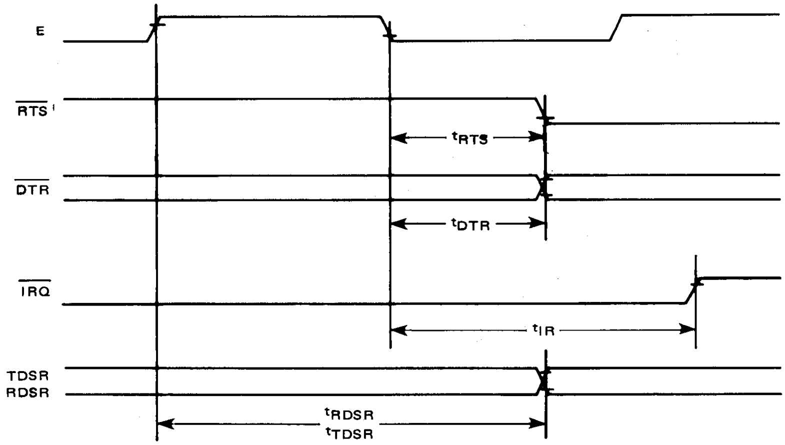


FIGURE 4 — TRANSMIT DATA OUTPUT DELAY AND REQUEST-TO-SEND DELAY TIMING



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 — TDSR/RDSR DELAYS, $\overline{IRQ}$ RELEASE DELAY, $\overline{RTS}$ AND $\overline{DTR}$ DELAY TIMING



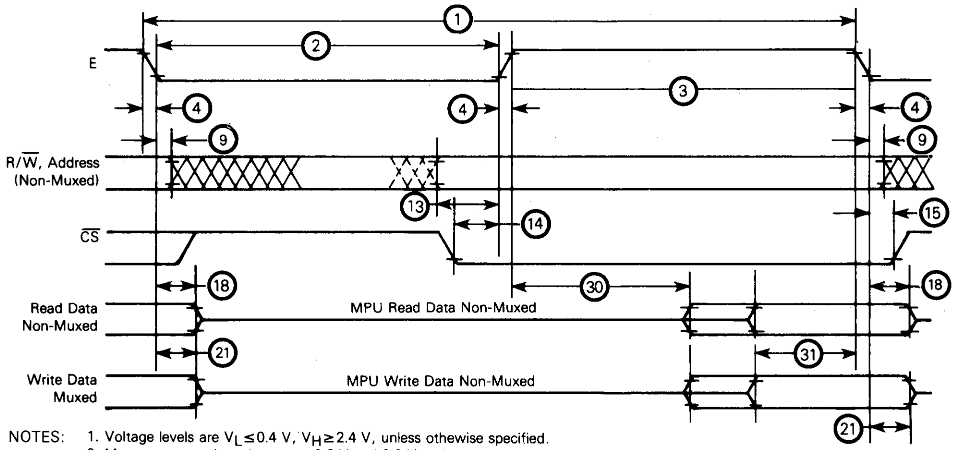
NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

BUS TIMING CHARACTERISTICS (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC6854 Min	MC6854 Max	MC68A54 Min	MC68A54 Max	MC68B54 Min	MC68B54 Max	Unit
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

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR}max (High Impedance).

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

← A Frame →

01111110	8 Bits Per Byte	8 Bits Per Byte	8 Bits Per Byte	Variable Length (5-8)	16 Bit	01111110
(Opening) Flag	Address* Field	Control* Field	Logical Control Sub-Field (Option)		Frame Check Sequence Field	(Closing) Flag

*Extendable (Optional)

← Information Field (Optional) →

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.

3

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.

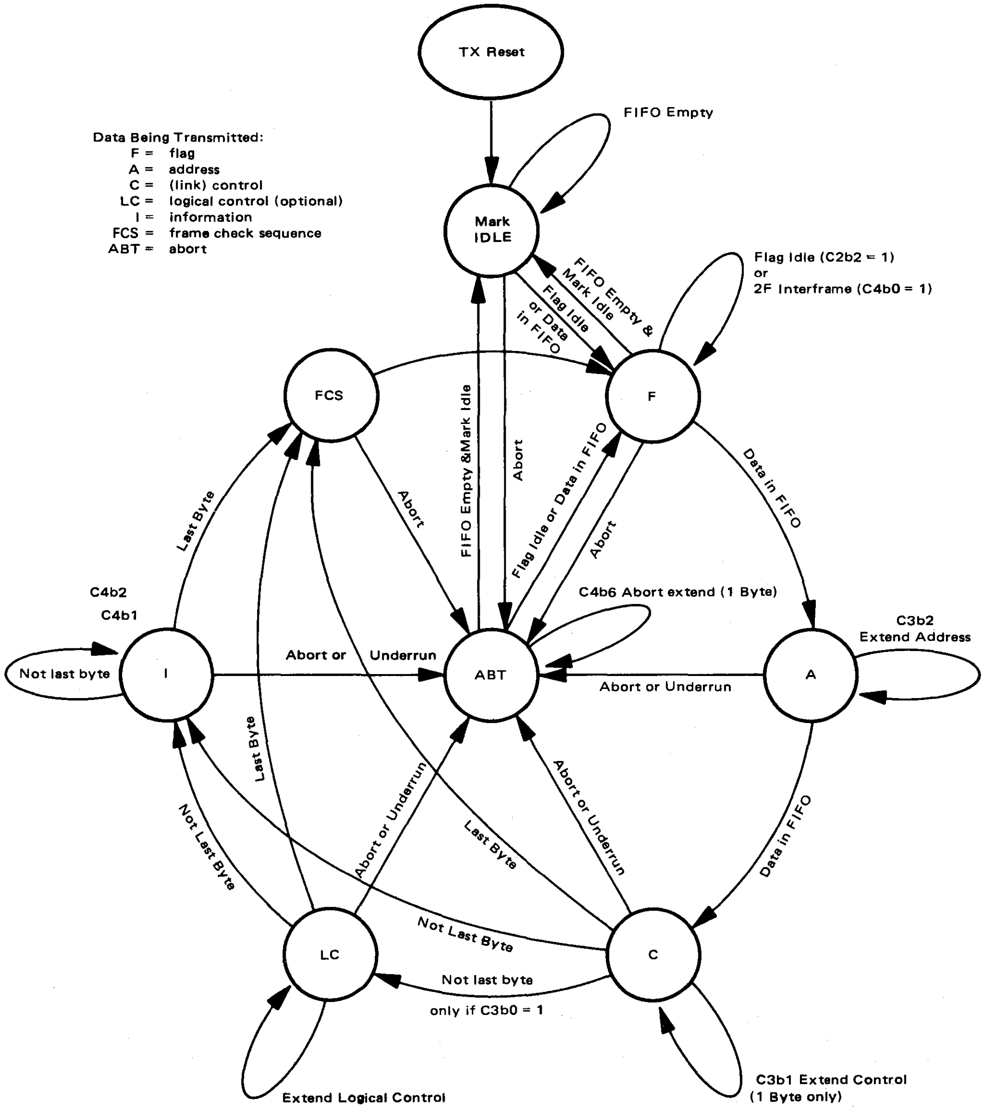
TRANSMITTER OPERATION

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/$\overline{\text{1-Byte}}$ 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.

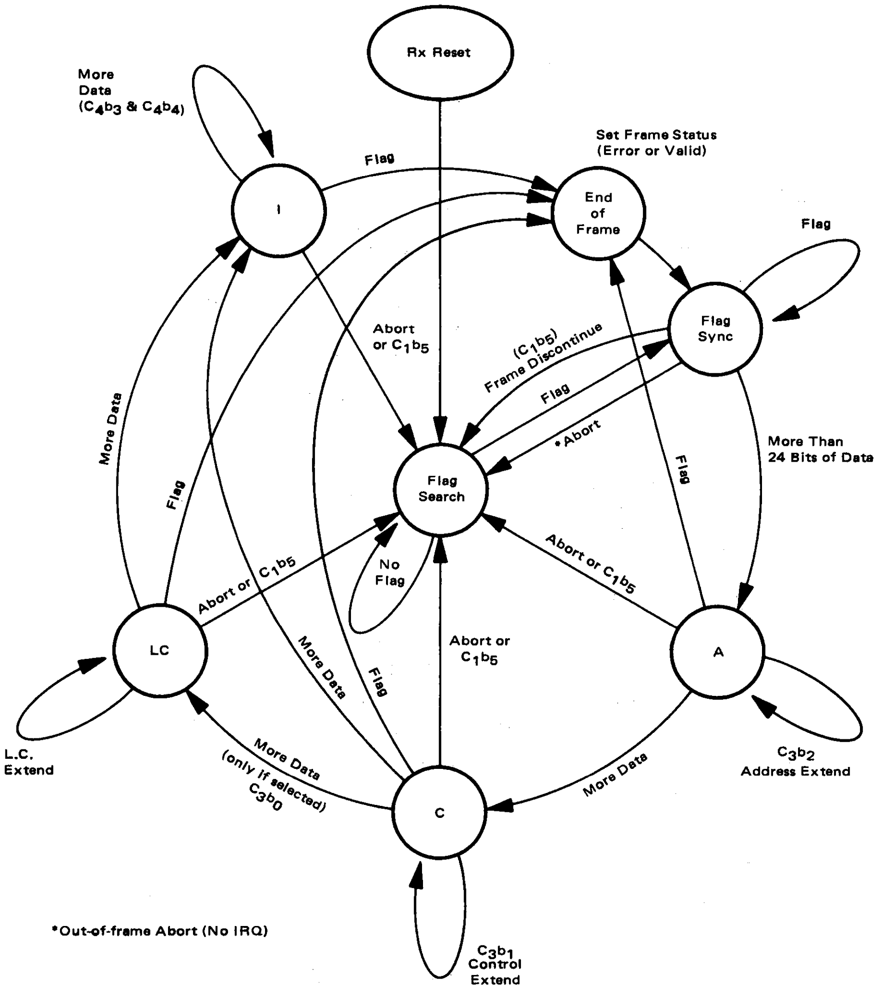
The first byte (Address field) should be written into the Tx FIFO at the "Frame Continue" address. Then the transmission of a frame automatically starts. If the transmitter is in a mark idle state, the transfer of an address causes an opening flag within two or three transmitter clock cycles. If the 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)



3

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

3

FIGURE 9a — TYPICAL LOOP CONFIGURATION

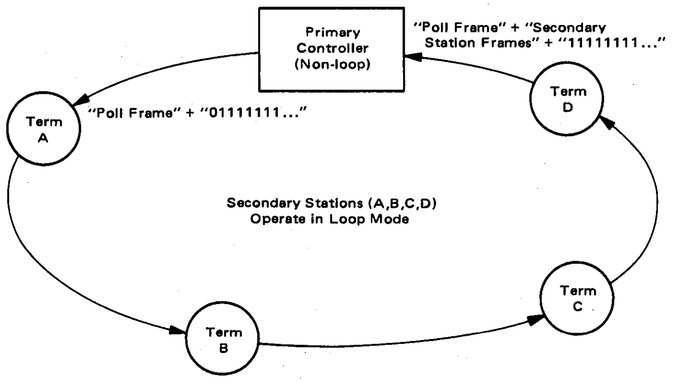
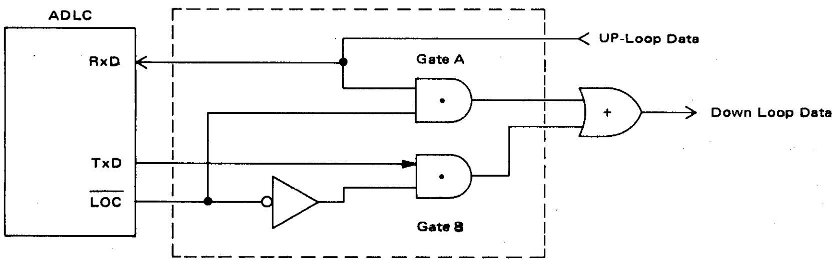


FIGURE 9b — EXAMPLE OF EXTERNAL LOOP LOGIC



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 $\overline{LOC}/\overline{DTR}$ 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 $\overline{LOC}/\overline{DTR}$ 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 $\overline{FD}$ output to go low. IRQ is generated if 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 $\overline{RESET}$ 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 $\overline{LOC}/\overline{DTR}$ 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 ($\overline{IRQ}$), 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 ($\overline{CS}$, RS0, and RS1) and R/$\overline{W}$ 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 ($\overline{CS}$) — An ADLC read or write operation is enabled only when the $\overline{CS}$ input is low and the E clock input is high. (E•$\overline{CS}$).

Register Selects (RS0, RS1) — When the Register Select inputs are enabled by (E•$\overline{CS}$), 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/$\overline{W}$) — The R/$\overline{W}$ input controls the direction of data flow on the data bus when it is enabled by (E•$\overline{CS}$). When R/$\overline{W}$ 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 ($\overline{RESET}$) — The $\overline{RESET}$ input provides a means of resetting the ADLC from a hardware source. In the "low state," the $\overline{RESET}$ 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, $\overline{RTS}$, Loop Mode, and Loop On-Line/$\overline{DTR}$.
- Clears all stored status condition of the status registers.
- Outputs: $\overline{RTS}$ and $\overline{LOC/DTR}$ 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 $\overline{RESET}$ cannot be changed when $\overline{RESET}$ is "low."

Interrupt Request Output ($\overline{IRQ}$) — $\overline{IRQ}$ 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. $\overline{IRQ}$ will be low as long as the $\overline{IRQ}$ status bit is set and is high if the $\overline{IRQ}$ 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 ($\overline{RTS}$) — 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 $\overline{RTS}$ bit goes high, the $\overline{RTS}$ output is forced low. When the $\overline{RTS}$ bit returns low, the $\overline{RTS}$ 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 $\overline{RTS}$ occurs after the completion of a Flag, an Abort, or when the RTS control bit is reset during a mark idling state. When the $\overline{RESET}$ input is low, the $\overline{RTS}$ output goes high.

Clear-to-Send Input ($\overline{CTS}$) — The $\overline{CTS}$ input provides a real-time inhibit to the TDRA status bit and its associated interrupt. The positive transition of $\overline{CTS}$ is stored within the ADLC to ensure its occurrence will be acknowledged by the system. The stored $\overline{CTS}$ information and its associated $\overline{IRQ}$ (if enabled) are cleared by writing a "1" in the Clear Tx Status bit or in the Transmitter Reset bit.

Data-Carrier-Detect Inupt ($\overline{DCD}$) — The $\overline{DCD}$ input provides a real-time inhibit to the receiver section. A high level on the $\overline{DCD}$ input resets and inhibits the receiver register, but data in the Rx FIFO from a previous frame is not disturbed. The positive transition of $\overline{DCD}$ is stored within the ADLC to ensure that its occurrence will be acknowledged by the system. The stored $\overline{DCD}$ information and its associated $\overline{IRQ}$ (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 ($\overline{LOC/DTR}$) — The $\overline{LOC/DTR}$ output serves as a $\overline{DTR}$ output in the non-loop mode or as a Loop Control output in the loop mode. When $\overline{LOC/DTR}$ output performs the $\overline{DTR}$ function, it is turned on and off by means of the $\overline{LOC/DTR}$ control bit. When the $\overline{LOC/DTR}$ control bit is high the $\overline{DTR}$ output will be low. In the loop mode the $\overline{LOC/DTR}$ output provides the means of controlling the external loop interface hardware to go On-line or Off-line. When the $\overline{LOC/DTR}$ control bit is SET and the loop has "idled" for 7 bit times or more (RxD=01111111...), the $\overline{LOC/DTR}$ output will go low (online). The $\overline{RESET}$ input being low will cause the $\overline{LOC/DTR}$ output to be high.

Flag Detect Output ($\overline{FD}$) — An output to indicate the reception of a flag and initiate an external time-out counter for the loop mode operation. The $\overline{FD}$ 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.

3

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. $\overline{\text{TDSR}}$ 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	$R/\overline{W}$	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

3

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

	Bit #	RS1 RS0 = 00 Status Register #1	RS1 RS0 = 01 Status Register #2	RS1 RS0 = 10 Receiver Data Register	RS1 RS0 = 11
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	$\overline{CTS}$	FCS Error	Bit 4	
	5	Tx Underrun	$\overline{DCD}$	Bit 5	
	6	TDRA/Frame Complete	Rx Overrun	Bit 6	
	7	$\overline{IRQ}$ 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)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	0	0	X	TxRS	RxRS	Discontinue	TDSR Mode	RDSR Mode	TIE	RIE	AC

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, àn interrupt request caused by RDA status is inhibited, and the ADLC does not request data transfer via the $\overline{IRQ}$ 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 $\overline{IRQ}$ 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 $\overline{DCD}$ input.) Rx RS is set by forcing a low level on the $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{CTS}$ input. Tx RS is set by forcing a low level on the $\overline{RESET}$ input or by writing a "1" from the data bus. It must be reset by writing a "0" after $\overline{RESET}$ has gone high.

CONTROL REGISTER 2 (CR2)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	1	0	0	RTS	CLR TxST	CLR RxST	Tx Last	FC/TDRA Select	F/M Idle	2/1 Byte	PSE

3

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 $\overline{CTS}$ 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 $\overline{RTS}$ output to be low (the active state). When the RTS bit returns low and data is being transmitted, the $\overline{RTS}$ 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 $\overline{RTS}$ output will go high (the inactive state) within two bit times.

CONTROL REGISTER 3 (CR3)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
0	1	0	1	LOC/ DTR	GAP/ TST	Loop	FDSE	01/11 Idle	AEX	CEX	LCF

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)

RS1	RS0	R/$\overline{W}$	AC	7	6	5	4	3	2	1	0
				NRZI/NRZ	ABT_{EX}	ABT	Rx		Tx		"FF"/F
1	1	0	1				WLS_2	WLS_1	WLS_2	WLS_1	

3

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

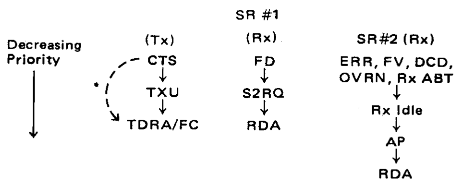
The Status Register #1 is the main status register. The IRQ bit indicates whether the ADLC requests service or not. The S2RQ bit indicates whether any bits in status register #2 request any service. TDRA and RDA, because they are most often used, are located in bit positions that are more convenient 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								

3

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 — 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 $\overline{\text{RTS}}$ Release** — In many cases, a MODEM will require a delay for releasing $\overline{\text{RTS}}$. An 8-bit or 16-bit delay can be added to the ADLC $\overline{\text{RTS}}$ 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 $\overline{\text{RTS}}$ control bit. The transmitter will transmit eight or sixteen "1's" and the $\overline{\text{RTS}}$ 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 ($\overline{\text{CTS}}$)** — The $\overline{\text{CTS}}$ 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, $\overline{\text{CTS}}$ also inhibts the TDSR DMA request. The $\overline{\text{CTS}}$ 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.

MOTOROLA

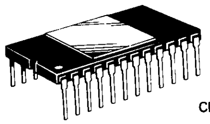
MC6859

Advance Information

MOS

DEPLETION LOAD
(N-CHANNEL, SILICON-GATE)

DATA SECURITY DEVICE



L SUFFIX
CERAMIC PACKAGE
CASE 716

DATA SECURITY DEVICE

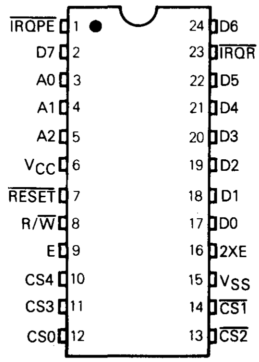
The MC6859 Data Security Device (DSD) is a monolithic MOS integrated circuit designed to be integrated into a wide range of equipment requiring protection of data by the employment of cryptographic measures.

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).

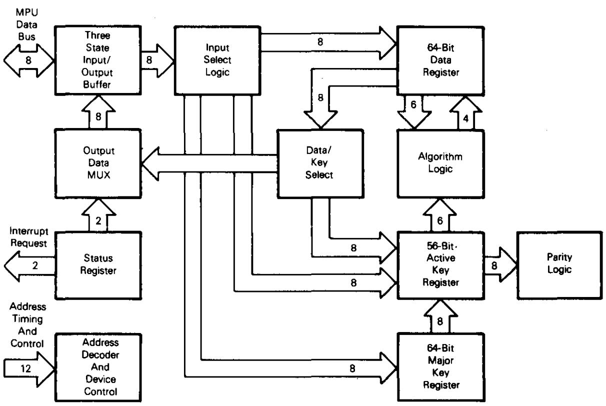
Through the use of flexible on-chip control and status circuitry and external control lines, the DSD provides direct capability of adapting the functional implementation of the DES algorithm for various specific system requirements for data protection.

- 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

PIN ASSIGNMENT



DATA SECURITY DEVICE BLOCK DIAGRAM



NOTICE
This product may not be exported without prior approval from the U.S. Department of State, Office of Munitions Control.

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

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 MC6859	T_A	T_L to T_H 0 to 70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Package	θ_{JA}	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 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:

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

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
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 (I_{Load} = 1.6 mA) (I_{Load} = 3.2 mA) (See Figure 2)	 D0-D7 $\overline{IRQPE}$, $\overline{IRQR}$	V_{OL}	 – –	 – –	 V_{SS}+0.4 V_{SS}+0.6	V
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 = 0°C)		P_{INT}	–	–	1000	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
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{IRQPE}$, $\overline{IRQR}$	C_{out}	–	–	50	pF

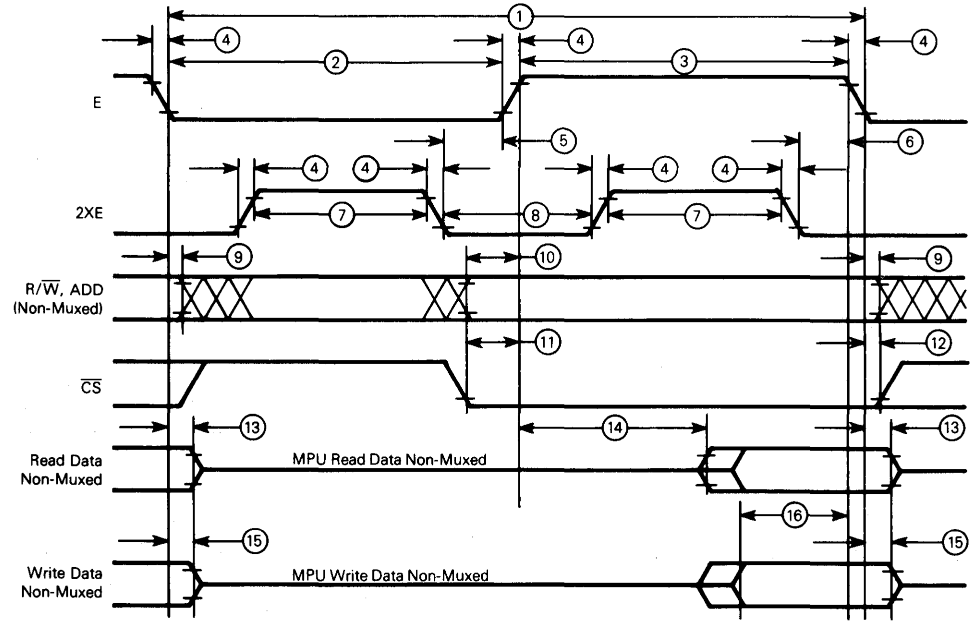
BUS TIMING CHARACTERISTICS

Ident. Number	Characteristic	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1.0	10	μs
2	Pulse Width, E Low	PW_{EL}	430	–	ns
3	Pulse Width, E High	PW_{EH}	450	–	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	ns
5	2XE to E High Delay Time	t_{DH}	0	–	ns
6	2XE to E Low Delay Time	t_{DL}	0	–	ns
7	Pulse Width 2XE Low	PW_{2L}	210	–	ns
8	Pulse Width 2XE High	PW_{2H}	220	–	ns
9	Address Hold Time	t_{AH}	10	–	ns
10	Address Setup Time Before E	t_{AS}	80	–	ns
11	Chip Select Setup Time Before E	t_{CS}	80	–	ns
12	Chip Select Hold Time	t_{CH}	10	–	ns
13	Read Data Hold Time	t_{DHR}	20	50*	ns
14	Output Data Delay Time	t_{DHW}	–	290	ns
15	Write Data Hold Time	t_{DDR}	10	–	ns
16	Input Data Setup Time**	t_{DSW}	165	–	ns
17	Interrupt Release Time	t_{IR}	1200	–	ns

*The data bus output buffers are no longer sourcing or sinking current by t_{DHR} maximum (high impedance).

**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.

FIGURE 1 — BUS TIMING



NOTES:
1. Voltage levels shown are $V_L \le 0.4$ 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.

FIGURE 2 — BUS TIMING TEST LOADS

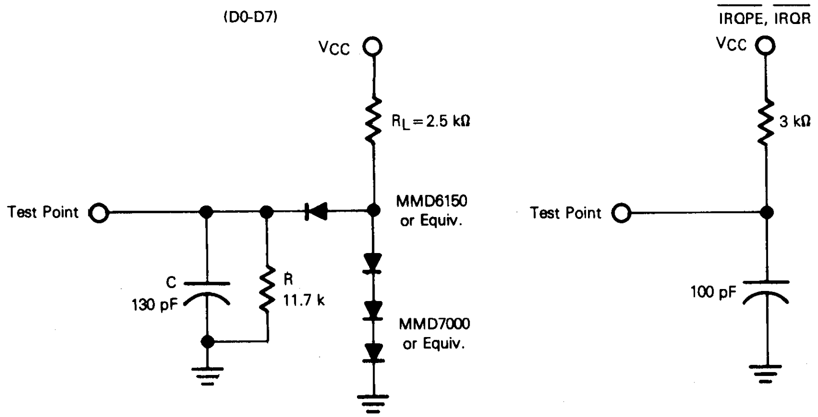
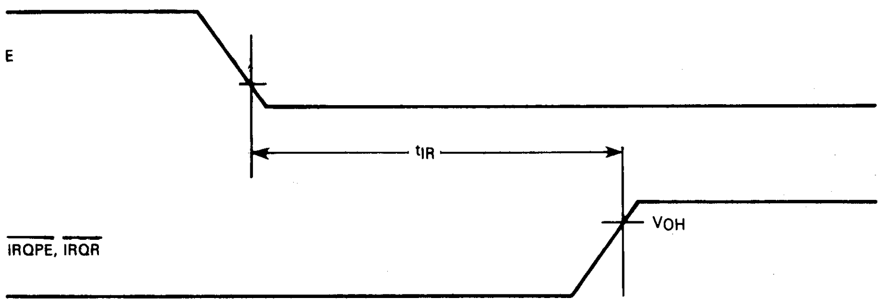


FIGURE 3 — INTERRUPT RELEASE TIME



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/$\overline{W}$) line, an external $\overline{RESET}$ line, three register select lines, an Enable (System ϕ2) 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, $\overline{CS1}$, $\overline{CS2}$, 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 $\overline{CS1}$ = $\overline{CS2}$ = 0, the device is selected.

Read/Write (R/$\overline{W}$) — With the DSD selected, this input controls the direction of data transfer on the data bus. When R/$\overline{W}$ is high, data in the DSD is read by the MPU on the trailing edge of E. A low state on the R/$\overline{W}$ line enables data transfer from the MPU on the trailing edge of the 2XE signal.

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 ($\overline{RESET}$) — 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 $\overline{RESET}$ 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 $\overline{RESET}$ 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/$\overline{W}$ 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/$\overline{W}$ 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 A0, A1, A2	R/$\overline{W}$	Operational Mode
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 A0, A1, A2	R/$\overline{W}$	Control Mode
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 ($\overline{IRQR}$) — This active low output signals the MPU that the DSD is ready to initiate another operation. The $\overline{IRQR}$ signal will be inactive during encryption/decryption or key transfer.

Parity Error Interrupt Request ($\overline{IRQPE}$) — This active low output is used to signal the MPU that the DSD has detected a parity error. The $\overline{IRQPE}$ 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.

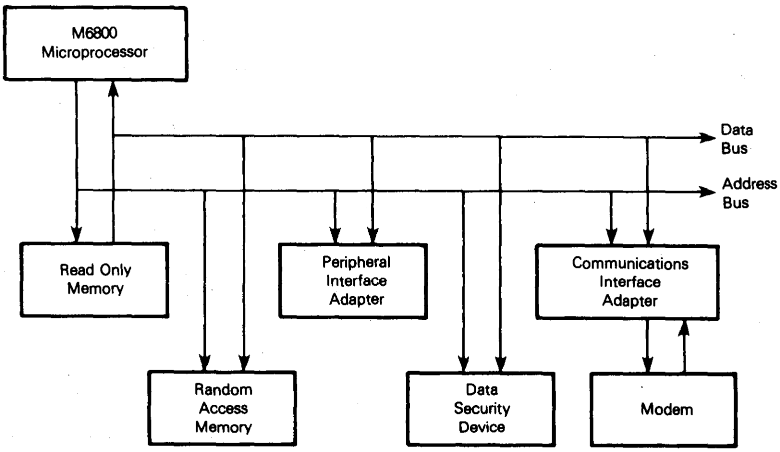
Read Data — This command is normally executed upon completion of the encipher/decipher algorithm (indicated by $\overline{\text{READY}}$ = 0). A read prior to completion of busy will result in all zeros being read from D0-D7. As each byte of data is read, zeros are automatically shifted into the data 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.

3

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)

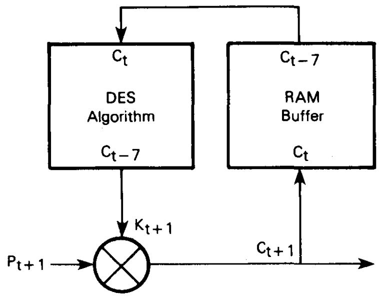
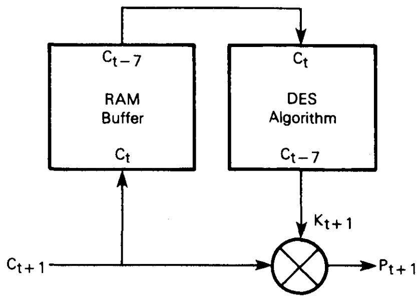


FIGURE 6 — CFB ENCIPHER DATA FLOW (RECEIVING)



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

3

ORDERING INFORMATION

MC68A59CL

- 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
  - 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	MC6859L	0 to 70°C

MOTOROLA

MC6875
MC6875A

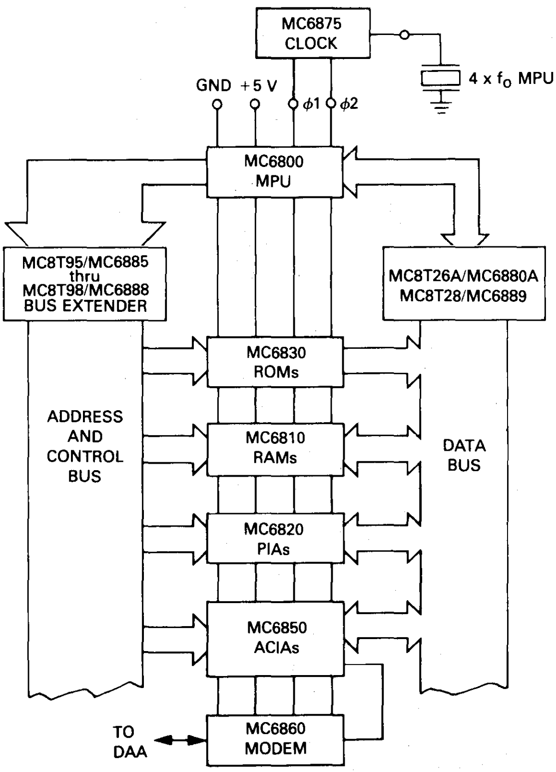
Specifications and Applications Information

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.

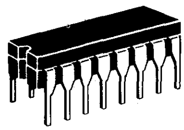
Schottky technology is employed for high speed and PNP-buffered inputs are employed for NMOS compatibility. A single +5 V power supply, and a crystal or RC network for frequency determination are required.

Typical MPU System with Bus Extenders



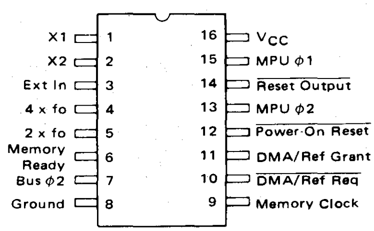
M6800 TWO-PHASE CLOCK GENERATOR/DRIVER

SCHOTTKY MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620-02

PIN CONNECTIONS



ORDERING INFORMATION

Device	Temperature Range	Package
MC6875L	0 to +70°C	Ceramic Dip
MC6875AL	-55 to +125°C	Ceramic Dip

3

ABSOLUTE MAXIMUM RATINGS (Unless otherwise noted $T_A = 25^oC$.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	+7.0	Vdc
Input Voltage	V_I	+5.5	Vdc
Operating Ambient Temperature Range	T_A		^oC
MC6875L		0 to +70	
MC6875AL		-55 to +125	
Storage Temperature Range	T_{stg}	–65 to +150	^oC
Operating Junction Temperature	T_J	175	^oC

NOTE:
Operation of the MC6875AL over the full military temperature range (to maximum T_A) will result in excessive operating junction temperature.

The use of a clip on 16 pin heat sink similar to AAVID Engineering, Inc., Model 5007 ($R_{\theta CA}$ = 18°C/W) is recommended above $T_A \approx$ 95°C.

Contact AAVID Engineering, Inc.
30 Cook Court
Laconia, New Hampshire 03246
Tel. (603) 524-4443

RECOMMENDED OPERATING CONDITIONS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	+4.75 to +5.25	Vdc
Operating Ambient Temperature Range	T_A	0 to +70	^oC

ELECTRICAL CHARACTERISTICS

(Unless otherwise noted specifications apply over recommended power supply and temperature ranges. Typical values measured at V_{CC} = 5.0 V and $T_A = 25^oC$.)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage – High Logic State					
MPU $\phi 1$ and $\phi 2$ 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 $\phi 2$ 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 $\phi 1$ and $\phi 2$ 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 $\phi 2$ 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
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_{IK}	–	–	-1.0	V
(V_{CC} = 4.75 V, I_{IC} = -5.0 mA) MC6875AL		–	–	-1.5	
Input Current – High Logic State					
Ext. In, Memory Ready and $\overline{\text{DMA/Refresh Request}}$ Inputs (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 DMA/Refresh Request Inputs at High Logic State)	I_{CCN}	–	–	150	mA
Memory Ready Stretch Operation (Memory Ready Input at Low Logic State; DMA/Refresh Request Input at High Logic State)	I_{CCMR}	–	–	135	mA
DMA/Refresh Request Stretch Operation (Memory Ready Input at High Logic State; 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 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

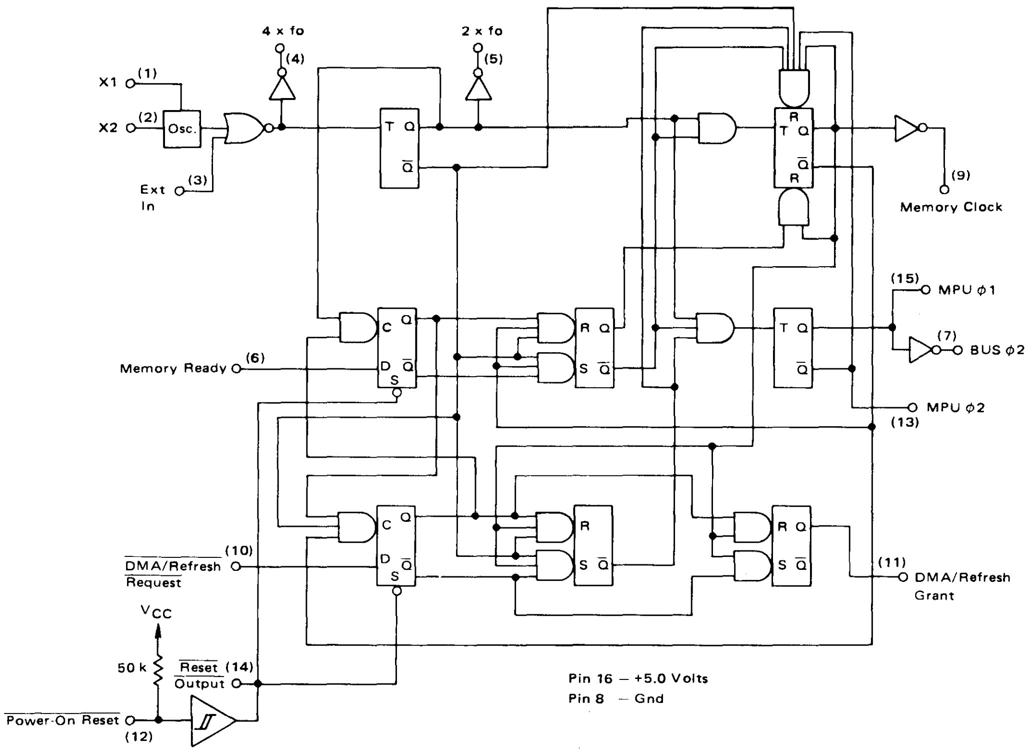


FIGURE 2 – TYPICAL HYSTERESIS CHARACTERISTIC OF RESET FUNCTION

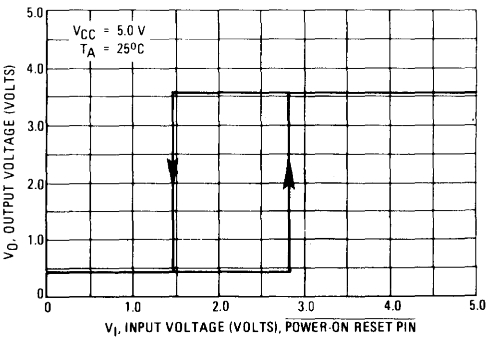


FIGURE 3 – TIMING DIAGRAM FOR MPU φ1 AND φ2

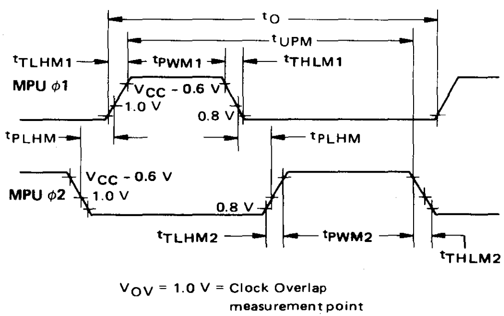


FIGURE 4 – TIMING DIAGRAM FOR NON-STRETCHED OPERATION
(Memory Ready and DMA/Refresh Request held high continuously)
Ext. In Input Voltage: 0 V to 3.0 V, f = 8.0 MHz, Duty Cycle = 50%, $t_{TLHEX} = t_{THLEX} = 5.0$ ns

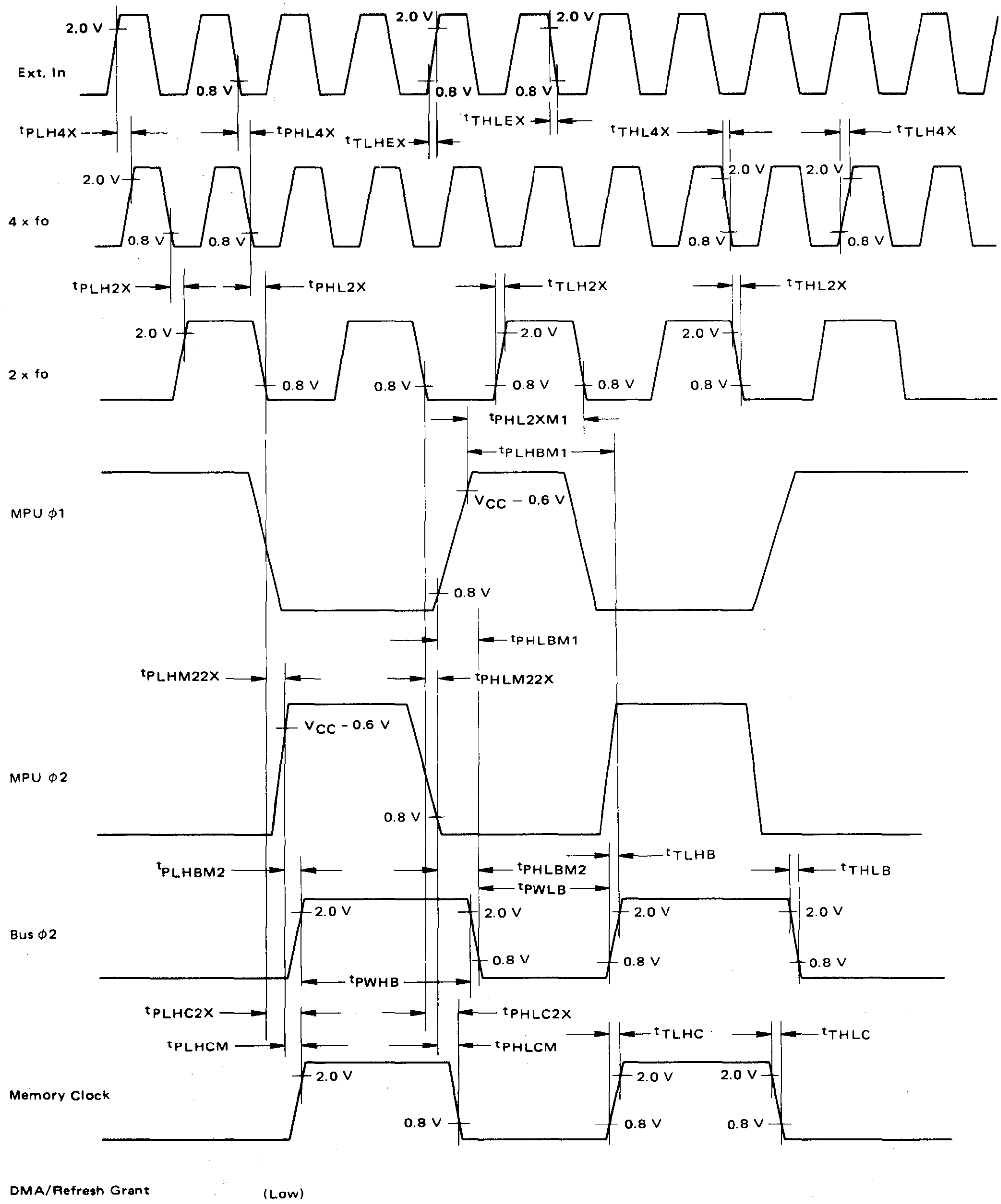


FIGURE 5 – TIMING DIAGRAM FOR MEMORY READY STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLMR} = t_{TLHMR} = 5.0$ ns

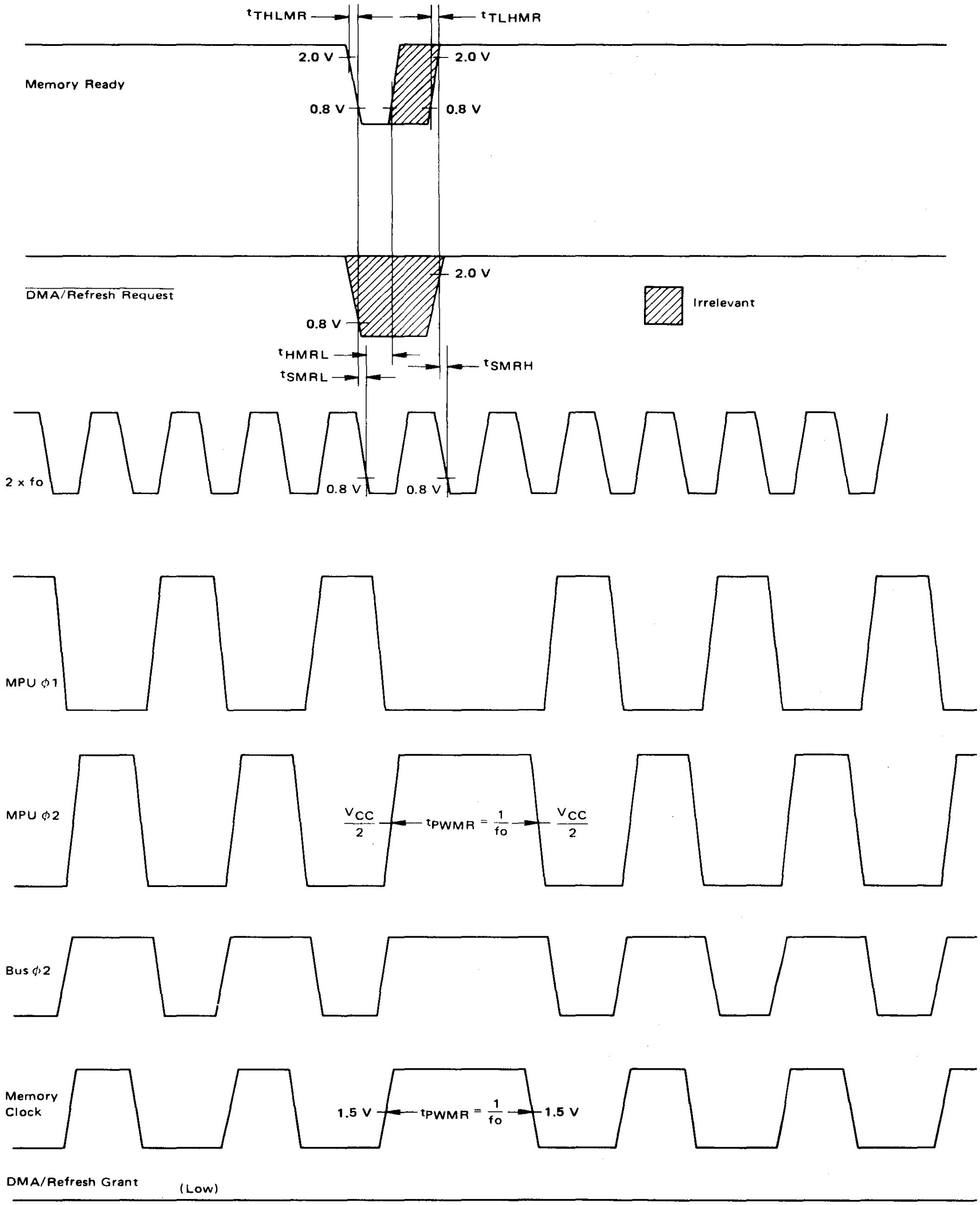
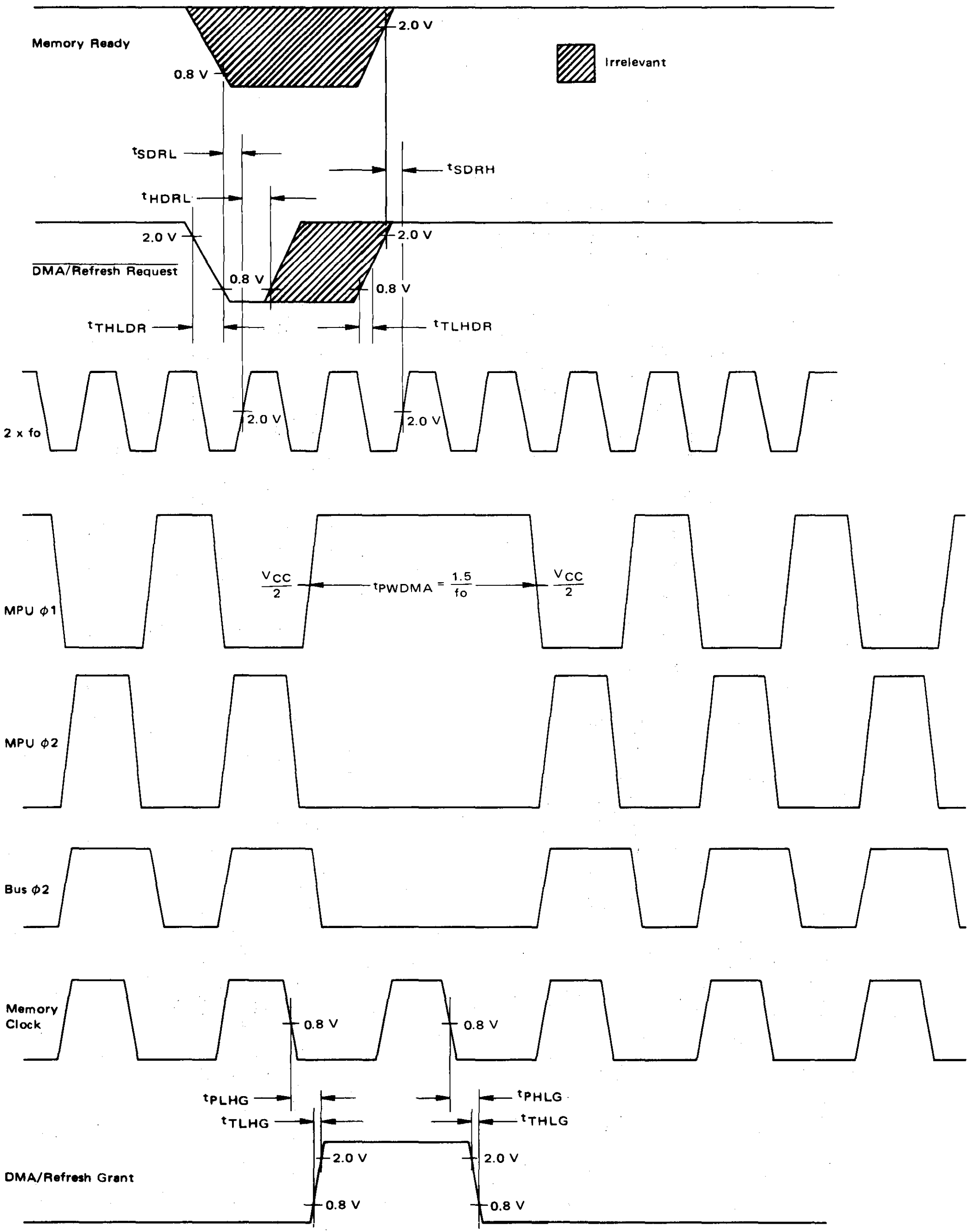


FIGURE 6 – TIMING DIAGRAM FOR $\overline{\text{DMA/REFRESH REQUEST}}$ STRETCH OPERATION
(Minimum Stretch Shown)
Input Voltage: 3.0 to 0 V, $t_{THLDR} = t_{TLHDR} = 5.0$ ns



3

FIGURE 7 – POWER ON RESET

Input Voltage: 0 to 5.0 V, f = 100 kHz – Pulse Width = 1.0 μs, $t_{TLH} = t_{THL}$ = 25 ns

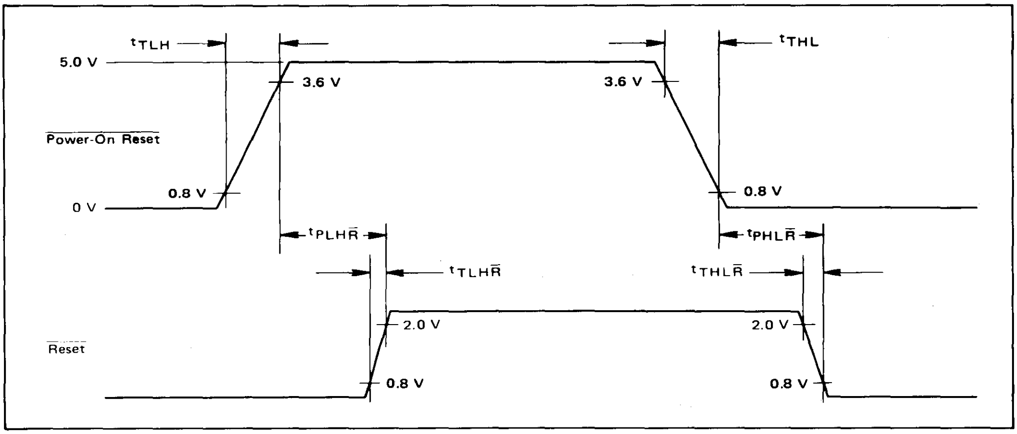
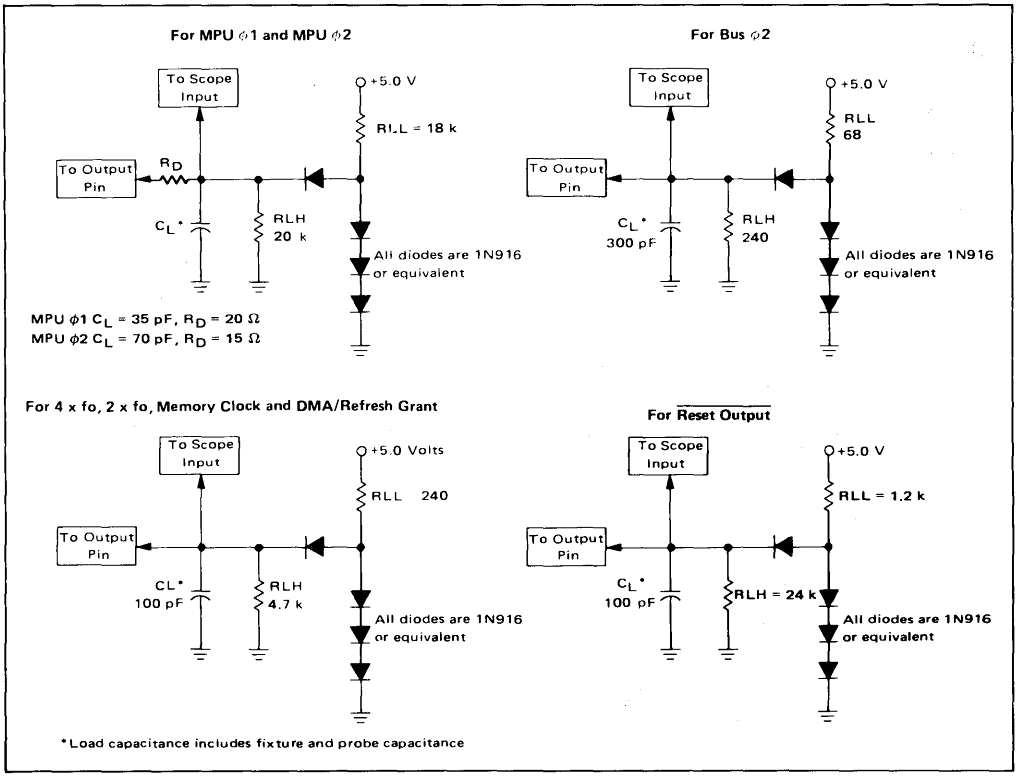


FIGURE 8 – LOAD CIRCUITS



NOTE:

Operation of the MC6875AL over the full military temperature range (to maximum T_A) will result in excessive operating junction temperature.

The use of a clip on 16 pin heat sink similar to AAVID Engineering, Inc., Model 5007 ($R_{\theta CA}$ = 18°C/W) is recommended above $T_A \approx 95°C$.

Contact AAVID Engineering, Inc.
30 Cook Court
Laconia, New Hampshire 03246
Tel. (603) 524-4443

APPLICATIONS INFORMATION

FIGURE 9 – TYPICAL RC FREQUENCY versus VOLTAGE

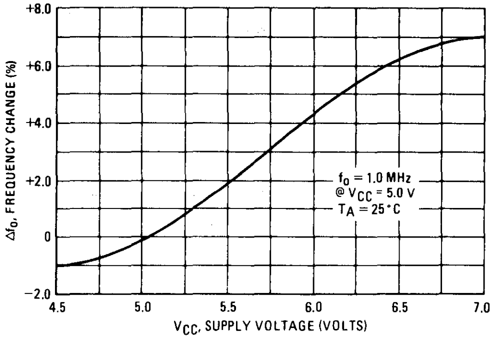


FIGURE 10 – TYPICAL RC FREQUENCY versus TEMPERATURE

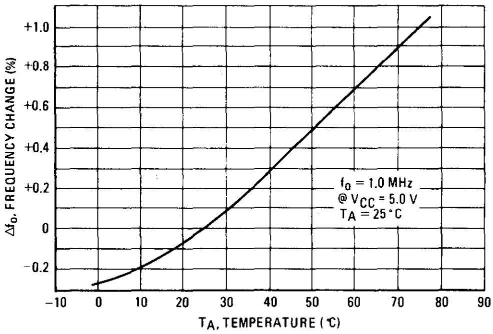
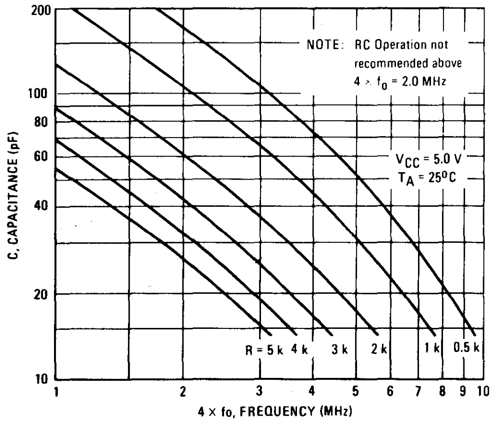


FIGURE 11 – TYPICAL FREQUENCY versus RESISTANCE FOR C VARIABLE



GENERAL

The MC6875 Clock Generator/Driver should be located on the same board and within two inches of the MC6800 MPU. Series damping resistors of 10-30 ohms may be utilized between the MC6875 and the MC6800 on the $\phi 1$ and $\phi 2$ clocks to suppress overshoot and reflections.

The V_{CC} pin (pin 16) of the MC6875 should be bypassed to the ground pin (pin 8) at the package with a 0.1 µF capacitor. Because of the high peak currents associated with driving highly capacitive loads, an adequately large ground strip to pin 8 should be used on the MC6875. Grounds should be carefully routed to minimize coupling of noise to the sensitive oscillator inputs. Unnecessary grounds or ground planes should be avoided near pin 2 or the frequency determining components. These components should be located as near as possible to the respective pins of the MC6875. Stray capacitance near pin 2 or the crystal, can affect the frequency. The can of the crystal should not be grounded. The ground side of the crystal or the C of the R-C oscillator should be connected as directly as possible to pin 8.

Unused inputs should be connected to V_{CC} or ground. $\overline{\text{Memory Ready}}$, $\overline{\text{DMA/Refresh Request}}$ 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

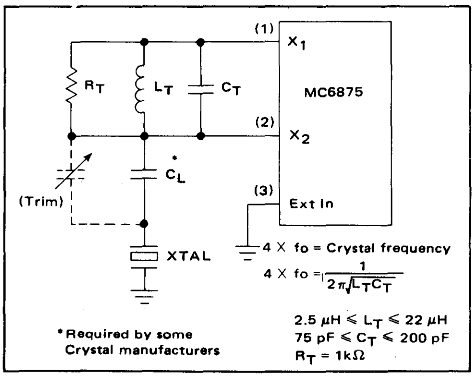
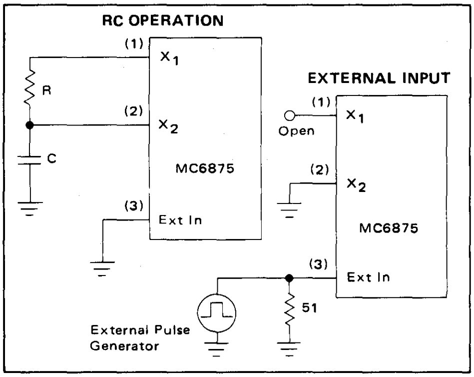


TABLE 1 — OSCILLATOR COMPONENTS

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

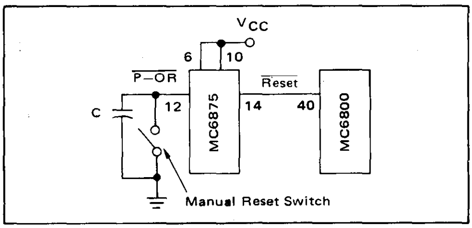
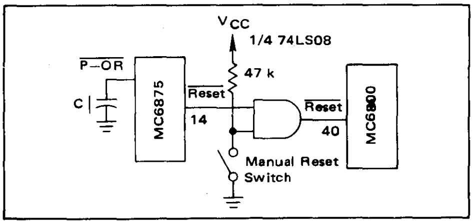


FIGURE 15 — MANUAL RESET FOR SYSTEMS USING DYNAMIC RAM OR SLOW STATIC RAM IN CONJUNCTION WITH MEMORY READY OR DMA/REFRESH REQUEST INPUTS



MOTOROLA

MC8T26A
MC6880A

QUAD THREE-STATE BUS TRANSCEIVER

This quad three-state bus transceiver features both excellent MOS or MPU compatibility, due to its high impedance PNP transistor input, and high-speed operation made possible by the use of Schottky diode clamping. Both the −48 mA driver and −20 mA receiver outputs are short-circuit protected and employ three-state enabling inputs.

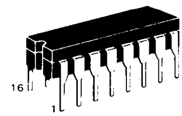
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier diode clamps to suppress excessive undershoot voltages.

The MC8T26A is identical to the NE8T26A and it operates from a single +5 V supply.

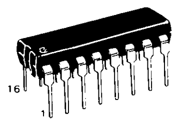
- High Impedance Inputs
- Single Power Supply
- High Speed Schottky Technology
- Three-State Drivers and Receivers
- Compatible with M6800 Family Microprocessor

QUAD THREE-STATE BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

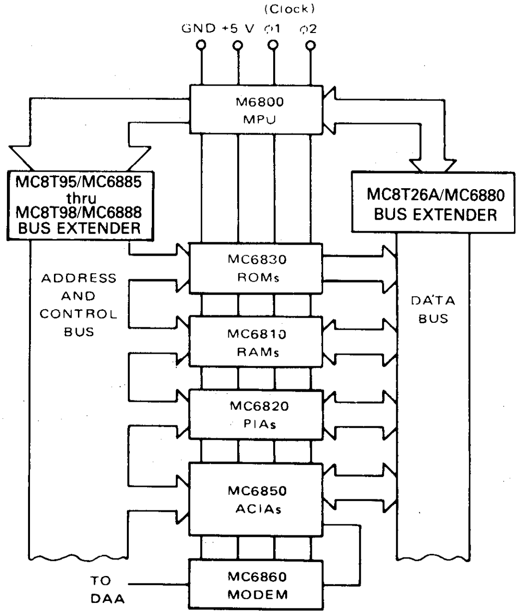


L SUFFIX
CERAMIC PACKAGE
CASE 620-02

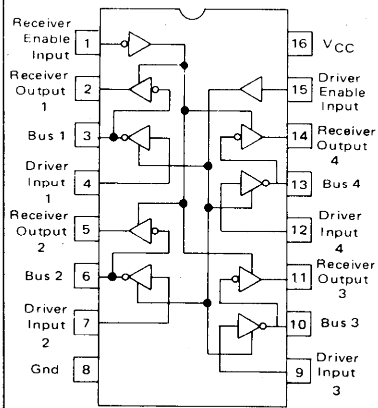


P SUFFIX
PLASTIC PACKAGE
CASE 648-05

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS – MC8T26A
MC6880A



ORDERING INFORMATION			
Device	**Alternate**	**Temperature Range**	**Package**
MC6880AL	MC8T26AL	0 to +75°C	Ceramic DIP
MC6880AP	MC8T26AP	0 to +75°C	Plastic DIP

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

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Junction Temperature	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 ≤ V_{CC} ≤ 5.25 V and 0°C ≤ T_A ≤ 75°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current – Low Logic State					
(Receiver Enable Input, $V_{IL(\overline{RE})}$ = 0.4 V)	$I_{IL(\overline{RE})}$	–	–	-200	µA
(Driver Enable Input, $V_{IL(DE)}$ = 0.4 V)	$I_{IL(DE)}$	–	–	-200	
(Driver Input, $V_{IL(D)}$ = 0.4 V)	$I_{IL(D)}$	–	–	-200	
(Bus (Receiver) Input, $V_{IL(B)}$ = 0.4 V)	$I_{IL(B)}$	–	–	-200	
Input Disabled Current – Low Logic State	$I_{IL(D)\,DIS}$				
(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)	$V_{IL(B)}$	–	–	0.55	
Input Voltage – High Logic State					
(Receiver Enable Input)	$V_{IH(\overline{RE})}$	2.0	–	–	V
(Driver Enable Input)	$V_{IH(DE)}$	2.0	–	–	
(Driver Input)	$V_{IH(D)}$	2.0	–	–	
(Receiver Input)	$V_{IH(B)}$	2.0	–	–	
Output Voltage – Low Logic State					
(Bus Driver) Output, $I_{OL(B)}$ = 48 mA)	$V_{OL(B)}$	–	–	0.5	V
(Receiver Output, $I_{OL(R)}$ = 20 mA)	$V_{OL(R)}$	–	–	0.5	
Output Voltage – High Logic State					
(Bus (Driver) Output, $I_{OH(B)}$ = –10 mA)	$V_{OH(B)}$	2.4	3.1	–	V
(Receiver Output, $I_{OH(R)}$ = –2.0 mA)	$V_{OH(R)}$	2.4	3.1	–	
(Receiver Output, $I_{OH(R)}$ = –100 µA, V_{CC} = 5.0 V)		3.5	–	–	
Output Disabled Leakage Current – High Logic State					
(Bus Driver) Output, $V_{OH(B)}$ = 2.4 V)	$I_{OHL(B)}$	–	–	100	µA
(Receiver Output, $V_{OH(R)}$ = 2.4 V)	$I_{OHL(R)}$	–	–	100	
Output Disabled Leakage Current – Low Logic State					
(Bus Output, $V_{OL(B)}$ = 0.5 V)	$I_{OLL(B)}$	–	–	–100	µA
(Receiver Output, $V_{OL(R)}$ = 0.5 V)	$I_{OLL(R)}$	–	–	–100	
Input Clamp Voltage					
(Driver Enable Input $I_{ID(DE)}$ = –12 mA)	$V_{IC(DE)}$	–	–	-1.0	V
(Receiver Enable Input $I_{IC(RE)}$ = +12 mA)	$V_{IC(RE)}$	–	–	-1.0	
(Driver Input $I_{IC(D)}$ = –12 mA)	$V_{IC(D)}$	–	–	-1.0	
Output Short-Circuit Current, V_{CC} = 5.25 V (1)					
(Bus (Driver) Output)	$I_{OS(B)}$	-50	–	-150	mA
(Receiver Output)	$I_{OS(R)}$	-30	–	-75	
Power Supply Current (V_{CC} = 5.25 V)	I_{CC}	–	–	87	mA

(1) Only one output may be short-circuited at a time.

3

SWITCHING CHARACTERISTICS (Unless otherwise noted, specifications apply at $T_A = 25^oC$ and $V_{CC} = 5.0$ V)

Characteristic	Symbol	Figure	Min	Max	Unit
Propagation Delay Time from Receiver (Bus) Input to High Logic State Receiver Output	$t_{PLH(R)}$	1	–	14	ns
Propagation Delay Time from Receiver (Bus) Input to Low Logic State Receiver Output	$t_{PHL(R)}$	1	–	14	ns
Propagation Delay Time from Driver Input to High Logic State Driver (Bus) Output	$t_{PLH(D)}$	2	–	14	ns
Propagation Delay Time from Driver Input to Low Logic State Driver (Bus) Output	$t_{PHL(D)}$	2	–	14	ns
Propagation Delay Time from $\overline{\text{Receiver Enable}}$ Input to High Impedance (Open) Logic State Receiver Output	$t_{PLZ(RE)}$	3	–	15	ns
Propagation Delay Time from $\overline{\text{Receiver Enable}}$ Input to Low Logic Level Receiver Output	$t_{PZL(RE)}$	3	–	20	ns
Propagation Delay Time from Driver Enable Input to High Impedance Logic State Driver (Bus) Output	$t_{PLZ(DE)}$	4	–	20	ns
Propagation Delay Time from Driver Enable Input to Low Logic State Driver (Bus) Output	$t_{PZL(DE)}$	4	–	25	ns

3

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

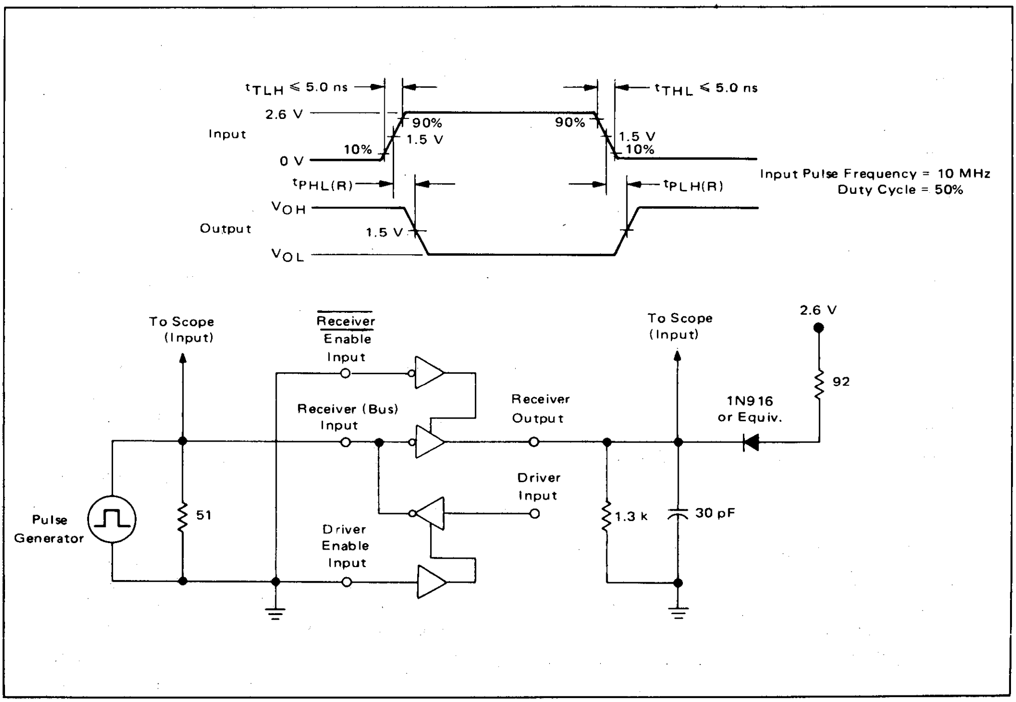


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

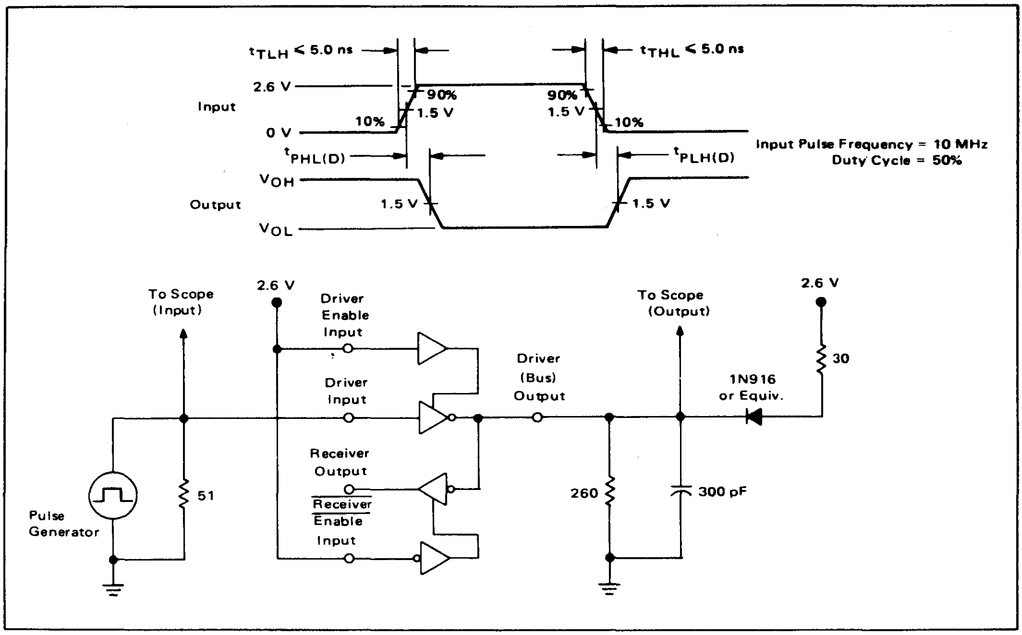


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

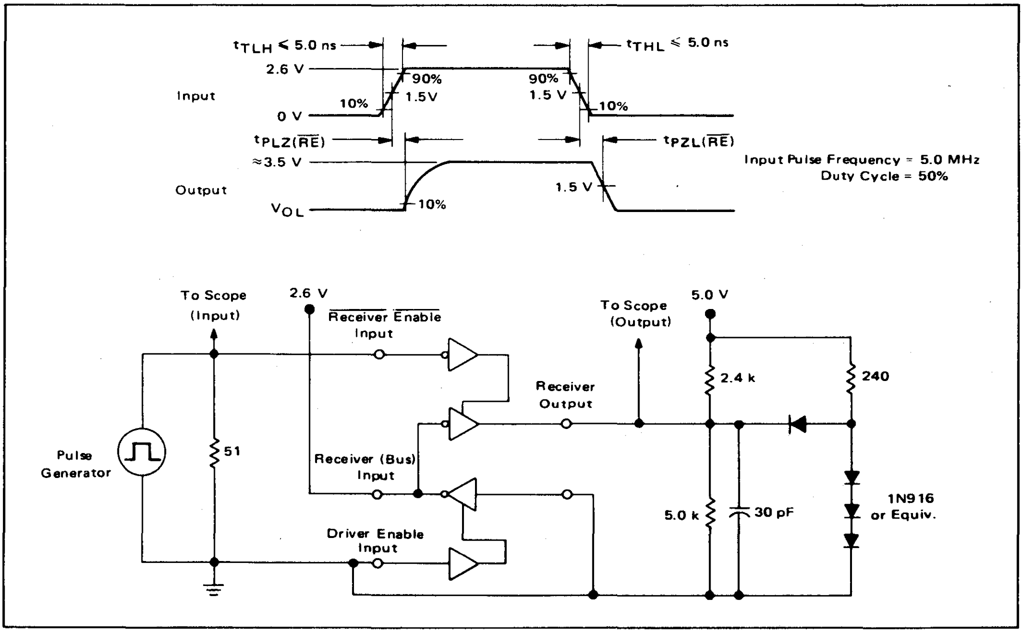


FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

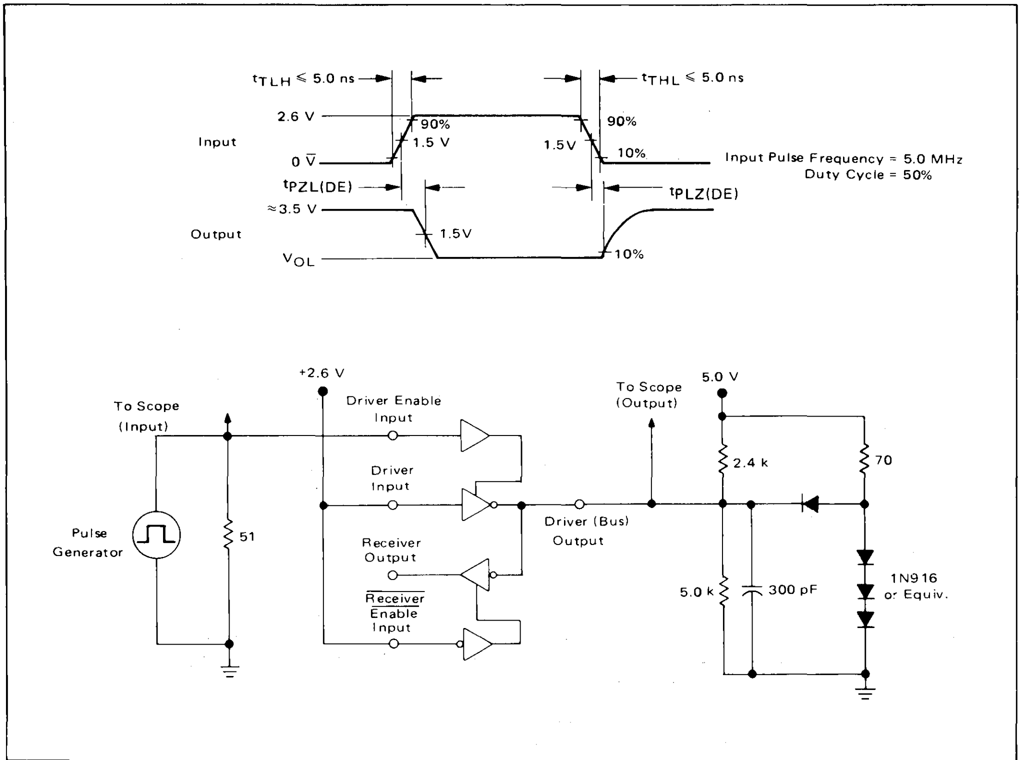
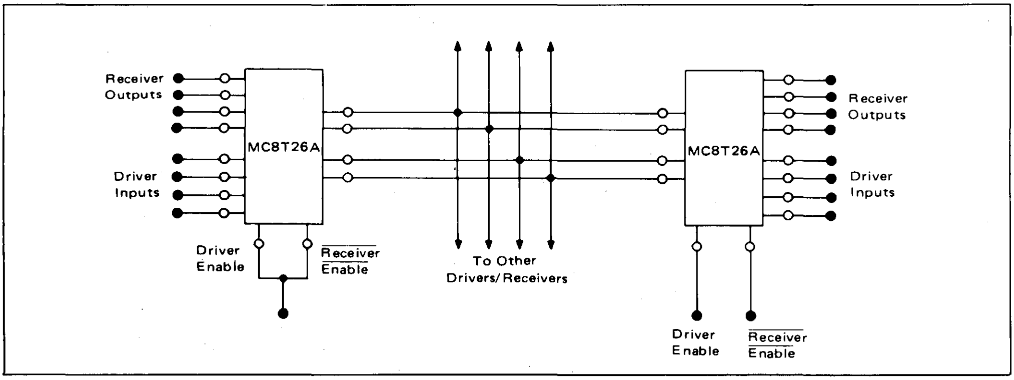


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS



MOTOROLA

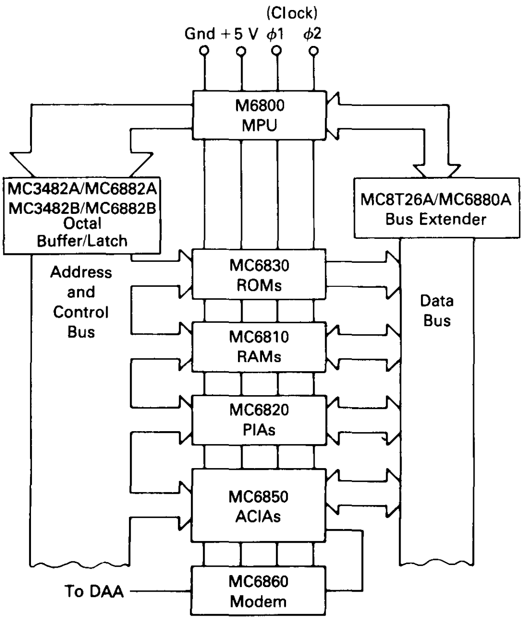
MC3482A/MC6882A
MC3482B/MC6882B

OCTAL THREE-STATE BUFFER/LATCH

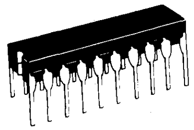
This series of devices combines four features usually found desirable in bus-oriented systems: 1) High impedance logic inputs insure that these devices do not seriously load the bus; 2) Three-state logic configuration allows buffers not being utilized to be effectively removed from the bus; 3) Schottky technology allows for high-speed operation; 4) 48 mA drive capability.

- Inverting and Non-Inverting Options of Data
- SN74S373 Function Pinouts
- Eight Transparent Latches/Buffers in a Single Package
- Full Parallel-Access for Loading and Reloading
- Buffered Control Inputs
- All Inputs Have Hysteresis to Improve Noise Rejection
- High Speed – 8.0 ns (Typ)
- Three-State Logic Configuration
- Single +5 V Power Supply Requirement
- Compatible with 74S Logic or M6800 Microprocessor Systems
- High Impedance PNP Inputs Assure Minimal Loading of the Bus

MICROPROCESSOR BUS EXTENDER APPLICATION



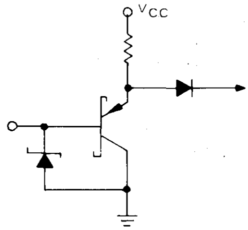
OCTAL THREE-STATE BUFFER/LATCH



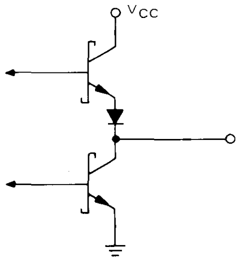
L SUFFIX
CASE 732-03

3

INPUT EQUIVALENT CIRCUIT



OUTPUT EQUIVALENT CIRCUIT



ORDERING INFORMATION

(Temperature Range for the following devices = 0 to +75°C.)

Device	Alternate	Package
MC3482AL	MC6882AL	Ceramic DIP
MC3482BL	MC6882BL	Ceramic DIP

3

MAXIMUM RATINGS ($T_A = 25^oC$ 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	^oC
Storage Temperature Range	T_{stg}	−65 to +150	^oC
Operating Junction Temperature Ceramic Package	T_J	 175	^oC

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, $0^oC \leq T_A \leq 75^oC$ and $4.75\ V \leq V_{CC} \leq 5.25\ V$)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Voltage — High Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IH}	2.0	–	–	V
Input Voltage — Low Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IL}	–	–	0.8	V
Input Current — High Logic State (V_{CC} = 5.25 V, V_{IH} = 2.4 V)	I_{IH}	–	–	40	μA
Input Current — Low Logic State (V_{CC} = 5.25 V, V_{IL} = 0.5 V, $V_{IL(\overline{OE})}$ = 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 (C_L = 20 pF)								ns
High Output Level to High Impedance	$t_{PHZ(\overline{OE})}$	–	8.0	15	–	6.0	13	
Low Output to High Impedance	$t_{PLZ(\overline{OE})}$	–	20	27	–	15	23	
High Impedance to High Output	$t_{PZH(\overline{OE})}$	–	9.0	16	–	11	18	
High Impedance to Low Output	$t_{PZL(\overline{OE})}$	–	13	20	–	9.0	16	

AC SETUP CHARACTERISTICS (V_{CC} = 5.0 V, 0°C ≤ T_A ≤ +75°C, unless otherwise noted, typical @ T_A = 25°C.)

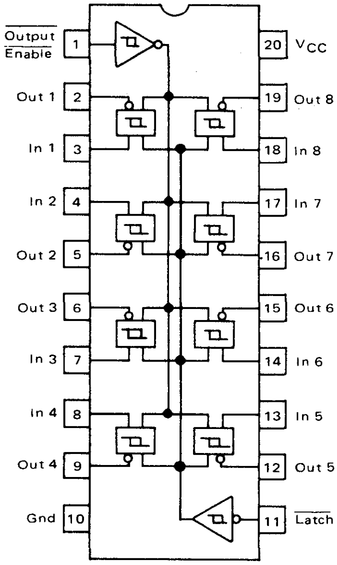
Characteristic	Symbol	MC3482A/ MC6882A			MC3482B/ MC6882B			Unit
		Min	Typ	Max	Min	Typ	Max	
Setup Time (Data to Negative Going Latch Enable)	$t_{su(D)}$	10	0	–	7.0	0	–	ns
Hold Time (Data to Negative Going Latch Enable)	$t_{h(D)}$	10	–	–	8.0	–	–	ns
Minimum Latch Enable Pulse Width (High or Low)	$t_{W(L)}$	–	15	–	–	15	–	ns

3

3

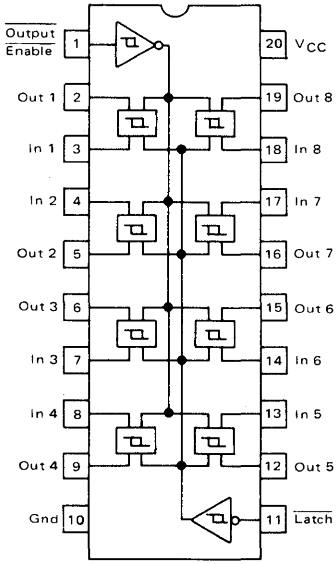
PIN CONNECTIONS AND TRUTH TABLES

MC3482A/MC6882A



Output Enable	Latch	Input	Output
0	1	0	1
0	1	1	0
0	0	X	Q_o
1	X	X	Z

MC3482B/MC6882B



Output Enable	Latch	Input	Output
0	1	0	0
0	1	1	1
0	0	X	Q_o
1	X	X	Z

FIGURE 1 – TEST CIRCUIT FOR SWITCHING CHARACTERISTICS

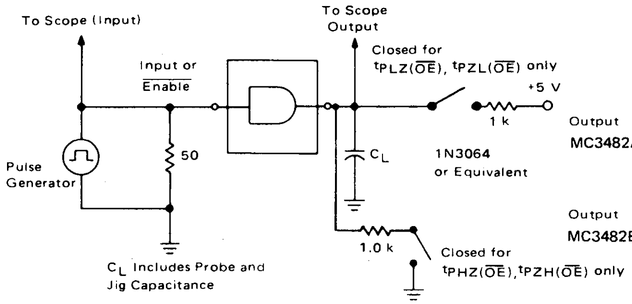


FIGURE 2 – WAVEFORMS FOR PROPAGATION DELAY TIMES DATA TO OUTPUT

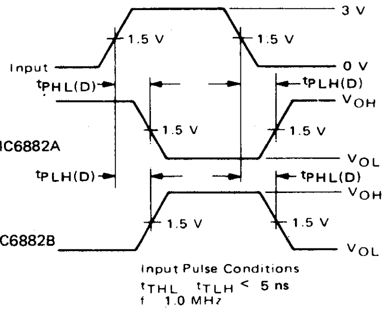


FIGURE 3 – WAVE FORMS FOR AC SETUP AND LATCH DISABLE TO OUTPUT DELAY

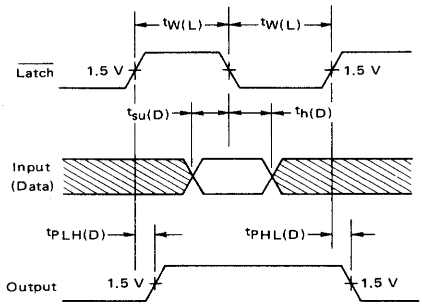
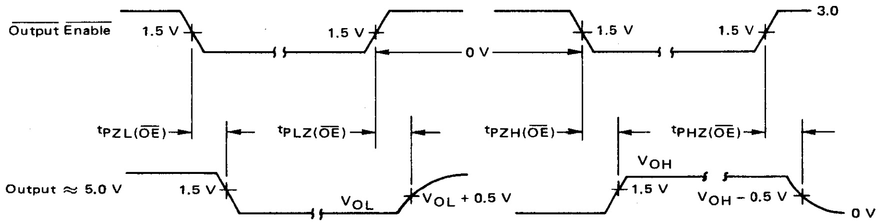


FIGURE 4 – WAVEFORMS FOR PROPAGATION DELAY TIMES – OUTPUT ENABLE TO OUTPUT



MOTOROLA

SN74LS783
MC6883

Advance Information

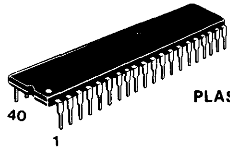
SYNCHRONOUS ADDRESS MULTIPLEXER

The SN74LS783/MC6883 brings together the MC6809E (MPU), the MC6847 (Color Video Display Generator) and dynamic RAM to form a highly effective, compact and cost effective computer and display system.

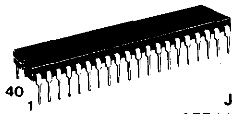
- MC6809E, MC6800, MC6801E, MC68000 and MC6847 (VDG) Compatible
- Transparent MPU/VDG/Refresh
- RAM size — 4K, 8K, 16K, 32K or 64K Bytes (Dynamic or Static)
- Addressing Range — 96K Bytes
- Single Crystal Provides All Timing
- Register Programmable:
  - VDG Addressing Modes
  - VDG Offset (0 to 64K)
  - RAM Size
  - Page Switch
  - MPU Rate (Crystal ÷ 16 or ÷ 8)
  - MPU Rate (Address Dependent or Independent)
- System "Device Selects" Decoded 'On Chip'
- Timing is Optimized for Standard Dynamic RAMs
- +5.0 V Only Operation
- Easy Synchronization of Multiple SAM Systems
- DMA Mode

SYNCHRONOUS ADDRESS MULTIPLEXER

LOW POWER SCHOTTKY

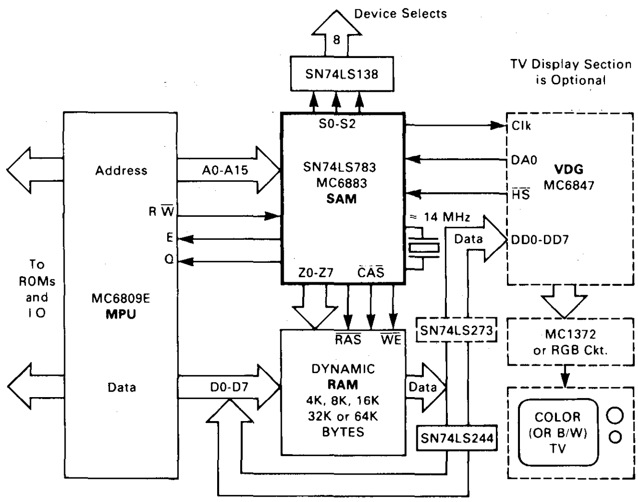


N SUFFIX
PLASTIC PACKAGE
CASE 711



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 ($\overline{RAS1}$)
7	VClk	Z6	34
8	DA0	Z5	33
9	$\overline{HS}$	Z4	32
10	$\overline{WE}$	Z3	31
11	$\overline{CAS}$	Z2	30
12	$\overline{RAS0}$	Z1	29
13	Q	Z0	28
14	E	S0	27
15	R/$\overline{W}$	S1	26
16	A0	S2	25
17	A1	A7	24
18	A2	A6	23
19	A3	A5	22
20	Gnd	A4	21

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

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_{IOscIn}	−0.5 to V_{CC}	Vdc
Input Current Osc_{In}	I_{IOscIn}	−0.5 to +5.0	mA

GUARANTEED OPERATING RANGES

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Operating Ambient Temperature Range	T_A	0	25	75	°C
Output Current High	I_{OH}				mA
$\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$		—	—	−1.0	
All Other Outputs		—	—	−0.2	
Output Current Low	I_{OL}				mA
$\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$		—	—	8.0	
VClk		—	—	0.8	
All Other Outputs		—	—	4.0	

DC CHARACTERISTICS (Unless otherwise noted specifications apply over recommended power supply and temperature ranges.)

Characteristic	Symbol	Min	Typ	Max	Units
Input Voltage — High Logic State	V_{IH}	2.0	—	—	V
Input Voltage — Low Logic State	V_{IL}	—	—	0.8	V
Input Clamp Voltage (V_{CC} = Min, I_{in} = −18 mA) All Inputs Except Osc_{In}	V_{IK}	—	—	−1.5	V
Input Current — High Logic State at Max Input Voltage	I_I				µA
(V_{CC} = Max, V_{in} = 5.25 V) VClk Input		—	—	200	
(V_{CC} = Max, V_{in} = 5.25 V) DA0 Input		—	—	100	
(V_{CC} = Max, V_{in} = 5.25 V Osc_{In} = Gnd) Osc_{Out} Input		—	—	250	
(V_{CC} = Max, V_{in} = 7.0 V) All Other Inputs Except Osc_{In}		—	—	100	
Input Current High Logic State (V_{CC} = Max, V_{in} = 2.7 V) All Inputs Except VClk, DA0 Osc_{In}, Osc_{Out}	I_{IH}	—	—	20	µA
Input Current — Low Logic State	I_{IL}				mA
(V_{CC} = Max, V_{in} = 0.4 V) DA0 Input		—	—	−1.2	
(V_{CC} = Max, V_{in} = 0.4 V) VClk Input		—	−30	−60	
(V_{CC} = Max, V_{in} = 0.4 V, Osc_{In} = Gnd) Osc_{Out} Input		—	—	−8	
(V_{CC} = Max, V_{in} = 0.4 V) All Other Inputs Except Osc_{In}		—	—	−.4	
Output Voltage — High Logic State					V
(V_{CC} = Min, I_{OH} = −1.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$	$V_{OH(C)}$	3.0	—	—	
(V_{CC} = Min, I_{OH} = −0.2 mA) E, Q	$V_{OH(E)}$	V_{CC} − 0.75	—	—	
(V_{CC} = Min, I_{OH} = −0.2 mA) All Other Outputs	V_{OH}	2.7	—	—	
Output Voltage — Low Logic State					V
(V_{CC} = Min, I_{OL} = 8.0 mA) $\overline{RAS0}$, $\overline{RAS1}$, $\overline{CAS}$, $\overline{WE}$	$V_{OL(C)}$	—	—	0.5	
(V_{CC} = Min, I_{OL} = 4.0 mA) E, Q Outputs	$V_{OL(E)}$	—	—	0.5	
(V_{CC} = Min, I_{OL} = 0.8 mA) VClk Output	$V_{OL(V)}$	—	—	0.6	
(V_{CC} = Min, I_{OL} = 4.0 mA) All Other Outputs	V_{OL}	—	—	0.5	
Power Supply Current	I_{CC}	—	180	230	mA
Output Short-Circuit Current	I_{OS}	30	—	225	mA

3

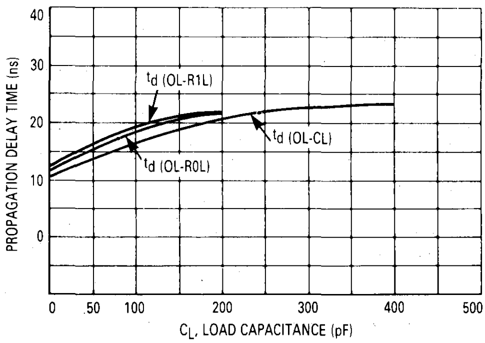
AC CHARACTERISTICS (4.75 V≤V_{CC}≤5.25 V and 0≤T_A≤70°C, unless otherwise noted).

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



3

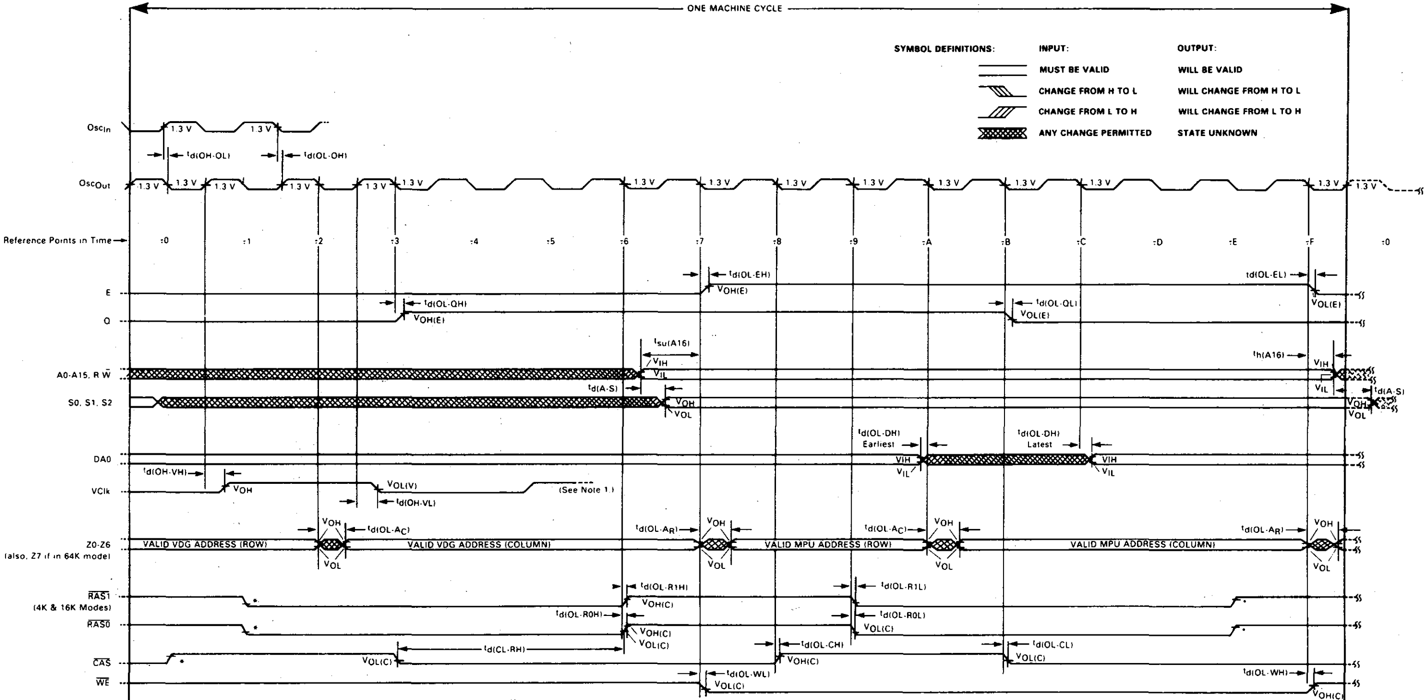
PIN DESCRIPTION TABLE

		Name	No.	Function
Input Pins	Power	V_{CC}	40	Apply + 5 volts ± 5%. SAM draws less than 230 mA.
		Gnd	20	Return Ground for +5 volts.
	MPU Address and Control	A15	36	Most Significant Bit.
		A14	37	
		A13	38	MPU address bits A0-A15. These 16 signals come directly from the MPU and are used to directly address up to 64K memory locations or to indirectly address up to 96K memory locations. (See pages 17 and 18 for memory maps). Each input is approximately equivalent to one low power Schottky load.
		A12	39	
		A11	1	
		A10	2	
		A9	3	
		A8	4	
		A7	24	
		A6	23	
		A5	22	
		A4	21	
		A3	19	
		A2	18	
		A1	17	
		A0	16	Least Significant Bit.
		$R/\overline{W}$	15	MPU READ or $\overline{WRITE}$. This signal comes directly from the MPU and is used to enable writing to the SAM control register, dynamic RAM (via $\overline{WE}$), and to enable device select #0.
		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.

3

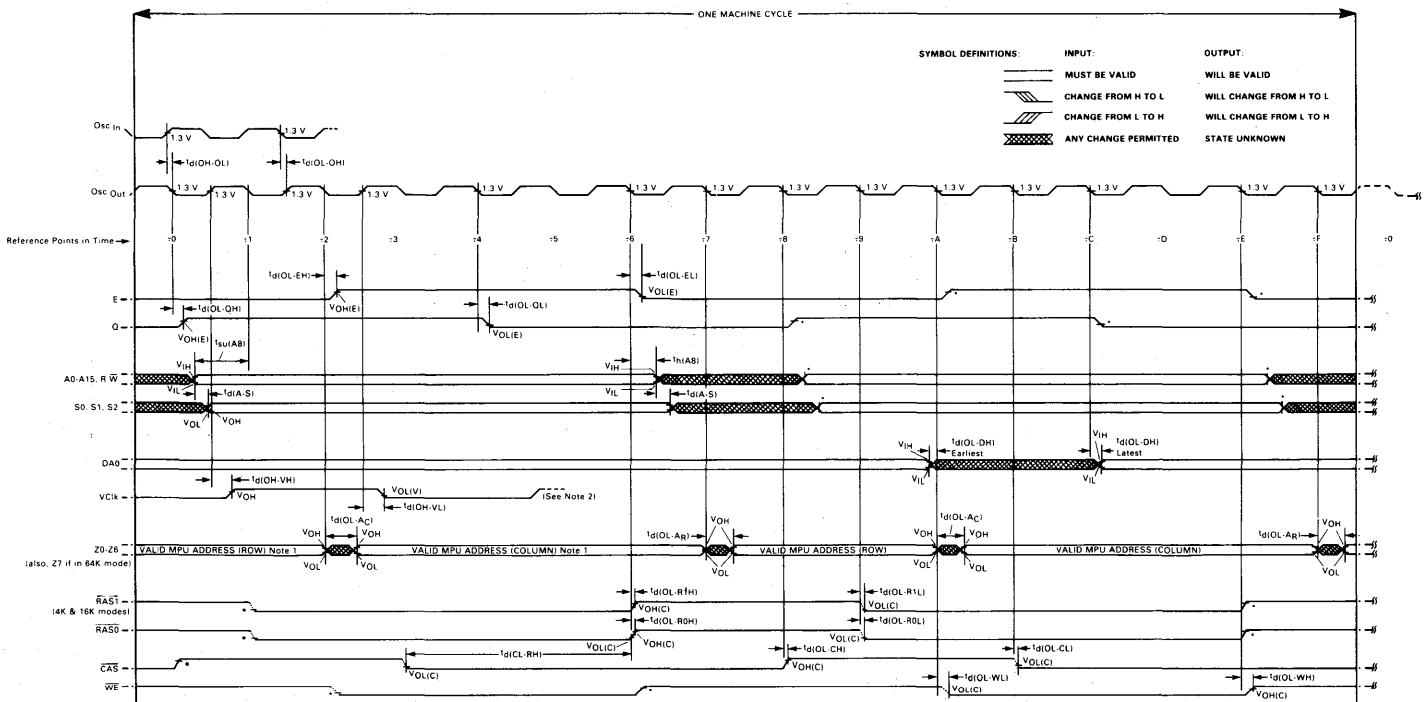
FIGURE 2 — TIMING WAVEFORMS for MPU RATE = SLOW



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



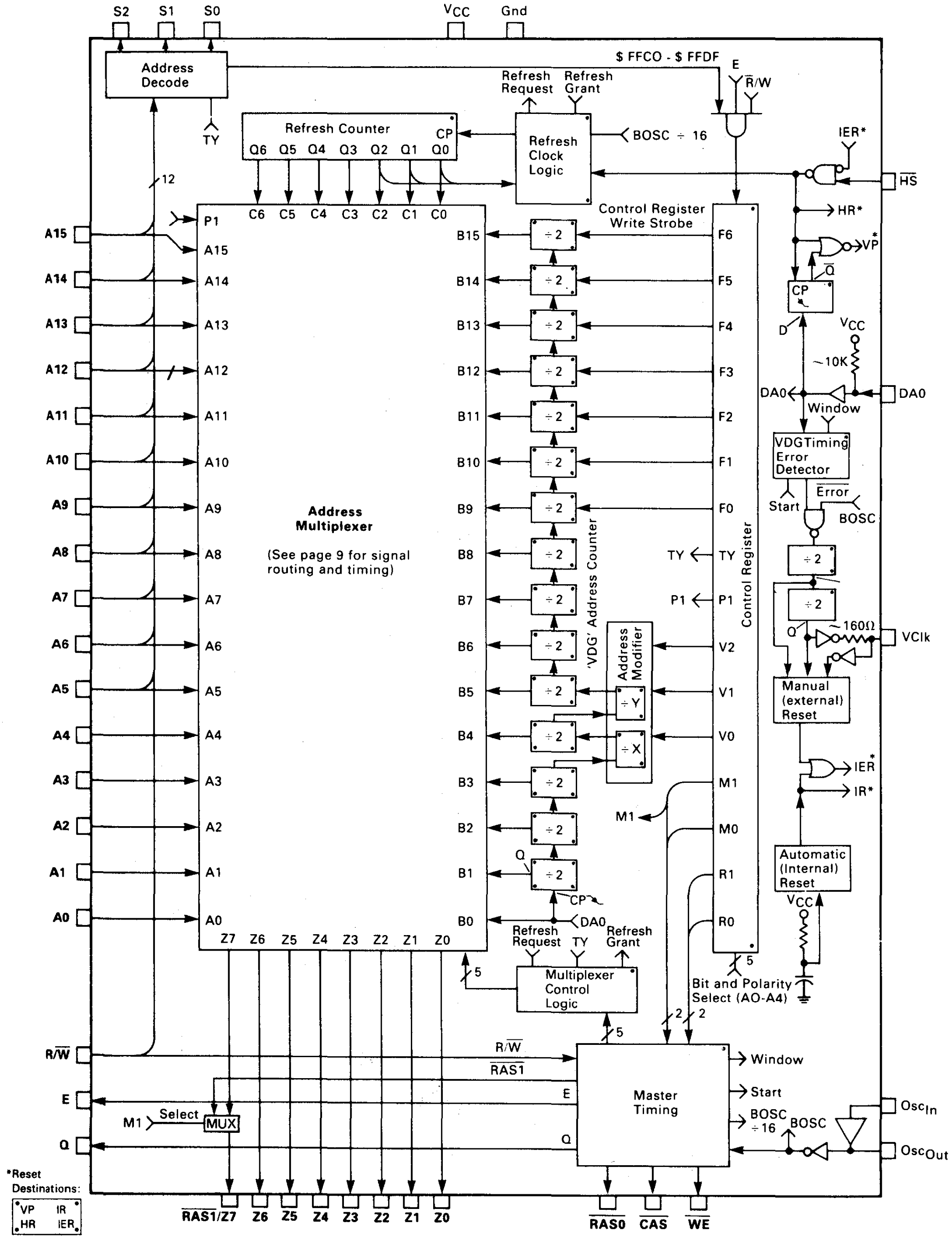
Timing points marked with "" are defined elsewhere (specifically, 8 cycles of "Osc_{Out}" to the left or the right.)

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 $t_{d(OH-VH)}$ nanoseconds of $\tau0, \tau1, \tau2, \ldots$ or τF.

3

FIGURE 4 — SAM BLOCK DIAGRAM



Dots indicate which internal signals reset logic blocks

SAM BLOCK DIAGRAM DESCRIPTION

MPU Addresses (A0 – A15):

These 16 signals come directly from the MPU and are used to directly address up to 64K memory locations (K=1024) or to indirectly address up to 96K memory locations, by using a paging bit "P" (see pages 17 and 18 for memory maps.) Each input is approximately equivalent to one low power Schottky load.

VDG Address Counter (B0 – B15):

These 16 signals are derived from one input (DA0) which is the least significant bit of the VDG address. Most of the counter is simply binary. However, to duplicate the various addressing modes of the MC6847 VDG, ADDRESS MODIFIER logic is used. Selected by three VDG mode bits (V2, V1, and V0) from the SAM CONTROL REGISTER, eight address modifications are obtained as shown in Figure 5.

Also, notice that bits B9-B15 may be loaded from bits F0-F6 from the CONTROL REGISTER. This allows the starting address of the VDG display to be offset (in ½K increments) from \$0000 to \$FFFF†. B9-B15 are loaded when a VERTICAL PRE-LOAD(VP) pulse is generated. VP goes active (high) when $\overline{HS}$ from the VDG rises if DA0 is high (or a high impedance.) This condition should occur only while the TV electron beam is in vertical blanking and is simply implemented by connecting $\overline{FS}$ and $\overline{MS}$ together on the MC6847. The VP pulse also **clears** bits B1 – B8.

Finally, a HORIZONTAL RESET (HR) pulse may also affect the counter by clearing bits B1 – B3 or B1 – B4 when $\overline{HS}$ from the VDG is LOW (see Figure 5.) The HR pulse should occur only while the TV electron beam is in horizontal blanking.

In summary, DA0 clocks the VDG ADDRESS COUNTER; HR initializes the horizontal portion and VP initializes the vertical portion of the VDG ADDRESS COUNTER.

3

REFresh Address Counter (C0 – C6):

A seven bit binary counter with outputs labeled C0 – C6 supplies bursts of eight* sequential addresses triggered by a $\overline{HS}$ high to low transition. Thus, while the TV electron beam is in horizontal blanking, eight sequential addresses are accessed. Likewise, the next eight addresses are accessed during the next horizontal blanking period, etc. In this manner, all 128 addresses are refreshed in less than 1.1 milliseconds.

Address Multiplexer:

Occupying a large portion of the block diagram in Figure 4, is the address multiplexer which outputs bits Z0-Z7 (as addresses to dynamic RAM's.) Inputs to the address multiplexer include the VDG address (B0 – B15) the REFresh address (C0- C6) and the MPU address (A0 – A15) or (A0 – A14 plus one paging bit "P".) The paging bit "P" is one bit in the SAM CONTROL REGISTER that is used in place of A15 when memory map TYpe #0 is selected (via the SAM CONTROL REGISTER "TY" bit.)

Figure 6 shows which inputs are routed to Z0 – Z7 and **when** the routing occurs relative to one SAM machine cycle. Notice that Z7 and $\overline{RAS1}$ share the same pin. Z7 is selected if "M1" in the SAM CONTROL REGISTER IS HIGH (Memory size = 64K.)

Address Decode:

At the top left of Figure 4, is the Address Decode block. Outputs S2, S1, and S0 form a three bit encoded binary word(S). Thus S may be one of eight values (0 through 7) with each value representing a different range of MPU addresses. (To enable peripheral ROM's or I/O, decode the S2, S1, and S0 bits into eight seperate signals by using a 74LS138, 74LS155 or 74LS156. Notice that S2, S1, and S0 are **not** gated with any timing signals such as E or Q.)

Along with the A5 – A15 inputs is the MEMORY MAP TYpe bit (TY.) This bit is soft-programmable (as are **all** 16 bits in the SAM CONTROL REGISTER,) and selects one of two memory maps. Memory map #0 is intended to be used in systems that are primarily **ROM** based. Whereas, memory map #1 is intended for a primarily **RAM** based system with 64K contiguous RAM locations (minus 256 locations.) The various meanings of S2, S1, S0 are tabulated in Figure 16 (page 19) and again on pages 17 and 18.

In addition to S2, S1, and S0 outputs is a decode of \$FFC0 through \$FFDF which, when gated with E and $R/\overline{W}$, results in the write strobe for the SAM CONTROL REGISTER.

SAM Control Register

As shown in Figure 4, the CONTROL REGISTER has 16 "outputs":

VDG Addressing Modes:	V2, V1, V0	MPU Rate:	R1, R0
VDG Address OFFset:	F6, F5, F4, F3, F2, F1, F0	Memory Size (RAM):	M1, M0
32K Page Switch:	P	Memory Map TYpe:	TY

When the SAM is reset (see page 10,) all 16 bits are cleared. To **set** any one of these 16 bits, the MPU simply **writes** to a unique** **odd** address (within \$FFC1 through \$FFDF.) To **clear** any one of these 16 bits, the MPU

* If $\overline{HS}$ is held low longer than 8 μs, then the number of sequential addresses in one refresh "BURST" is proportional to the time interval during which $\overline{HS}$ is low.

** See pages 17 or 18 for specific addresses.

† In this document, the "\$" symbol always preceeds hexidecimal characters.

simply writes to a unique** **even** address (within $FFCO through $FFDE.) Note that the **data** on the MPU data bus is irrelevant.

Inputs to the control register include A4, A3, A2, A1 (which are used to select **which one** of 16 bits is to be cleared or set), A0 (which determines the polarity . . . clear or set,) and $\overline{\text{R}}$/W, E and $FFCO – $FFDF (which restrict the method, timing and addresses for changing one of the 16 bits.) For more detailed descriptions of the purposes of the 16 control bits, refer to related sections in the BLOCK DIAGRAM DESCRIPTION (pages 8 through 12) and the PROGRAMMING GUIDE (pages 14 through 18).

** See pages 17 or 18 for specific addresses.

FIGURE 5 — VDG ADDRESS MODIFIER

Mode			Division Variables		Bits Cleared by $\overline{\text{HS}}$ (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	M1	M0	Signal Source	Row/Column	Signals Routed to Z0–Z7 Z7	Z6	Z5	Z4	Z3	Z2	Z1	Z0	Timing (Figure 2)
4K	0	0	MPU	ROW	*	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	*	L	A11	A10	A9	A8	A7	A6	TA-TF
			VDG	ROW	*	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	*	L	B11	B10	B9	B8	B7	B6	T2-T7
			REF	ROW	*	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	*	L	L	L	L	L	L	L	T2-T7
16K	0	1	MPU	ROW	*	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	*	A13	A12	A11	A10	A9	A8	A7	TA-TF
			VDG	ROW	*	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	*	B13	B12	B11	B10	B9	B8	B7	T2-T7
			REF	ROW	*	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	*	L	L	L	L	L	L	L	T2-T7
64K (dynamic)	1	0	MPU	ROW	A7	A6	A5	A4	A3	A2	A1	A0	T7-TA
				COL	P/A15***	A14	A13	A12	A11	A10	A9	A8	TA-TF
			VDG	ROW	B7	B6	B5	B4	B3	B2	B1	B0	TF-T2
				COL	B15	B14	B13	B12	B11	B10	B9	B8	T2-T7
			REF	ROW	L	C6	C5	C4	C3	C2	C1	C0	TF-T2
				COL	L	L	L	L	L	L	L	L	T2-T7
64K (static)	1	1	MPU	ROW	A7	A6	A5	A4	A3	A2	A1	A0	T7-**T9**
				COL	P/A15***	A14	A13	A12	A11	A10	A9	A8	**T9**-TF
			VDG	ROW	B7	B6	B5	B4	B3	B2	B1	B0	TF-**T1**
				COL	B15	B14	B13	B12	B11	B10	B9	B8	**T1**-T7
			REF	ROW	L	C6	C5	C4	C3	C2	C1	C0	TF-**T1**
				COL	L	L	L	L	L	L	L	L	**T1**-T7

Notes: "L" implies logical LOW level.

*Z7 functions as RAS1 and its level is address dependent. For example, when using two banks of 16K x 1 RAMs, $\overline{\text{RAS0}}$ 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.

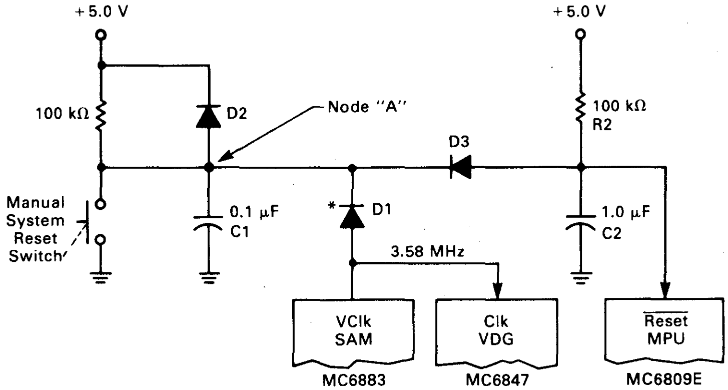
External Reset

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.

3

FIGURE 7 — EXTERNAL RESET CIRCUITRY



VDG Synchronization

In order for the VDG and MPU to share the same dynamic RAM (see page 13,) the **VDG clock must be stopped** until the VDG data fetch and MPU data fetch are synchronized as shown in Figure 12. Once synchronized, the VDG clock resumes its 3.579545 MHz rate and is not stopped again unless an extreme temperature change (or SAM reset) occurs. When stopped, the VDG clock remains stopped for **no more than** 32 Osc_{Out} cycles (approximately 2 microseconds.)

In the block diagram in Figure 4, DA0 enters a block labeled VDG Timing Error Detector. If DA0 rises **between** time reference points** τ_A and τ_C, then $\overline{Error}$ is high and VClk is the result of dividing BOSC (Buffered Osc_{Out} $\approx$ 14 MHz) by four. However, if DA0 rises **outside** the time Window τ_A to τ_C, then $\overline{Error}$ goes LOW and the VDG stops. A START pulse at time reference point τ_B (center of Window) restarts the VDG . . . properly synchronized.

*Use a diode with sufficiently low forward voltage drop to meet V_{IL} requirement at VClk.

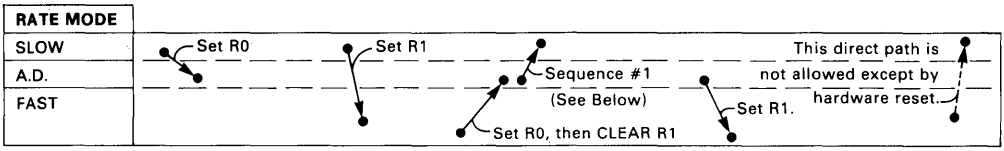
**See timing diagrams on page 5 and 6.

Changing the MPU Rate (by changing SAM control register bits R0, R1).

Two bits in the SAM control register determine the period of both "E" and "Q" MPU clocks. Three rate modes are implemented as follows:

RATE MODE	R1	R0	
SLOW	0	0	The frequency of "E" (and "Q") is f crystal ÷ 16. This rate mode is automatically selected when the SAM is reset. Note that system timing is least critical in this "SLOW" rate mode.
A.D. (Address Dependent)	0	1	The frequency of "E" (and "Q") is either f crystal ÷ 16 or f crystal ÷ 8, depending on the address the MPU is presenting.
FAST	1	X	The frequency of "E" (and "Q") is f crystal ÷ 8. This is accomplished by stealing the time that is normally used for VDG/REFRESH, and using this time for the MPU. Note: Neither VDG display nor dynamic RAM refresh are available in the "FAST" rate mode. (Both are available in SLOW and A.D. rate modes).

When changing between any two of the three rate modes, the following procedures must be followed to ensure that MPU timing specifications are met:



May be ANY address from $0000 to $7FFF.

SEQUENCE #1:

```
7D 00 00 TST #$0000 . . . Synchronizes STA instruction to write during T2-TG (See Figure #8).*
21 00    BRN 00
B7 FF D6 STA #$FFD6 . . . Clears bit R0
```

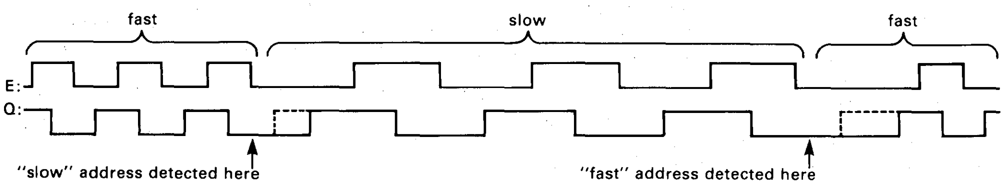
*Note: "TST" instruction affects MC6809E condition code register.

Changing the MPU Rate (In Address Dependent Mode)

When the SAM control register bits "R1", and "R0" are programmed to "0" and "1", respectively, the Address Dependent Rate Mode is selected. In this mode, the ÷ 16 MPU rate is automatically used when addressing within $0000 to $7FFF* or $FF00 to $FF1F ranges. Otherwise the ÷ 8 MPU rate is automatically used. (Refer to Figure 8 for sample "E" and "Q" waveforms yielding ÷ 8 to ÷ 16 and ÷ 16 to ÷ 8 rate changes). This mode often nearly doubles the MPU throughput while still providing transparent VDG and dynamic, RAM refresh functions. For example, since much of the MPU's time may be spent performing internal MPU functions (address = $FFFF)**, accessing ROM (address = $8000 to $FEFF) or accessing I/O (address = $FF20 — $FF5F), the faster f crystal ÷ 8 MPU rate may be used much of the time.

Note: The VDG operates normally when using the SLOW or A.D. rate modes. However, in the FAST rate mode, the VDG is not allowed access to the dynamic RAM.

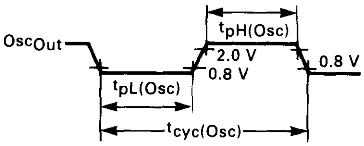
FIGURE 8 — RATE CHANGE E AND Q WAVEFORMS



*When using Memory Map 0, addresses $0000 to $7FFF may access Dynamic RAM.
**The MC6809 outputs $FFFF on A0–A15 when no other valid addresses are being presented.

Oscillator

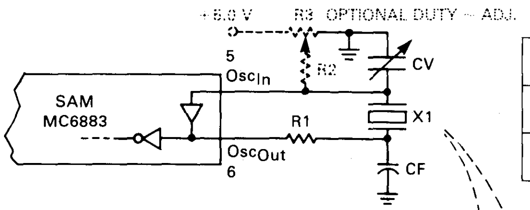
In Figure 4, an amplifier between Osc_{In} and Osc_{Out} provides the gain for oscillation (using a crystal as shown in Figure 9.) Alternately, Pin 5 (Osc_{In}) may be grounded while Pin 6 (Osc_{Out}) may be driven at low-power Schottky levels as shown in Figure 10. Also, see V_{IH}, V_{IL} on page 2.



AC Specifications*

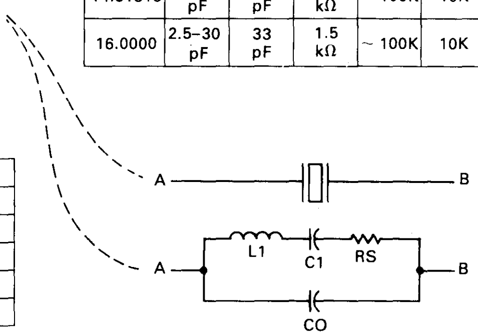
	Max	Typ	Min	Units
$t_{pH(Osc)}$	—	30	22	ns
$t_{pL(Osc)}$	—	30	22	ns
$t_{cyc(Osc)}$	—	70	62.4	ns

FIGURE 9 — CRYSTAL OSCILLATOR



Suggested Component Values

Freq. MHz	CV*	CF*	R1*	R2*	R3*	X1
14.31818	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*
16.0000	2.5–30 pF	33 pF	1.5 kΩ	~ 100K	10K	*

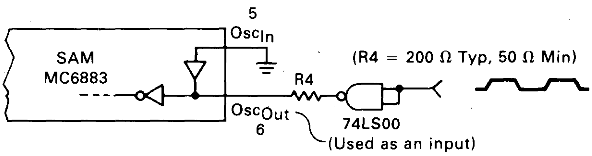


Recommended Crystal Parameters

	14.31818 MHz**	16.0000 MHz**
R_S	10 Ω ± 2.0 Ω	10 Ω ± 2.0 Ω
CO	5.0 pF ± 1.5 pF	6.0 pF ± 1.0 pF
C1	0.0245 pF ± 15%	0.0319 pF ± 15%
L1	5.05 mH	3.1 mH
Q	50K ± 10K	40K ± 10K

Calibration Tolerance: 0.002% at 26°C
Temperature Tolerance: 0.001% 0°C to 70°C

FIGURE 10 — TTL CLOCK INPUT



Typical input capacitances are 3.0 pF for Pin 5 and 5.5 pF for Pin 6.

*Optimum values depend on characteristics of the crystal (X1). For many applications, VClk must be 3.579545 MHz ± 50 Hz! Hence, Osc_{Out} must be made similarly "drift resistant" (by balancing temperature coefficients of X1, CV, CF, R1, R2 and R3).

**Specifically cut for MC6883 are International Crystal Manufacturing, Inc. Crystals (#167568 for 14.31818 MHz or #167569 for 16.0 MHz). However, other crystals may be used.

THEORY OF OPERATION

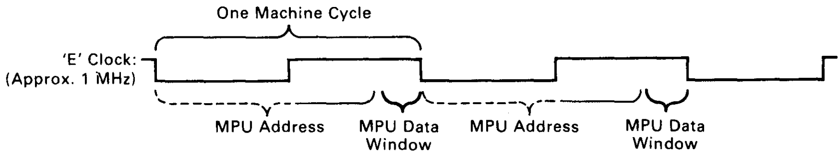
Video or No Video

Although the MC6883 may be used as a dynamic RAM controller **without** a video display*, most applications are likely to include a MC6847 video display generator (VDG). Therefore, this document emphasizes MC6883 with MC6847 systems.

Shared RAM (with interleaved DMA)

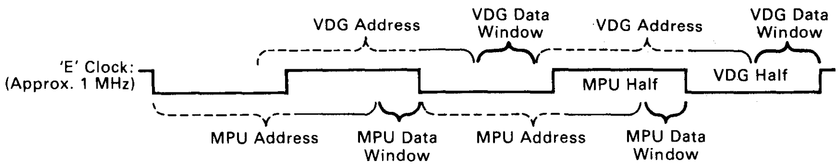
To minimize the number of RAM and interface chips, both the MPU and VDG share common dynamic RAM. Yet, the use of common RAM creates an apparent difficulty. That is, the MPU and VDG must both access the RAM without contention. This difficulty is overcome by taking advantage of the timing and architecture of Motorola MPU's (MC6800, MC6801E, MC6809E, MC68000). Specifically, **all** MPU accesses of external memory **always** occur in the **latter half** of the machine cycle, as shown below:

FIGURE 11 — MOTOROLA MPU TIMING



Similarly, the MC6847 (non-interlaced) VDG transfers a data byte in a half machine cycle (E or Φ2). Thus, when properly positioned, VDG and MPU RAM accesses interleave without contention as shown below:

FIGURE 12 — MOTOROLA MPU WITH VDG TIMING



This Interleaved Direct Memory Access (IDMA) is synchronized via the MC6883 by centering the VDG data window half-way between MPU data windows.**

The result is a shared RAM system without MPU/VDG RAM access contention, with both MPU and VDG running uninterrupted at normal operating speed, each transparent to the other.

RAM Refresh

Dynamic RAM refresh is accomplished by accessing eight*** sequential addresses every 64*** microseconds until 128 consecutive addresses have been accessed. To avoid RAM access contention between REFRESH and MPU, each of the 128 refresh accesses occupies the "VDG half" of the interleaved DMA (IDMA). Furthermore, refresh accesses occur only during the television retrace period (at which time the VDG doesn't need to access RAM).

In summary, the VDG, MPU and MC6883's Refresh Counter all transparently access the common dynamic RAM without contention or interruption.

Why IDMA?

Use of the interleaved direct memory access results in fast modification to variable portions of display RAM, by the MPU, without any distracting flashes on the screen (due to RAM access contention.) In addition, the MPU is not slowed down nor stopped by the MC6883; thereby, assuring accurate software timing loops without costly additional hardware timers. Furthermore, additional hardware and software to give "access permission" to the MPU is eliminated since the MPU may access RAM at **any** time.

* Only 1 pin, (DA0) out of 40 pins is dedicated to the video display.
** See VDG synchronization (page 10) for more detail.
*** When not using a MC6847, $\overline{HS}$ 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 ($\overline{\text{RAS}}$), column address select ($\overline{\text{CAS}}$) and write enable ($\overline{\text{WE}}$).

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 $\overline{\text{RAS0}}$) the following combinations require no additional address logic.

FIGURE 13 — RAM CONFIGURATIONS

Address: MSB ... LSB	Chip Select:	
Z5Z4Z3Z2Z1Z0	$\overline{\text{RAS0}}$	One or two banks of 4K x 8 (like MCM4027's)
Z5Z4Z3Z2Z1Z0	$\overline{\text{RAS1}}$ (= Z7)	
Z6Z5Z4Z3Z2Z1Z0	$\overline{\text{RAS0}}$	One or two banks of 16K x 8 (like MCM4116's)
Z6Z5Z4Z3Z2Z1Z0	$\overline{\text{RAS1}}$ (= Z7)	
Z7Z6Z5Z4Z3Z2Z1Z0	$\overline{\text{RAS0}}$	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.)

Note: "Slow" (0.9 MHz) operation can be accomplished using 1.0 MHz MC6809E and MC6821 devices. For "Fast" (1.8 MHz) operation, 2.0 MHz MC68B09E and MC68B21 devices must be used.

Memory Size

Two bits (M1 and M0) determine RAM memory size. The options are:

SIZE	M1	M0
One or two banks of 4K × 1 dynamic RAMs	0	0
One or two banks of 16K × 1 dynamic RAMs	0	1
One bank of 64K × 1 dynamic RAMs	1	0
Up to 64K static RAM*	1	1

*Requires a latch for demultiplexing the RAM address.

IMPORTANT!

Note: Be sure to program the SAM for the correct memory size **before** using RAM (i.e., for a subroutine stack).

Map Type

One bit (TY) is used to select between two memory map configurations.

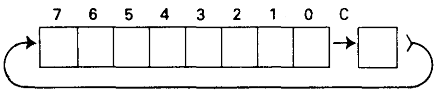
Refer to pages 17, 18 and 19 for details. Early versions of the SAM did not allow the "Fast" MPU rate to be used in conjunction with Map Type "TY = 1". Devices manufactured after January 1, 1983 allow both "Fast" and "Slow" MPU rates to be used with Map Type "TY = 1." (Date of manufacture is marked on devices as YYWW where YY is the year and WW is the week of manufacture.)

Writing To The SAM Control Register

Any bit in the control register (CR) may be set by writing to a specific unique address. Each bit has two unique addresses . . . writing to the **even** # address **clears** the bit and writing to the **odd** # address **sets** the bit. (Data on the data bus is irrelevant in this procedure.) The specific addresses are tabulated on pages 17 and 18.

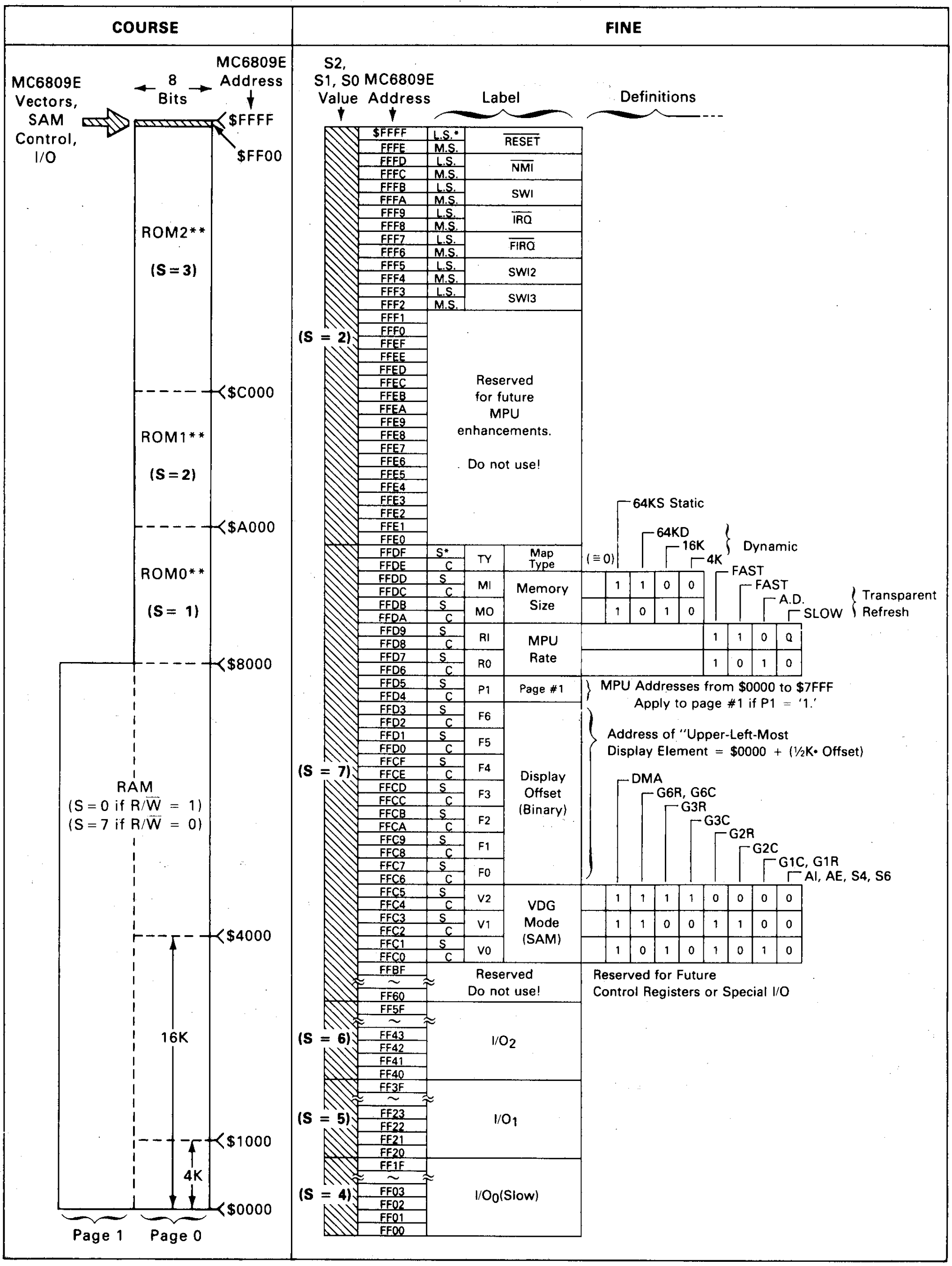
If desired, a short routine may be written to program the SAM CR "a word at a time". For example, the following routine copies "B" bits from "A" register to SAM CR addresses beginning with address "X".

SAM1	46		ROR	A
	24	06	BCC	SAM2
	30	01	INX	(LEAX1,X)
	A7	80	STA	0,X+
	20	02	BRA	SAM3
SAM2	A7	81	STA	0,X++
SAM3	5A		DEC	B
	26	F2	BNE	SAM1
	39		RTS	



3

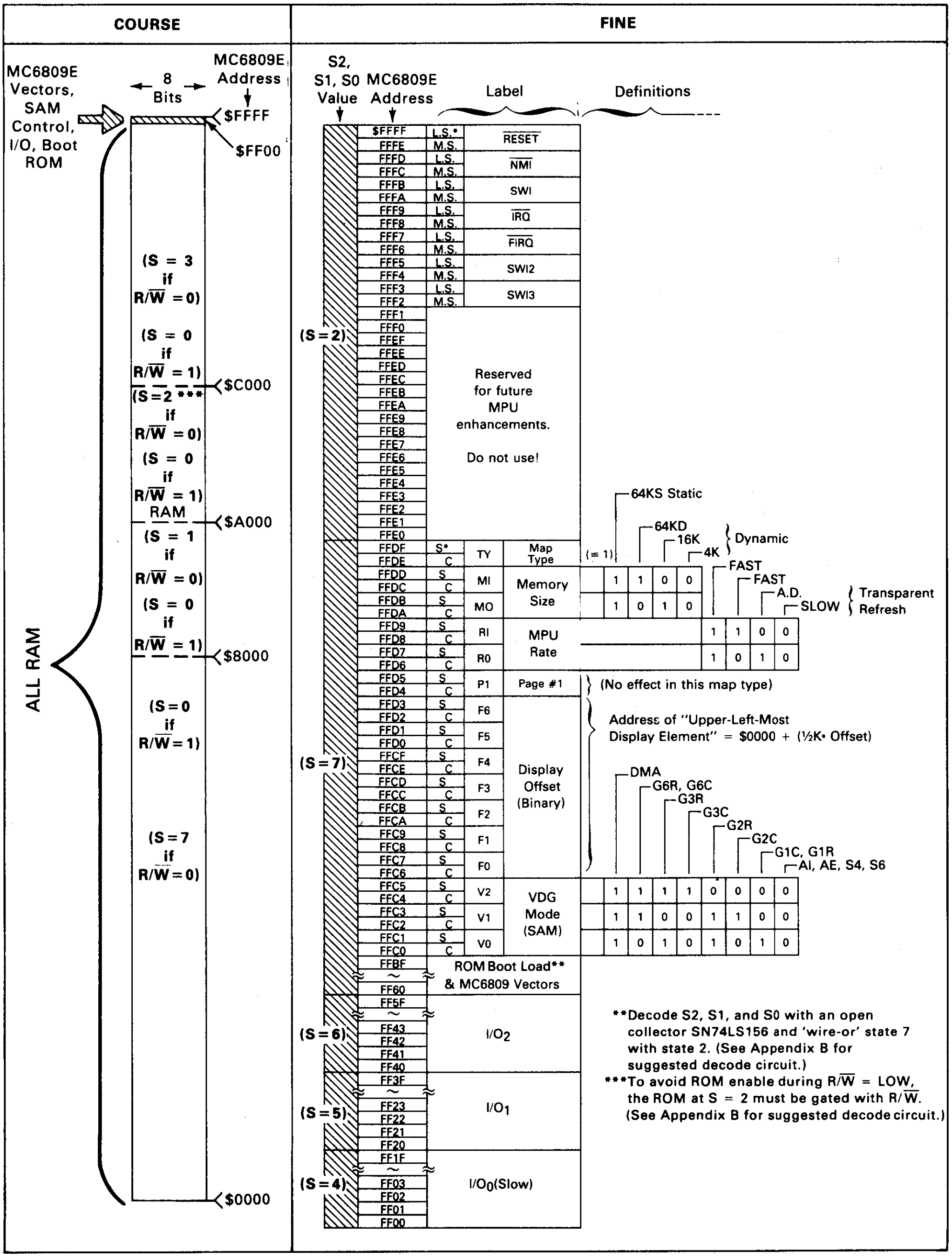
FIGURE 14 — MEMORY MAP (TYPE #0)



*Note:
M.S. ≡ Most Significant S ≡ Set Bit
L.S. ≡ Least Significant C ≡ Clear Bit
(All bits are cleared when SAM is reset.)
S = Device Select value = 4 x S2 + 2 x S1 + 1 x S0

**May also be RAM

FIGURE 15 — MEMORY MAP (TYPE #1)



*Note:

M.S. ≡ Most Significant   S ≡ Set Bit
L.S. ≡ Least Significant   C ≡ Clear Bit } (All bits are cleared when SAM is reset.)

S = Device Select value = 4 x S2 + 2 x S1 + 1 x S0

FIGURE 16 — MEMORY ALLOCATION TABLE
(Also, see the memory MAPs on pages 17 and 18.)

Type # 0: (Primarily for ROM based systems)

Address Range	S = 4(S2) + 2 (S1) + S0 S Value	Intended Use
$FFF2 to FFFF	2	MC6809E Vectors: $\overline{\text{Reset}}$, $\overline{\text{NMI}}$, SWI, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, SWI2, SWI3.
FFE0 to FFF1	2	Reserved for future MPU enhancements.
FFC0 to FFDF	7	SAM Control Register: V0, – V2, F0 – F6, P, R0, R1, M0, M1, TY.
FF60 to FFBF	7	Reserved for future control register enhancements.
FF40 to FF5F	6	I/O_2: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF20 to FF3F	5	I/O_1: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF00 to FF1F	4	I/O_0: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
C000 to FEFF	3	ROM2: 16K addresses. External cartridge ROM*.
A000 to BFFF	2	ROM1: 8K addresses. Internal ROM*. Note that MC6809E vector addresses select this ROM*.
8000 to 9FFF	1	ROM0: 8K addresses. Internal ROM*.
0000 to 7FFF	0 if R/$\overline{\text{W}}$ = 1 7 if R/$\overline{\text{W}}$ = 0	RAM: 32K addresses. RAM shared by MPU and VDG.

*Not restricted to ROM. For example, RAM or I/O may be used here.

Type # 1: (Primarily for RAM based systems)

Address Range	S = 4(S2) + 2 (S1) + S0 S Value	Intended Use
$FFF2 to FFFF	2	MC6809E Vectors: $\overline{\text{Reset}}$, $\overline{\text{NMI}}$, SWI, $\overline{\text{IRQ}}$, $\overline{\text{FIRQ}}$, SWI2, SWI3.
FFE0 to FFF1	2	Reserved for future MPU enhancements.
FFC0 to FFDF	7	SAM Control Register: V0 – V2, F0 – F_6, P, R0, R1, M0, M1, TY.
FF60 to FFBF	7	Small ROM: Boot load program and initial MC6809 vectors.
FF40 to FF5F	6	I/O_2: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0-A4.
FF20 to FF3F	5	I/O_1: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A0 – A4.
FF00 to FF1F	4	I/O_0: Input/Output (PIAs, ACIAs, etc.) To subdivide, use A2 – A4.
0000 to FEFF	0 if R/$\overline{\text{W}}$ = 1	RAM: 64K(– 256) addresses, shared by MPU and VDG. (If R/$\overline{\text{W}}$ = 0 then S = 3 for $C000–$FEFF; S = 2 for $A000–$BFFF; S = 1 for $8000–$9FFF and S = 7 for $0000–$7FFF.)

APPENDIX A

VDG/SAM Video Display System Offers 3 New Modes

by
Paul Fletcher

There are three new modes created when the VDG and SAM are used together in a video display system. These modes offer alphanumeric compatibility with 8 color low-to-high resolution graphics, 64Hx64V, 64Hx96V, 64Hx192V. The new modes S8, S12, and S24 are created by placing the VDG in the Alpha Internal mode and having the SAM in a 2K, 3K or 6K full color graphics mode. In all modes the VDG's $S/\overline{A}$ and Inv. pins are connected to data bits DD7 and DD6 to allow switching on the fly between Alpha and Semigraphics and between inverted and non-inverted alpha. This method is used in most VDG systems to obtain maximum flexibility.

The three modes divide the standard 8*12 dot box used by the VDG for the standard alpha and semigraphics modes into eight 4*3 dot boxes for the S8 mode, twelve 4*2 dot boxes for the S12 mode, and twenty-four 4*1 dot boxes for the S24 mode. Figure 17 shows the arrangement of these boxes. One byte is needed to control two horizontally consecutive boxes. It therefore takes four bytes for the S8, six bytes for the S12, and 12 bytes for the S24 mode to control the entire 8*12 dot box. These two horizontally consecutive boxes have four combinations of luminance controlled by bits B0 – B3. For convenience B2 should be made equal to B0 and B3 should be made equal to B1. This eliminates a screen placement problem which would cause other codes to change patterns when moved vertically on the screen. The illuminated boxes can be one of eight colors which are controlled by B4 – B6 (see Figure 18). The bytes needed to control all the boxes in the 8*12 dot box must be spaced 32 address spaces apart in the display RAM because of the addressing scheme orginally used in the VDG and duplicated by the SAM. This means to place an alphanumeric character on the TV screen it requires 4, 6, or 12 bytes depending on the mode used. These bytes are placed 32 memory locations apart in the display RAM (see Figure 18). This multiple byte format allows the mixing of character rows of different characters in the same 8*12 dot box creating new characters and symbols. It also allows overlining and underlining in eight colors by switching to semigraphics at the correct time.

These new modes optimize the memory versus screen density tradeoffs for RF performance on color TVs. This could make them the most versatile of all the modes depending on the users creativity and the software sophistication.

APPENDIX B

Memory Decode for "MAP TYPE = 1"

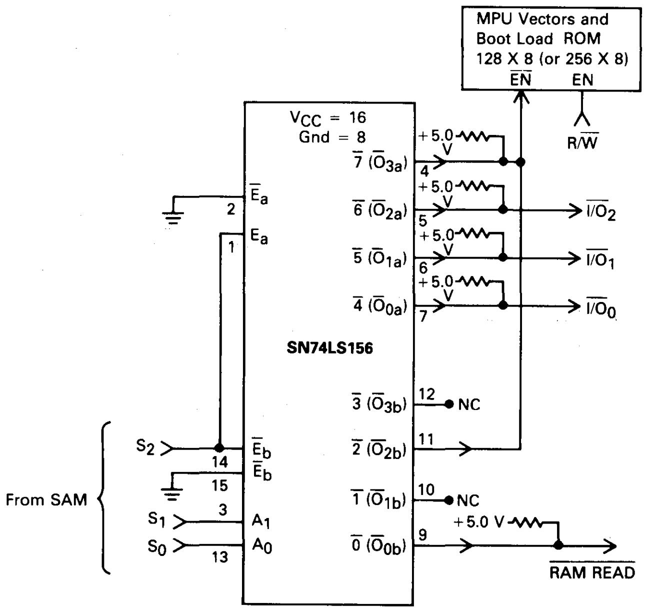
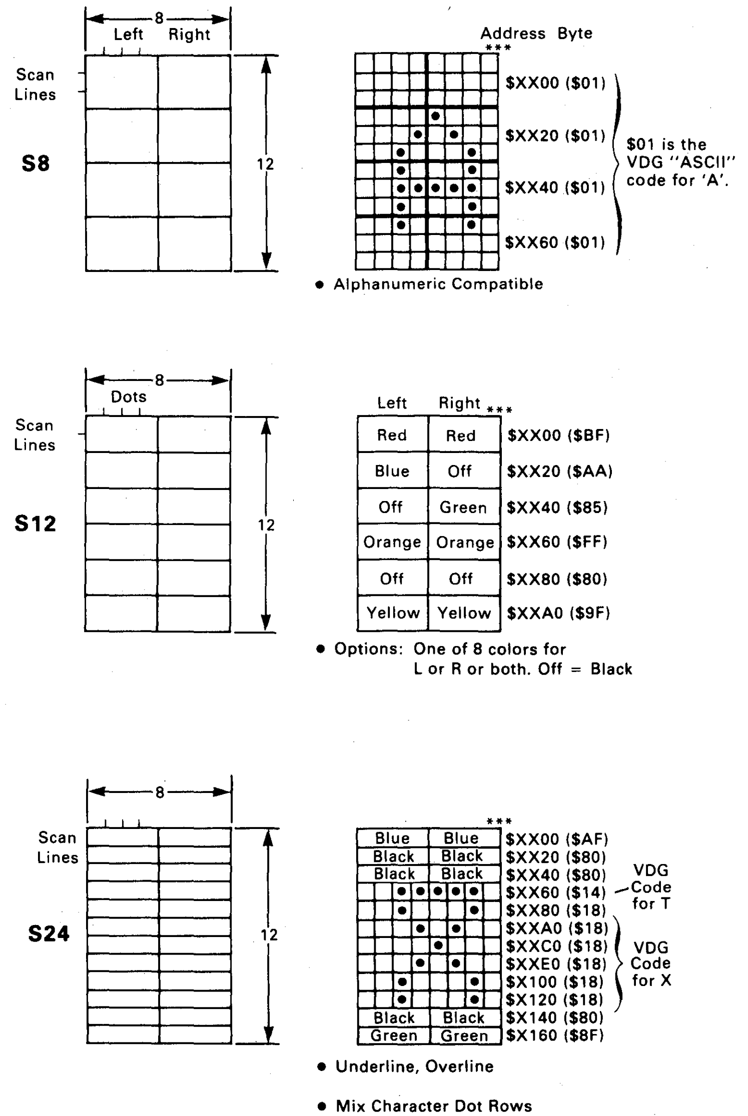
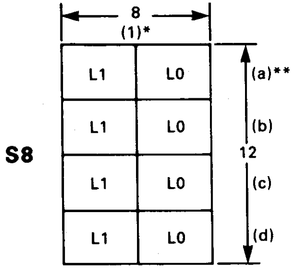


FIGURE 17 — DISPLAY MODES S8, S12, S24
Bit/Visible Dot Correlation



*** Characters will always remain in standard VDG positions.

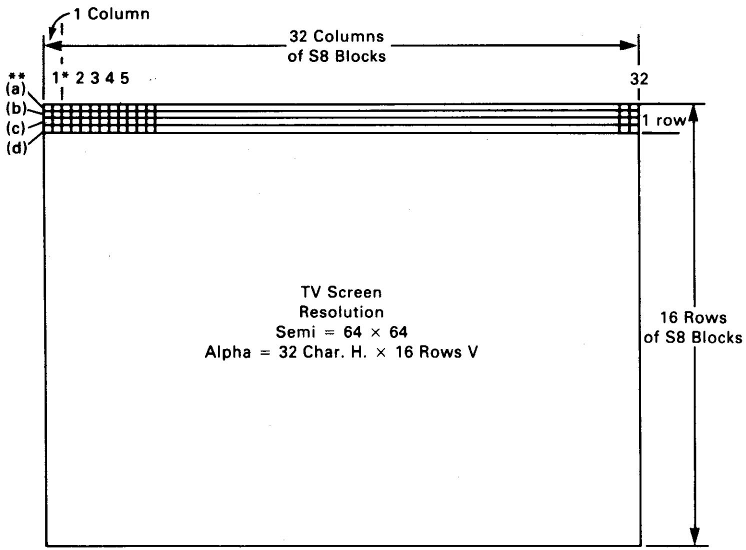
FIGURE 18 — S8 DISPLAY FORMAT EXAMPLES



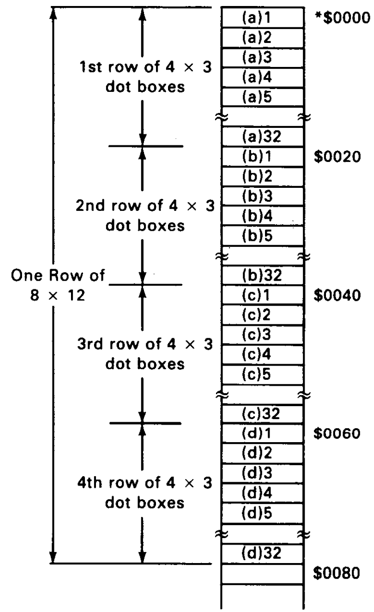
B7							B0	
X	C2	C1	C0	L3	L2	L1	L0	Semi
S/$\bar{A}$	Inv	X	X	X	X	X	X	Alpha
Extra		ASCII Code						

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	

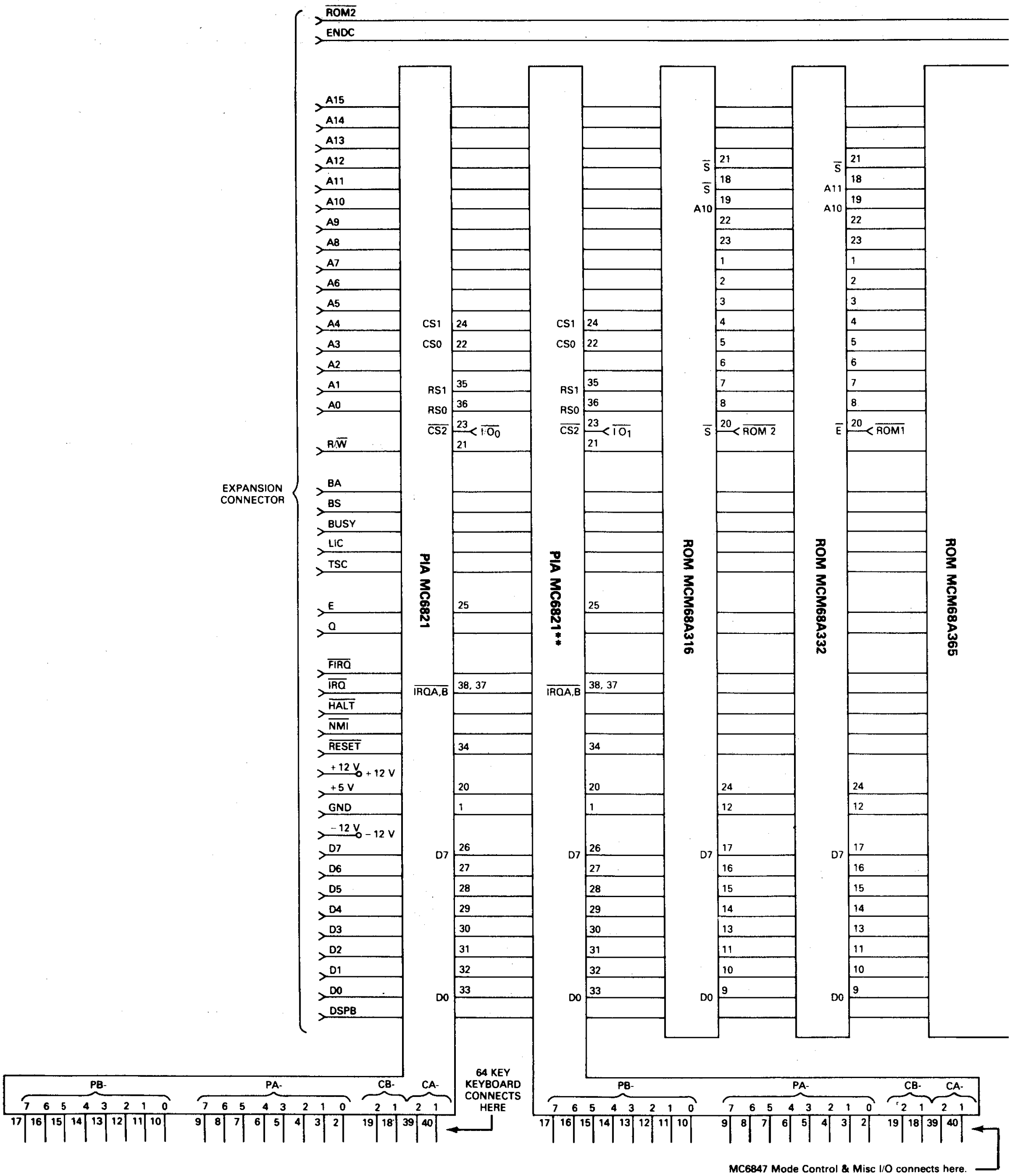


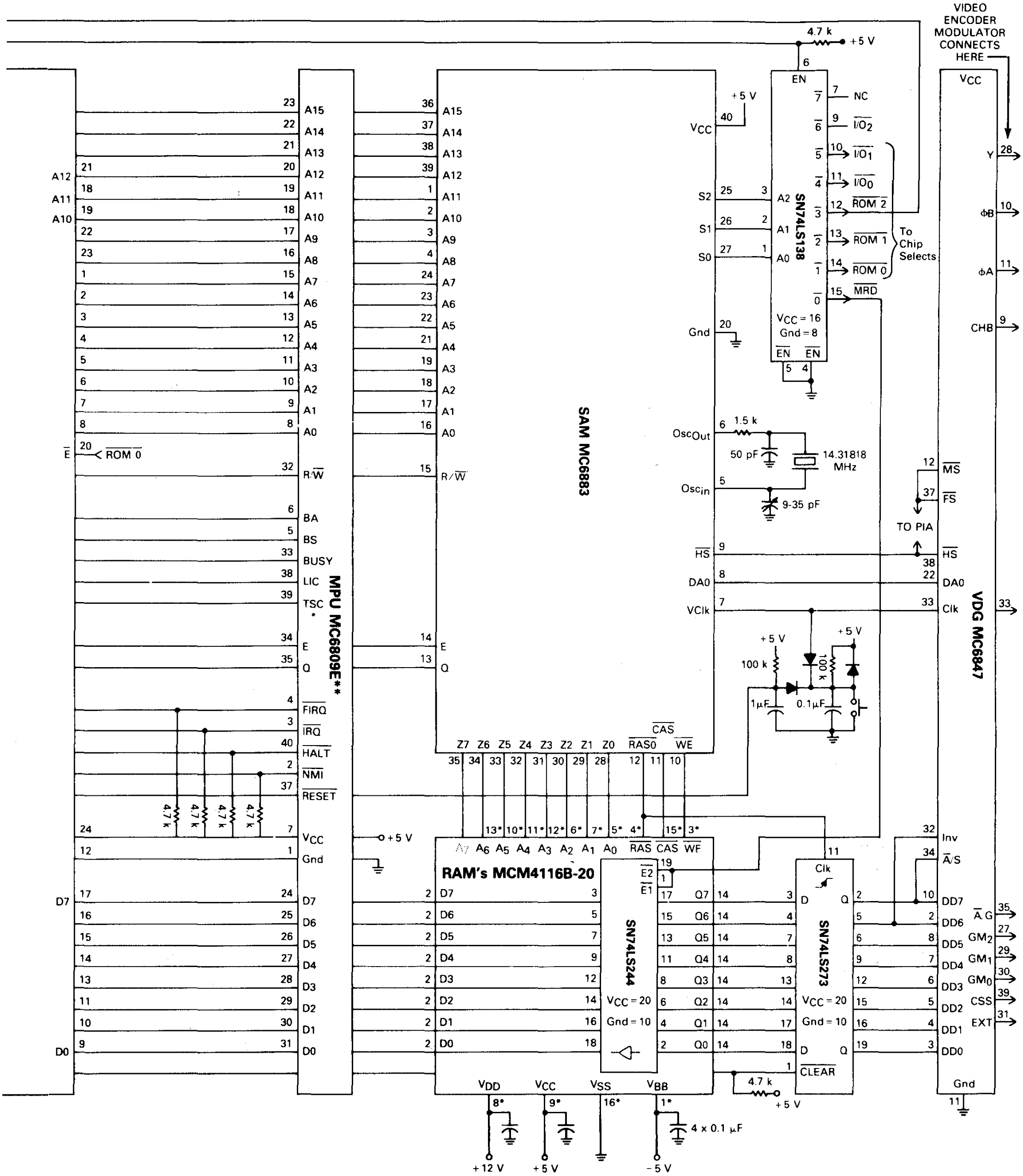
S8 Screen Memory Map



3

FIGURE 19 — EXAMPLE of MC6809E, MC6883 and MC6847 COMPUTER





*This pin number on 8 different RAM chips is connected to this point.
**See text . . . page 16

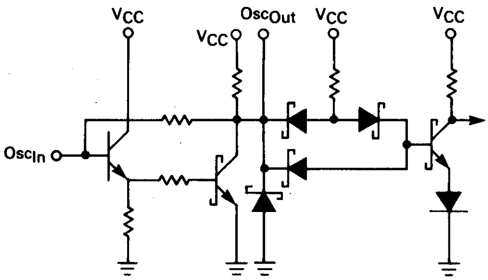


FIGURE 20 — EQUIVALENT OF OSCILLATOR INPUT AND OUTPUT

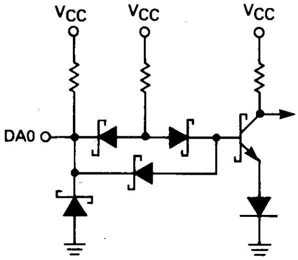


FIGURE 21 — DA0 INPUT

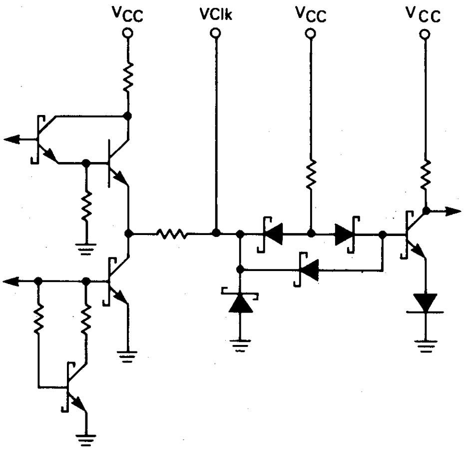


FIGURE 22 — VClk INPUT/OUTPUT

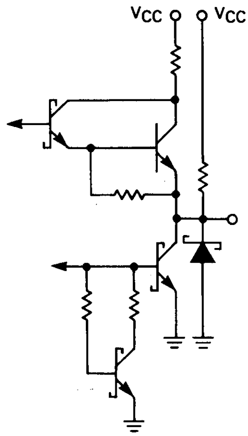


FIGURE 23 — E AND Q OUTPUTS

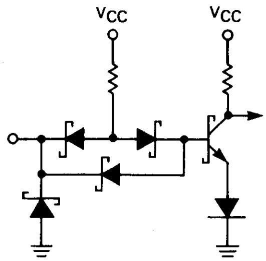


FIGURE 24 — TYPICAL INPUT

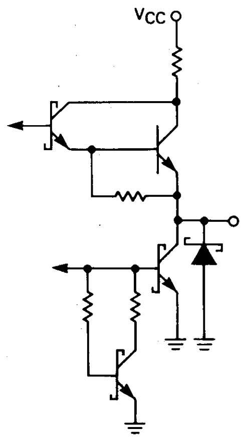


FIGURE 25 — TYPICAL OUTPUT

3

MC8T95/MC6885
MC8T96/MC6886
MC8T97/MC6887
MC8T98/MC6888

HEX THREE-STATE BUFFER INVERTERS

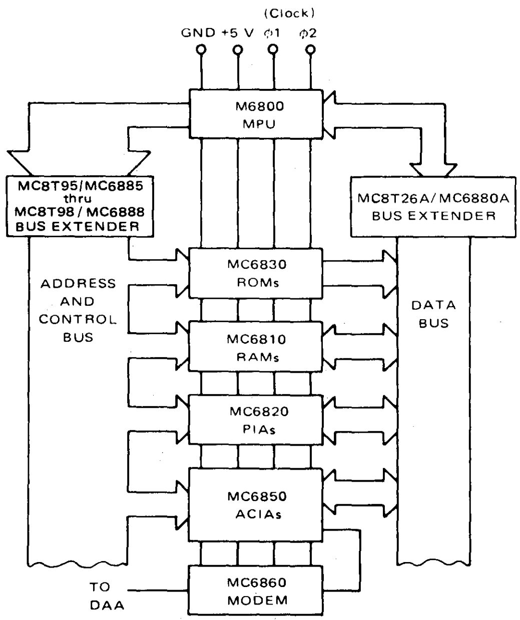
This series of devices combines three features usually found desirable in bus-oriented systems: 1) High impedance logic inputs insure that these devices do not seriously load the bus; 2) Three-state logic configuration allows buffers not being utilized to be effectively removed from the bus; 3) Schottky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

The devices differ in that the non-inverting MC8T95/MC6885 and inverting MC8T96/MC6886 provide a two-input Enable which controls all six buffers, while the non-inverting MC8T97/MC6887 and inverting MC8T98/MC6888 provide two Enable inputs – one controlling four buffers and the other controlling the remaining two buffers.

The units are well-suited for Address buffers on the M6800 or similar microprocessor application.

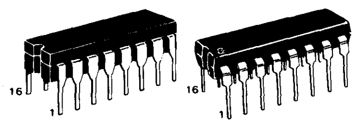
- High Speed – 8.0 ns (Typ)
- Three-State Logic Configuration
- Single +5 V Power Supply Requirement
- Compatible with 74LS Logic or M6800 Microprocessor Systems
- High Impedance PNP Inputs Assure Minimal Loading of the Bus

MICROPROCESSOR BUS EXTENDER APPLICATION



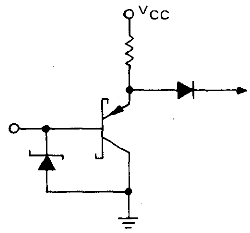
3

HEX THREE-STATE BUFFER/INVERTERS

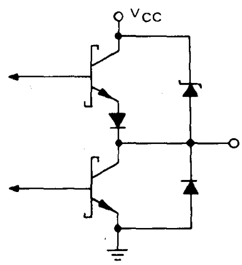


CASE 620 CASE 648

INPUT EQUIVALENT CIRCUIT



OUTPUT EQUIVALENT CIRCUIT



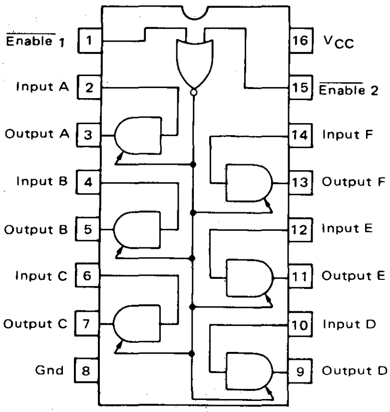
ORDERING INFORMATION

(Temperature Range for the following devices = 0 to +75°C)

DEVICE	ALTERNATE	PACKAGE
MC8T95L	MC6885L	Ceramic DIP
MC8T96L	MC6886L	Ceramic DIP
MC8T97L	MC6887L	Ceramic DIP
MC8T98L	MC6888L	Ceramic DIP
MC8T95P	MC6885P	Plastic DIP
MC8T96P	MC6886P	Plastic DIP
MC8T97P	MC6887P	Plastic DIP
MC8T98P	MC6888P	Plastic DIP

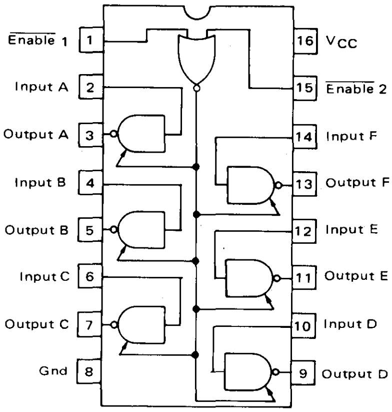
PIN CONNECTIONS AND TRUTH TABLES

MC8T95/MC6885



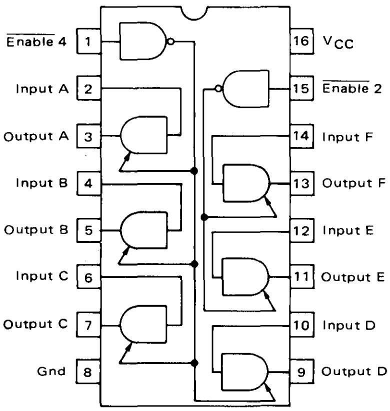
$\overline{\text{Enable}}$ 2	$\overline{\text{Enable}}$ 1	Input	Output
L	L	L	L
L	L	H	H
L	H	X	Z
H	L	X	Z
H	H	X	Z

MC8T96/MC6886



$\overline{\text{Enable}}$ 2	$\overline{\text{Enable}}$ 1	Input	Output
L	L	L	H
L	L	H	L
L	H	X	Z
H	L	X	Z
H	H	X	Z

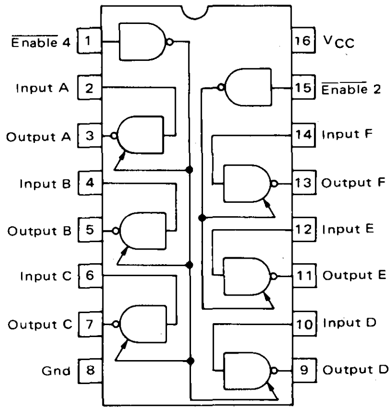
MC8T97/MC6887



$\overline{\text{Enable}}$	Input	Output
L	L	L
L	H	H
H	X	Z

L = Low Logic State
H = High Logic State
Z = Third (High Impedance) State
X = Irrelevant

MC8T98/MC6888



$\overline{\text{Enable}}$	Input	Output
L	L	H
L	H	L
H	X	Z

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

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Operating Junction Temperature	T_J		°C
Plastic Package		150	
Ceramic Package		175	

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, $0^oC \leqslant T_A \leqslant 75^oC$ and $4.75\ V \leqslant V_{CC} \leqslant 5.25\ V$)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Voltage – High Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IH}	2.0	–	–	V
Input Voltage – Low Logic State (V_{CC} = 4.75 V, T_A = 25°C)	V_{IL}	–	–	0.8	V
Input Current – High Logic State (V_{CC} = 5.25 V, V_{IH} = 2.4 V)	I_{IH}	–	–	40	µA
Input Current – Low Logic State (V_{CC} = 5.25 V, V_{IL} = 0.5 V, $V_{IL(\overline{E})}$ = 0.5 V)	I_{IL}	–	–	-400	µA
Input Current – High Impedance State (V_{CC} = 5.25 V, $V_{IL(I)}$ = 0.5 V, $V_{IH(\overline{E})}$ = 2.0 V)	$I_{IH(\overline{E})}$	–	–	-40	µA
Output Voltage – High Logic State (V_{CC} = 4.75 V, I_{OH} = -5.2 mA)	V_{OH}	2.4	–	–	V
Output Voltage – Low Logic State (I_{OL} = 48 mA)	V_{OL}	–	–	0.5	V
Output Current – High Impedance State	I_{OZ}				µA
(V_{CC} = 5.25 V, V_{OH} = 2.4 V)		–	–	40	
(V_{CC} = 5.25 V, V_{OL} = 0.5 V)		–	–	-40	
Output Short-Circuit Current (V_{CC} = 5.25 V, V_O = 0) (only one output can be shorted at a time)	I_{OS}	-40	-80	-115	mA
Power Supply Current (V_{CC} = 5.25 V)	I_{CC}				mA
MC8T95, MC8T97, MC6885, MC6887		–	65	98	
MC8T96, MC8T98, MC6886, MC6888		–	59	89	
Input Clamp Voltage (V_{CC} = 4.75 V, I_{IC} = -12 mA)	V_{IC}	–	–	-1.5	V
Output V_{CC} Clamp Voltage (V_{CC} = 0, I_{OC} = 12 mA)	V_{OC}	–	–	1.5	V
Output Gnd Clamp Voltage (V_{CC} = 0, I_{OC} = -12 mA)	V_{OC}	–	–	-1.5	V
Input Voltage (I_I = 1.0 mA)	V_I	5.5	–	–	V

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

Characteristic	Symbol	MC8T95/97 MC6885/87			MC8T96/98 MC6886/88			Unit
		Min	Typ	Max	Min	Typ	Max	
Propagation Delay Time – High to Low State	t_{PHL}							ns
(C_L = 50 pF)		3.0	–	12	4.0	–	11	
(C_L = 250 pF)		–	16	–	–	15	–	
(C_L = 375 pF)		–	20	–	–	18	–	
(C_L = 500 pF)		–	23	–	–	22	–	
Propagation Delay Time – Low to High State	t_{PLH}							ns
(C_L = 50 pF)		3.0	–	13	3.0	–	10	
(C_L = 250 pF)		–	25	–	–	22	–	
(C_L = 375 pF)		–	33	–	–	28	–	
(C_L = 500 pF)		–	42	–	–	35	–	
Transition Time – High to Low State	t_{THL}							ns
(C_L = 250 pF)		–	10	–	–	10	–	
(C_L = 375 pF)		–	11	–	–	13	–	
(C_L = 500 pF)		–	14	–	–	15	–	
Transition Time – Low to High State	t_{TLH}							ns
(C_L = 250 pF)		–	32	–	–	28	–	
(C_L = 375 pF)		–	42	–	–	38	–	
(C_L = 500 pF)		–	60	–	–	53	–	

3

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

Characteristic	Symbol	MC8T95/97 MC6885/87 Min	Typ	Max	MC8T96/98 MC6886/88 Min	Typ	Max	Unit
Propagation Delay Time – High State to Third State (C_L = 5.0 pF)	$t_{PHZ(\overline{E})}$	–	–	10	–	–	10	ns
Propagation Delay Time – Low State to Third State (C_L = 5.0 pF)	$t_{PLZ(\overline{E})}$	–	–	12	–	–	16	ns
Propagation Delay Time – Third State to High State (C_L = 50 pF)	$t_{PZH(\overline{E})}$	–	–	25	–	–	22	ns
Propagation Delay Time – Third State to Low State (C_L = 50 pF)	$t_{PZL(\overline{E})}$	–	–	25	–	–	24	ns

FIGURE 1 – TEST CIRCUIT FOR SWITCHING CHARACTERISTICS

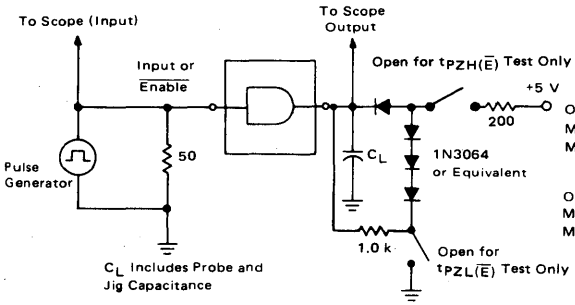


FIGURE 2 – WAVEFORMS FOR PROPAGATION DELAY TIMES INPUT TO OUTPUT

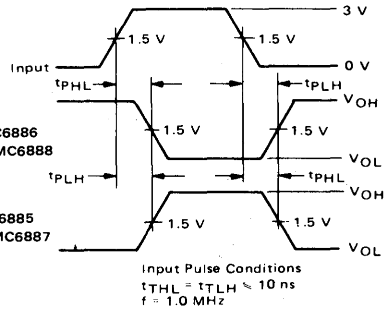
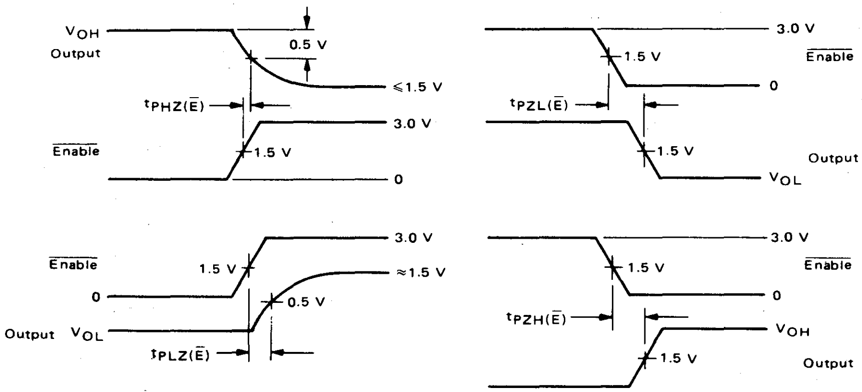
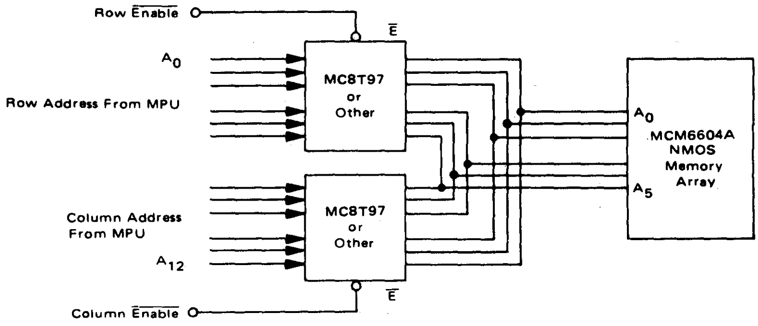


FIGURE 3 – WAVEFORMS FOR PROPAGATION DELAY TIMES – $\overline{\text{ENABLE}}$ TO OUTPUT



H = High-Logic State, L = Low-Logic State, Z = High Impedance State

FIGURE 4 – ADDRESS MULTIPLEXER FOR 16-PIN 4K NMOS MEMORY



3

MOTOROLA

MC8T28
MC6889

NONINVERTING QUAD THREE-STATE BUS TRANSCEIVER

This quad three-state bus transceiver features both excellent MOS or MPU compatibility, due to its high impedance PNP transistor input, and high-speed operation made possible by the use of Schottky diode clamping. Both the -48 mA driver and -20 mA receiver outputs are short-circuit protected and employ three-state enabling inputs.

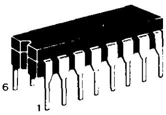
The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μA at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier diode clamps to suppress excessive undershoot voltages.

Propagation delay times for the driver portion are 17 ns maximum while the receiver portion runs 17 ns. The MC8T28 is identical to the NE8T28 and it operates from a single +5 V supply.

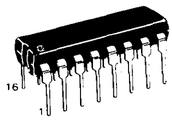
- High Impedance Inputs
- Single Power Supply
- High Speed Schottky Technology
- Three-State Drivers and Receivers
- Compatible with M6800 Family Microprocessor
- Non-Inverting

NONINVERTING BUS TRANSCEIVER

MONOLITHIC SCHOTTKY INTEGRATED CIRCUITS

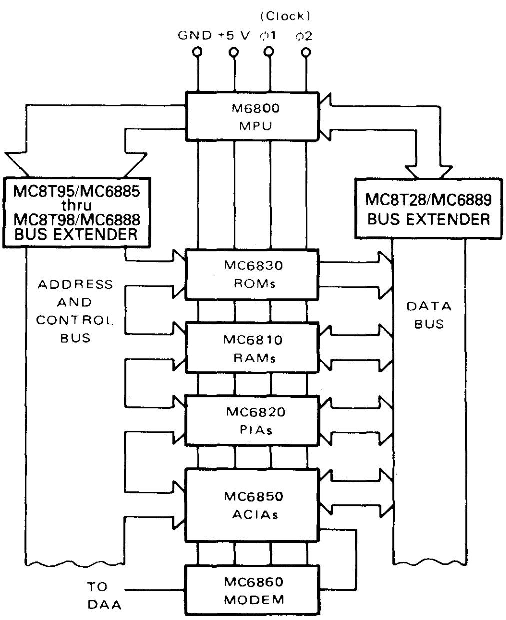


L SUFFIX
CERAMIC PACKAGE
CASE 620-02

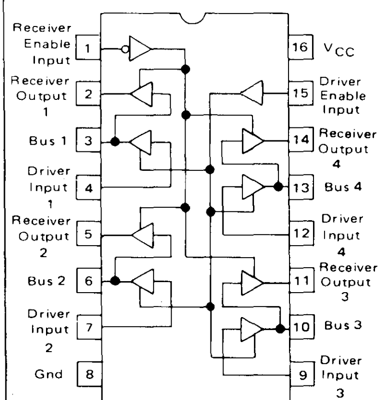


P SUFFIX
PLASTIC PACKAGE
CASE 648-05

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS — MC8T28 MC6889



ORDERING INFORMATION			
Device	**Alternate**	**Temperature Range**	**Package**
MC8T28L	MC6889L	0 to +75°C	Ceramic DIP
MC8T28P	MC6889P	0 to +75°C	Plastic DIP

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

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	8.0	Vdc
Input Voltage	V_I	5.5	Vdc
Junction Temperature Ceramic Package Plastic Package	T_J	 175 150	°C
Operating Ambient Temperature Range	T_A	0 to +75	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (4.75 V ≤ V_{CC} ≤ 5.25 V and 0°C ≤ T_A ≤ 75°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Current – Low Logic State (Receiver Enable Input, $V_{IL(\overline{RE})}$ = 0.4 V) (Driver Enable Input, $V_{IL(DE)}$ = 0.4 V) (Driver Input, $V_{IL(D)}$ = 0.4 V) (Bus (Receiver) Input, $V_{IL(B)}$ = 0.4 V)	 $I_{IL(\overline{RE})}$ $I_{IL(DE)}$ $I_{IL(D)}$ $I_{IL(B)}$			 -200 -200 -200 -200	μA
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) (Driver Enable Input, $V_{IH(DE)}$ = 5.25 V) (Driver Input, $V_{IH(D)}$ = 5.25 V)	 $I_{IH(\overline{RE})}$ $I_{IH(DE)}$ $I_{IH(D)}$			 25 25 25	μA
Input Voltage – Low Logic State (Receiver Enable Input) (Driver Enable Input (Driver Input) (Receiver Input)	 $V_{IL(\overline{RE})}$ $V_{IL(DE)}$ $V_{IL(D)}$ $V_{IL(B)}$	–	–	 0.85 0.85 0.85 0.85	V
Input Voltage – High Logic State (Receiver Enable Input) (Driver Enable Input) (Driver Input) (Receiver Input)	 $V_{IH(\overline{RE})}$ $V_{IH(DE)}$ $V_{IH(D)}$ $V_{IH(B)}$	 2.0 2.0 2.0 2.0	–	–	V
Output Voltage – Low Logic State (Bus Driver) Output, $I_{OL(B)}$ = 48 mA) (Receiver Output, $I_{OL(R)}$ = 20 mA)	 $V_{OL(B)}$ $V_{OL(R)}$	 – –	 – –	 0.5 0.5	V
Output Voltage – High Logic State (Bus (Driver) Output, $I_{OH(B)}$ = -10 mA) (Receiver Output, $I_{OH(R)}$ = -2.0 mA) (Receiver Output, $I_{OH(R)}$ = −100 μA, V_{CC} = 5.0 V)	 $V_{OH(B)}$ $V_{OH(R)}$	 2.4 2.4 3.5	 3.1 3.1 –	 – – –	V
Output Disabled Leakage Current – High Logic State (Bus Driver) Output, $V_{OH(B)}$ = 2.4 V) (Receiver Output, $V_{OH(R)}$ = 2.4 V)	 $I_{OHL(B)}$ $I_{OHL(R)}$	 – –	 – –	 100 100	μA
Output Disabled Leakage Current – Low Logic State (Bus Output, $V_{OL(B)}$ = 0.5 V) (Receiver Output, $V_{OL(R)}$ = 0.5 V)	 $I_{OLL(B)}$ $I_{OLL(R)}$	 – –	 – –	 -100 -100	μA
Input Clamp Voltage (Driver Enable Input $I_{ID(DE)}$ = −12 mA) (Receiver Enable Input $I_{IC(RE)}$ = +12 mA) (Driver Input $I_{IC(D)}$ = −12 mA)	 $V_{IC(DE)}$ $V_{IC(RE)}$ $V_{IC(D)}$	 – – –	 – – –	 -1.0 -1.0 -1.0	V
Output Short-Circuit Current, V_{CC} = 5.25 V (1) (Bus (Driver) Output) (Receiver Output)	 $I_{OS(B)}$ $I_{OS(R)}$	 -50 -30	 – –	 -150 -75	mA
Power Supply Current (V_{CC} = 5.25 V)	I_{CC}	–	–	110	mA

(1) Only one output may be short-circuited at a time.

3

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

Characteristic	Symbol	Min	Max	Unit
Propagation Delay Time—Receiver (C_L = 30 pF)	$t_{PLH(R)}$ $t_{PHL(R)}$	–	17 17	ns
Propagation Delay Time—Driver (C_L = 300 pF)	$t_{PLH(D)}$ $t_{PHL(D)}$	–	17 17	ns
Propagation Delay Time—Enable (C_L = 30 pF) – Receiver	$t_{PZL(R)}$ $t_{PLZ(R)}$	– –	23 18	ns
– Driver Enable (C_L 300 pF)	$t_{PZL(D)}$ $t_{PLZ(D)}$	– –	28 23	

3

FIGURE 1 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH(R)}$ AND $t_{PHL(R)}$

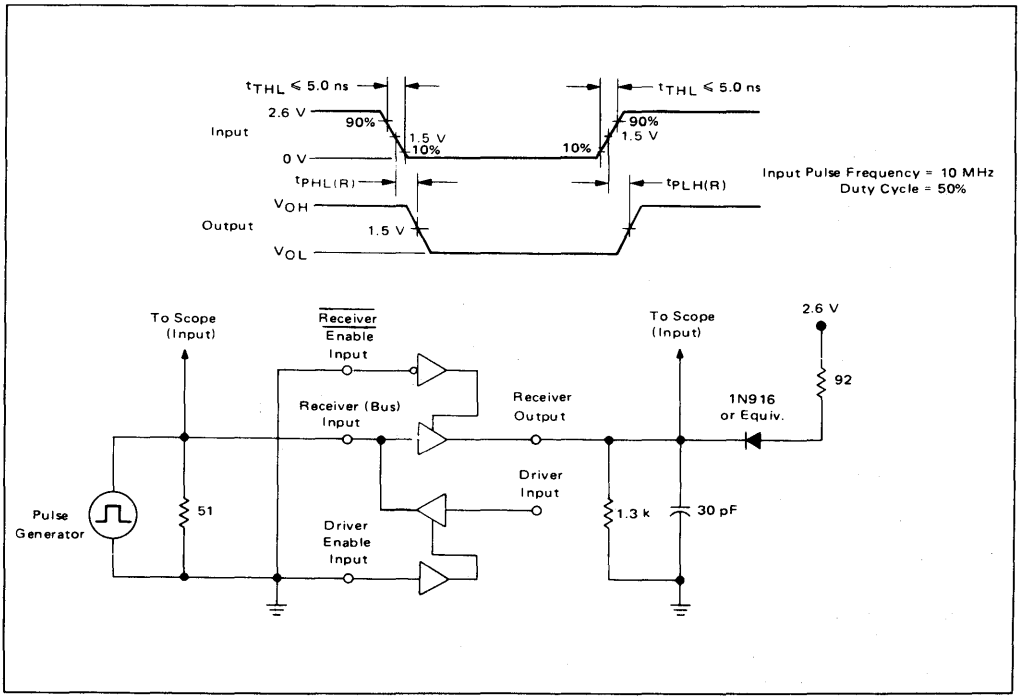


FIGURE 2 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

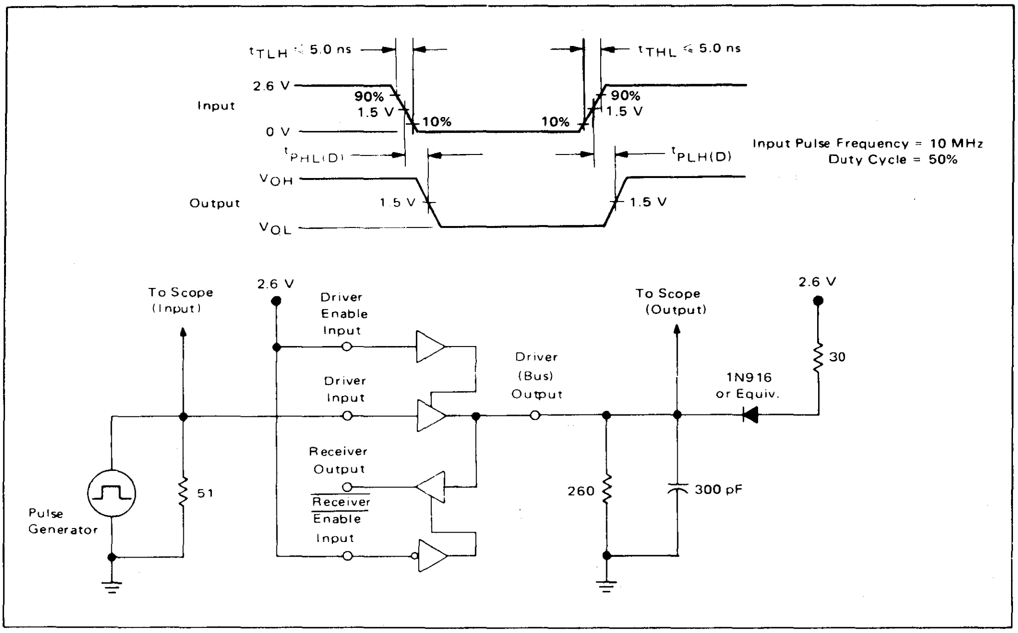
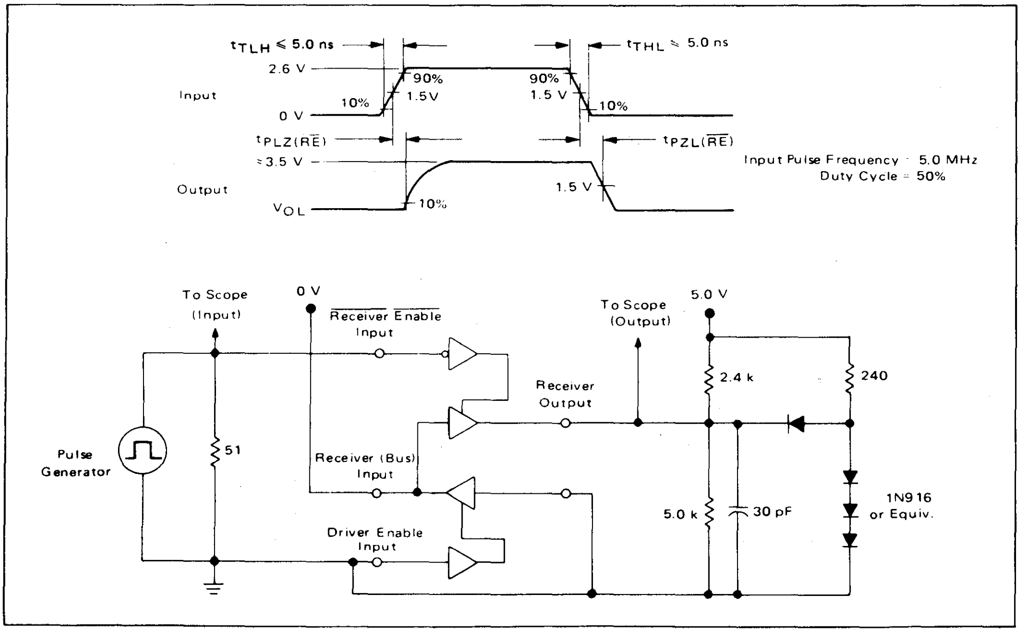


FIGURE 3 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$



3

FIGURE 4 – TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PZL(DE)}$

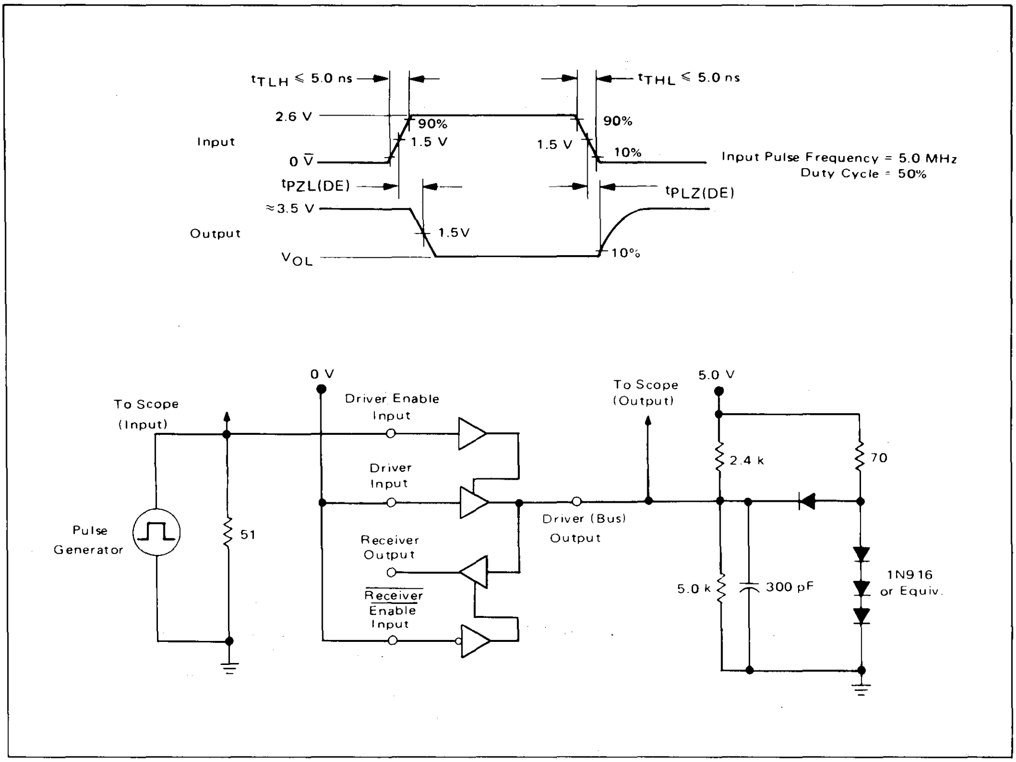
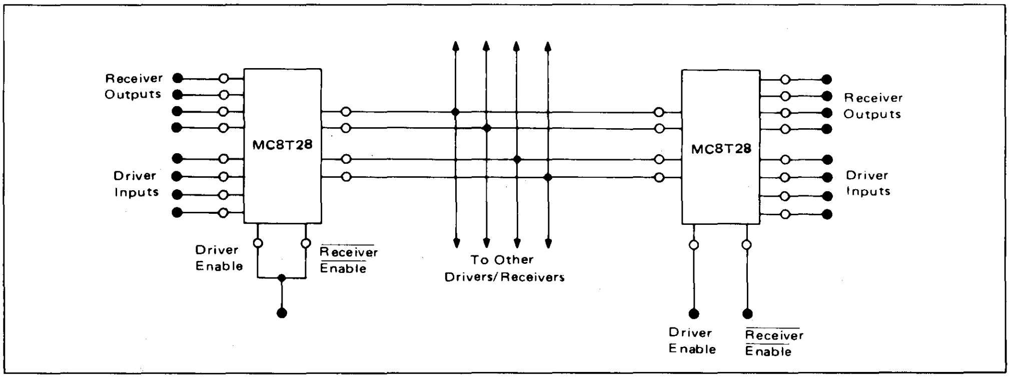


FIGURE 5 – BIDIRECTIONAL BUS APPLICATIONS



MC6890

Advance Information

MPU-BUS-COMPATIBLE 8-BIT D-TO-A CONVERTER

The MC6890 is a self-contained, bus-compatible, 8 bit (±0.19% accuracy) D-to-A converter system capable of interfacing directly with 8-bit microprocessors.

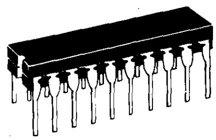
Available in both commercial and military temperature ranges, this monolithic converter contains master/slave registers to prevent transparency to data transitions during active enable; a laser-trimmed, low-TC, 2.5 V precision bandgap reference; and high stability, laser-trimmed, thin-film resistors for both reference input and output span and bipolar offset control.

A reset pin provides for overriding stored data and forcing I_{out} to zero.

- Direct Data Bus Link with All Popular TTL Level MPU's
- ±1/2 LSB Nonlinearity Over Temperature
- Fast Settling Time: 200 ns Typ
- Internal 2.5-V Precision Laser-Trimmed Voltage Reference (May Also Be Used Externally)
- Minimum $\overline{\text{Enable}}$ Pulse Width: 70 ns
- Fast $\overline{\text{Enable}}$: 10 ns Maximum Data Hold Time
- $\overline{\text{Reset}}$ Pin to Override Data
- Output Voltage Ranges: +5, +10, +20, or ±2.5, ±5, ±10 Volts
- Low Power: 90 mW Typ
- +5 V and -5 V to -15 V Supplies

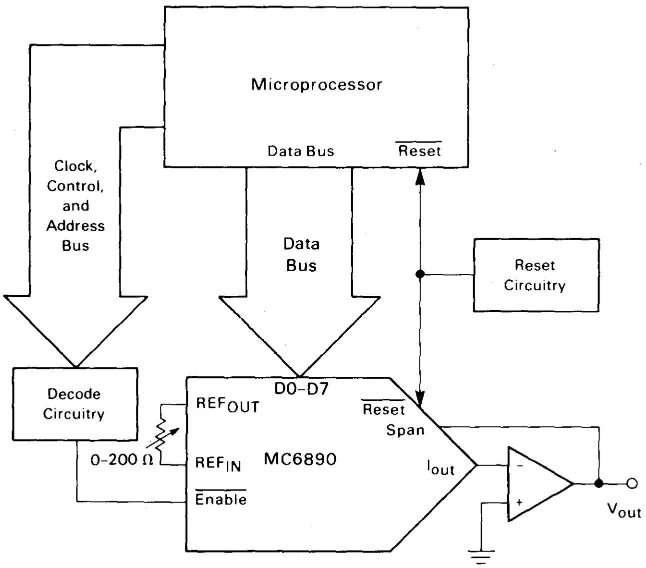
8-BIT MPU-BUS-COMPATIBLE DAC

SILICON MONOLITHIC INTEGRATED CIRCUIT

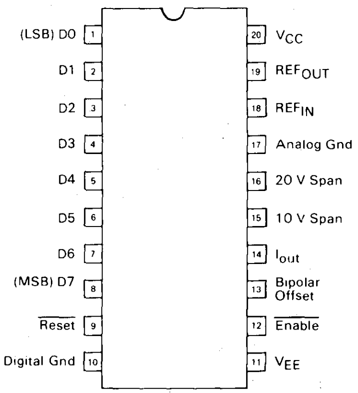


L SUFFIX
CASE 732-03

OPERATION WITH AN MPU



PIN CONNECTIONS



ORDERING INFORMATION

Device	Temperature Range	Package
MC6890L	0° to +70°C	Ceramic DIP
MC6890AL	-55° to +125°C	Ceramic DIP

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

3

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC} V_{EE}	+7.0 -18	Vdc
Digital Input Voltage, Pins 1-8, 12 Pin 9	V_{in}	-3.0 to +7.0 0 to +7.0	Vdc
Applied Output Voltage	V_{14}	V_{EE} +2.0 to V_{EE} +24	Vdc
Reference Amplifier Input	V_{18}	±7.5	Vdc
Operating Temperature Range MC6890L, MC6890AL	T_A	 0 to +70 -55 to +125	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Junction Temperature	T_J	+150	°C

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V, V_{EE} = -12 V, Pin 18 loaded only by Pin 19 through 100 Ω. Reset high, $T_A = T_{low}$ to T_{high}[1], unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Digital Input Logic Levels High Level, Logic 1 Low Level, Logic 0	 V_{IH} V_{IL}	 2.0 —	 — —	 — 0.8	Vdc
Digital Input Current Data (V_{IH} = 3.0 V) (V_{IL} = 0.4 V) Enable (V_{IH} = 3.0 V) (V_{IL} = 0.4 V) Reset (V_{IH} = V_{CC}) (V_{IL} = 0.4 V)	 I_{IH} I_{IL} I_{IH} I_{IL} I_{IH} I_{IL}	 — — — — — —	 0.001 0.5 0.001 -6.5 0.001 -1.0	 1.0 -10 1.0 -100 1.0 -15	 μA μA μA μA μA μA
Full Scale Output Current — Unipolar	I_O	-1.50	-1.992	-2.50	mA
Unipolar Zero Output — All Bits Off (T_A = 25°C)	—	—	0.010	0.20	μA
Output Voltage Temperature Coefficient Unipolar Zero Bipolar Zero Full Scale Range	TC_{VO}	 — — —	 ±1.0 ±5.0 ±20	 ±2.0 ±15 ±50	ppm of FSR/°C
Output Voltage, Full Scale Range (See Figure 3) (T_A = 25°C) (10 V Span) (20 V Span) (5.0 V Span)	V_O	 9.861 19.722 4.930	 9.961 19.922 4.980	 10.061 20.122 5.030	Vdc
Output Voltage, Bipolar Zero (MSB on) (See Figure 4) (T_A = 25°C) (10 V Span) (20 V Span) (5.0 V Span)	V_O	 — — —	 0 0 0	 ±20 ±40 ±10	mV
DAC Output Resistance — Exclusive of Span Resistors (T_A = 25°C) (See Figure 5)	R_O	1.0	5.0	—	MΩ
Resolution	—	8.0	8.0	8.0	Bits
Nonlinearity — Relative Accuracy (See Terminology)	NL	—	—	±0.19 (±1/2 LSB)	%
Differential Nonlinearity	Monotonicity Guaranteed				
Differential Nonlinearity (T_A = 25°C) (See Terminology)	—	—	—	±0.29 (±3/4 LSB)	%
Reference Input Resistor	R_{REF}	3800	4900	6800	Ω
Reference Output Voltage (T_A = 25°C)	V_{REF}	2.470	2.500	2.530	Vdc
Reference Output Impedance (T_A = 25°C) I_{load} = 0-3.0 mA	—	—	0.3	1.0	Ω
Reference Short Circuit Current (T_A = 25°C)	I_{REF}	15	30	50	mA
Reference Output Voltage Temperature Coefficient	$TC_{VO(REF)}$	—	±20	—	ppm/°C
Power Supply Range	V_{CC} V_{EE}	4.5 -16.5	5.0 -12	5.5 -4.5	Vdc
Power Supply Current — All Bits Low (V_{CC} = 5.0 V) (V_{EE} = -5.0 V) (V_{EE} = -15 V)	 I_{CC} I_{EE} I_{EE}	 — — —	 10 -10 -10	 20 -15 -15	mA
Power Supply Rejection (T_A = 25°C) To V_{CC} (V_{CC} = 4.5 to 5.5 V) To V_{EE} (V_{EE} = -4.5 V to -16.5 V)	PSR	 — —	 0.010 0.10	 ±1/10 ±1/2	LSB
Power Dissipation — All Bits Low For V_{CC} = 4.5 V, V_{EE} = -4.5 V For V_{CC} = 5.5 V, V_{EE} = -16.5 V	P_D	 — —	 90 220	 158 358	mW

NOTE 1: T_{low} = -55°C for MC6890A, 0° for MC6890
T_{high} = +125°C for MC6890A, +70°C for MC6890

AC SPECIFICATIONS (V_{CC} = 5.0 V, V_{EE} = -12 V, T_A = 25°C unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
Current Settling Time (Enable Positive Edge to ±1/2 LSB Output)	t_s	—	200	300*	ns
Data Setup Time	$t_{su(D)}$	70	40	—	ns
Data Hold Time	$t_{h(D)}$	10	0	—	ns
Pulse Widths					ns
$\overline{\text{Enable}}$	$t_{W(\overline{E})}$	70	20	—	
$\overline{\text{Reset}}$	$t_{W(\overline{R})}$	100*	—	—	
Propagation Delays					ns
$\overline{\text{Enable}}$, Low to High	$t_{PLH(\overline{E})}$	—	100	—	
Reset, High to Low ($I_O < 1.0\ \mu A$)	$t_{PHL(\overline{R})}$	—	250	—	

*Not 100% tested, guaranteed by design

FIGURE 1 — TIMING DIAGRAM

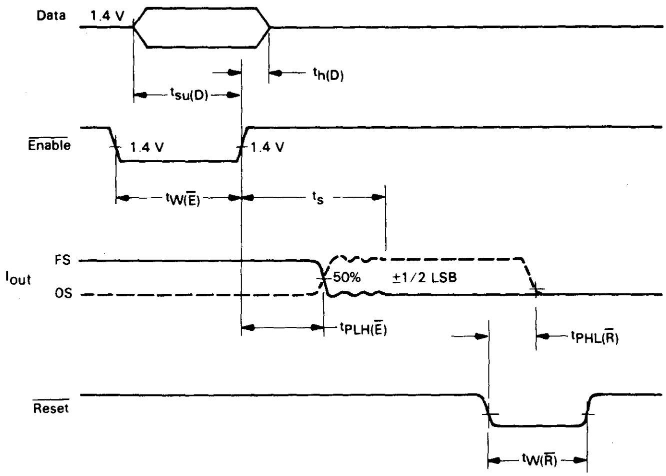
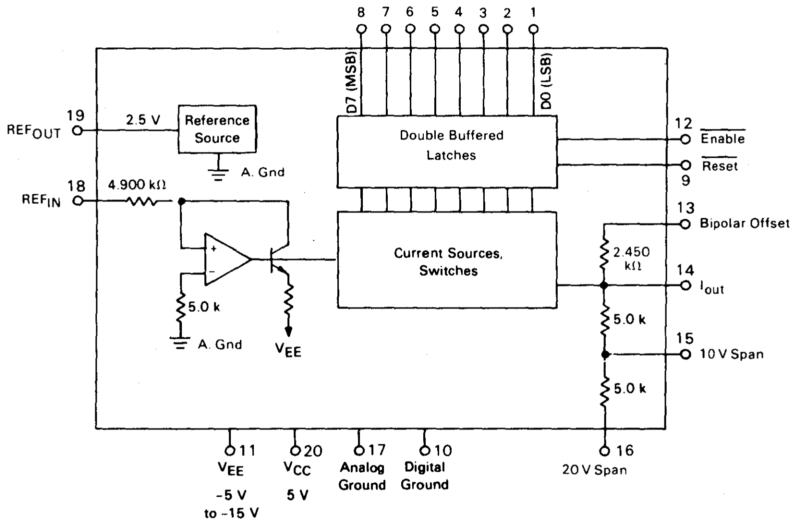


FIGURE 2 — BLOCK DIAGRAM

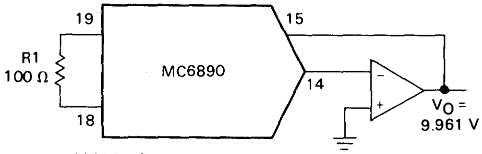


3

TEST FIGURES

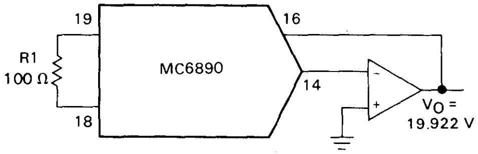
UNIPOLAR CONFIGURATIONS

FIGURE 3A



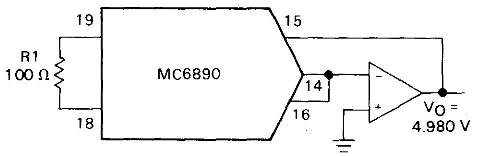
+10 V Configuration
Latched Input Code: 11111111

FIGURE 3B



+20 V Configuration
Latched Input Code: 11111111

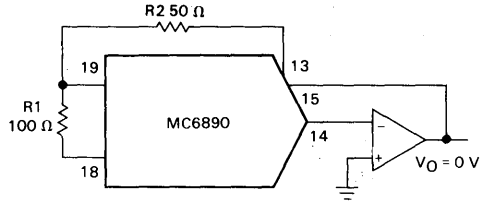
FIGURE 3C



+5.0 V Configuration
Latched Input Code: 11111111

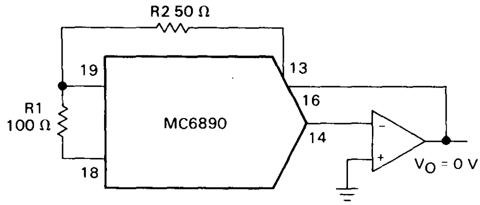
BIPOLAR CONFIGURATIONS

FIGURE 4A



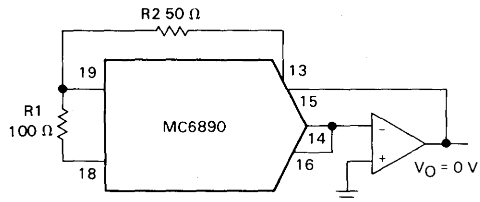
±5.0 V Configuration
Latched Input Code: 10000000

FIGURE 4B



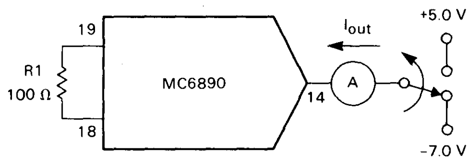
±10 V Configuration
Latched Input Code: 10000000

FIGURE 4C



±2.5 V Configuration
Latched Input Code: 10000000

FIGURE 5 TEST CONFIGURATION FOR DAC OUTPUT IMPEDANCE



Latched Input Code: 11111111

$$R_{out} = \frac{12\ V}{\Delta I_{out}}$$

TERMINOLOGY

Nonlinearity (Relative Accuracy) — Maximum output deviation from ideal straight line connecting zero and full-scale readings, expressed as a fraction of LSB or percent of full scale.

Differential Nonlinearity — Maximum deviation in the readings of any two adjacent input bit codes from the ideal LSB step, expressed in fractions of LSB or percentage of full scale. A differential nonlinearity value greater than 1 LSB may lead to non-monotonic operation.

Monotonicity — For every increase in the input digital word, the output current either remains the same or increases. The MC6890 is guaranteed to be monotonic over temperature.

Settling Time — The elapsed time from the $\overline{\text{Enable}}$ positive transition until the output has settled within an error band about its final value.

The worst case switching condition occurs when all bits are latched "on," which corresponds to a low-to-high transition for all bits. This time is typically 200 ns for the current output to settle to within ±1/2 LSB for 8 bit accuracy. These times apply when the output swing is limited to a small (<0.5 V) swing and the external output capacitance is under 10 pF.

Gain Error — The difference between the actual full scale range and the ideal full scale range. Based on a 0 to 10 V output configuration, the ideal FSR is $\frac{255}{256} \times 10\text{ V} = 9.961\text{ V}$.

Gain error is laser trimmed to less than ±1.0% with R1 = 100 Ω (Figure 3) and can be user trimmed to zero error with R1 = 200 Ω pot.

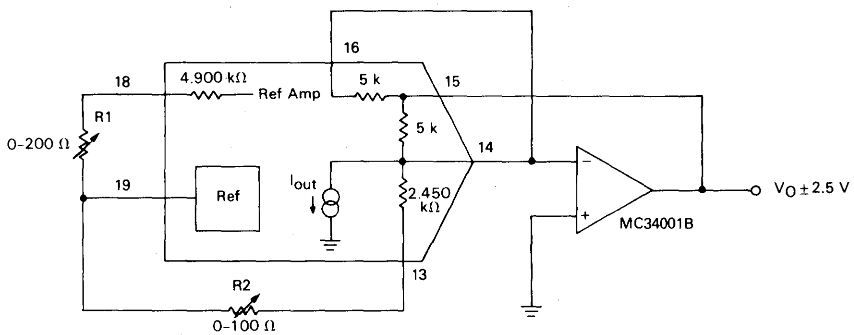
Bipolar Zero — Using the configuration shown in Figure 6 with R1 = 100 Ω, R2 = 50 Ω, with the MSB on and all other bits off, the output voltage reading compared to analog ground is expressed as a percentage of the full-scale range. Offset voltage of the output op amp must be nulled. Bipolar Zero error is laser trimmed to less than 0.20% and can be user trimmed to zero with R2 = 100 Ω pot.

Temperature Coefficients — (Unipolar zero, Bipolar zero, Gain and Reference Output). The maximum deviation of the particular parameter over the specified temperature range, divided by the temperature range, expressed in parts per million of Full Scale Range per degree C.

Power Supply Rejection — The change in full scale current caused by the specified change in V_{EE} or V_{CC} is expressed in LSB's.

$\overline{\text{Reset}}$ Function — The MC6890 has a $\overline{\text{Reset}}$ pin (9) that will force the DAC's registers, and therefore the DAC output current, to zero. This input is active low and should not occur simultaneously with an active $\overline{\text{Enable}}$ signal although no harm would result to the converter. The power dissipation increases slightly during $\overline{\text{Reset}}$ low. $\overline{\text{Reset}}$ should not be allowed to become more negative than ground.

FIGURE 6 — MC6890 IN TYPICAL BIPOLAR ±2.5 V OPERATION

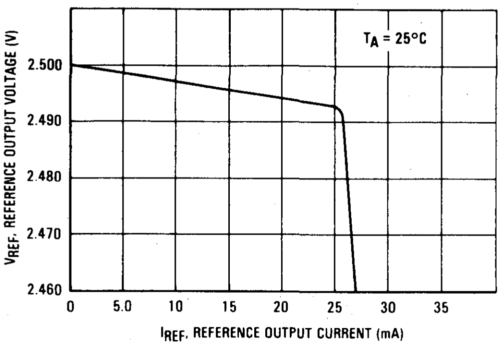


D7	D6	D5	D4	D3	D2	D1	D0	V_O (Volts)	
								R2 ≅ 60 Ω	R2 ≅ 50 Ω
1	1	1	1	1	1	1	1	+2.490	+2.480
1	1	1	1	1	1	1	0	+2.470	+2.460
1	0	0	0	0	0	0	0	+0.010	+0.000
0	1	1	1	1	1	1	1	−0.010	−0.020
0	0	0	0	0	0	0	1	−2.470	−2.480
0	0	0	0	0	0	0	0	−2.490	−2.500

3

TYPICAL PERFORMANCE CURVES

FIGURE 7 — REFERENCE VOLTAGE versus EXTERNAL LOAD CURRENT*



*External load current is in addition to Reference Input Current (Pin 18) of D/A converter.

FIGURE 8 — DIGITAL INPUT CHARACTERISTICS

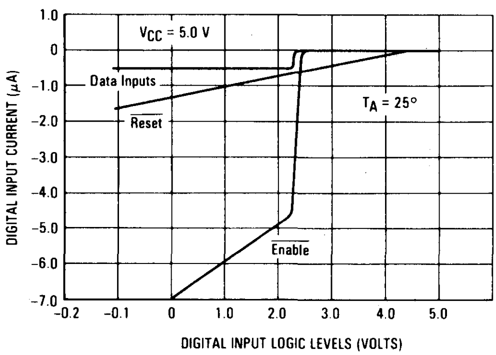
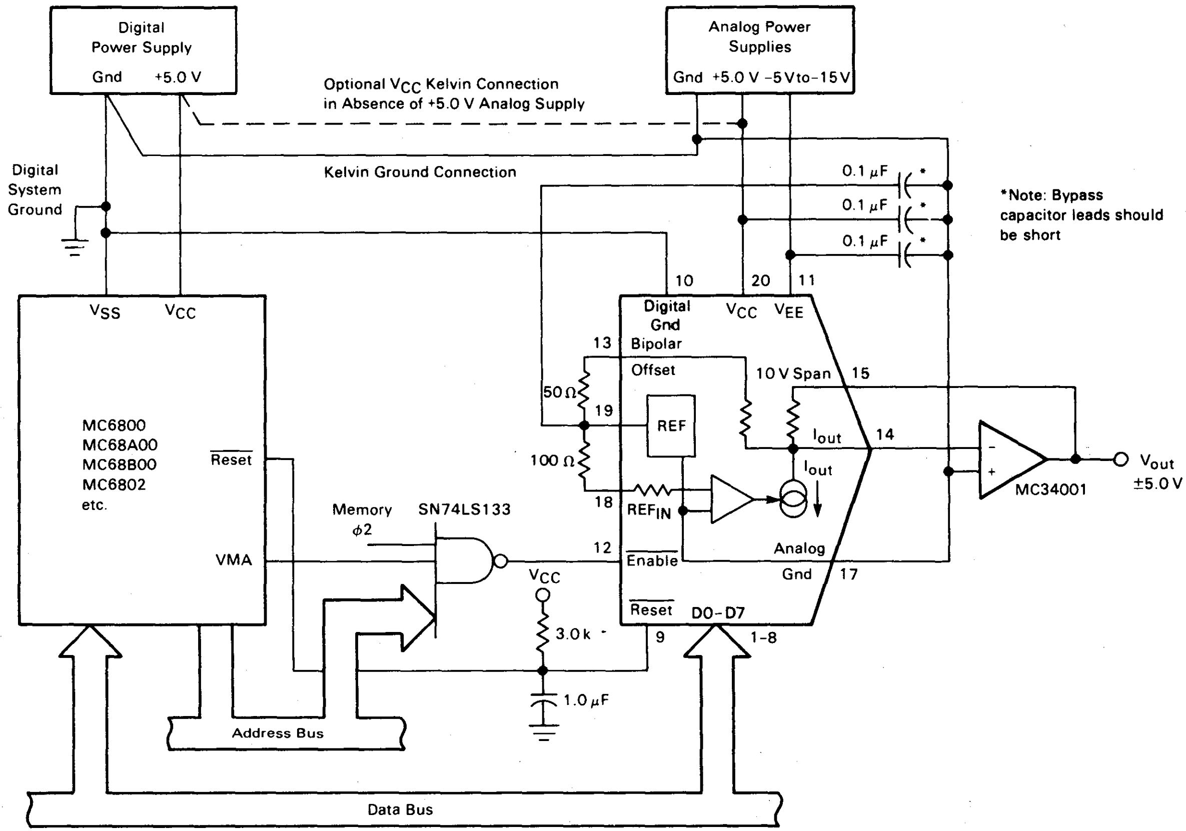


FIGURE 9 — TYPICAL APPLICATION OF THE MC6890 IN A MC6800 SERIES MPU SYSTEM



MOTOROLA

MC68120
MC68121

Advance Information

INTELLIGENT PERIPHERAL CONTROLLER

The MC68120/MC68121 Intelligent Peripheral Controller (IPC) is a general purpose, mask programmable peripheral controller. The IPC provides the interface between an M68000 or M6800 Family microprocessor and the final peripheral devices through a system bus and control lines. System bus data is transferred to and from the IPC via dual-port RAM while the software utilizes the semaphore registers to control RAM tasking or any other shared resource. Multiple operating modes range from a single chip mode with 21 I/O lines and 2 control lines to an expanded mode supporting an address space of 64K bytes. The MC68120 has 2K bytes of on-chip ROM to make full use of all operating modes. The MC68121 utilizes only the expanded address modes, due to the absence of on-chip ROM.

A serial communications interface, 16-bit timer, dual-ported RAM and semaphore registers are available for use by the IPC in all operating modes.

- System Bus Compatible with the Asynchronous M68000 Family
- System Bus Compatible with the MC6809 and Other M6800 Family Processors/Peripherals
- Local Bus Allows Interface with all M6800 Peripherals
- MC6801 Source and Object Code Compatible
- Upward Compatible with MC6800 Source and Object Code
- 2048 Bytes of ROM (MC68120 Only)
- 128 Bytes of Dual-Ported RAM
- Multiple Operation Modes Ranging from Single Chip to Expanded, with 64K Byte Address Space
- Six Shared Semaphore Registers
- 21 Parallel I/O Lines and 2 Handshake Lines (5 I/O Lines on MC68121)
- Serial Communications Interface (SCI)
- 16-Bit Three-Function Timer
- 8-Bit CPU and Internal Bus
- Halt/Bus Available Capability Control
- 8 × 8 Multiply Instruction
- TTL Compatible Inputs and Outputs
- External and Internal Interrupts

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

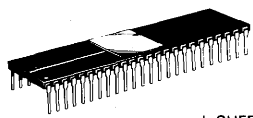
Package Type	Frequency (MHz)	Generic Number
Ceramic	1.0	MC68120L1 (Unicorn ROM)
L Suffix	1.0	MC68121L
	1.25	MC68120L1-1 (Unicorn ROM)
	1.25	MC68121L-1

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

HMOS

(HIGH-DENSITY N-CHANNEL SILICON-GATE)

INTELLIGENT PERIPHERAL CONTROLLER

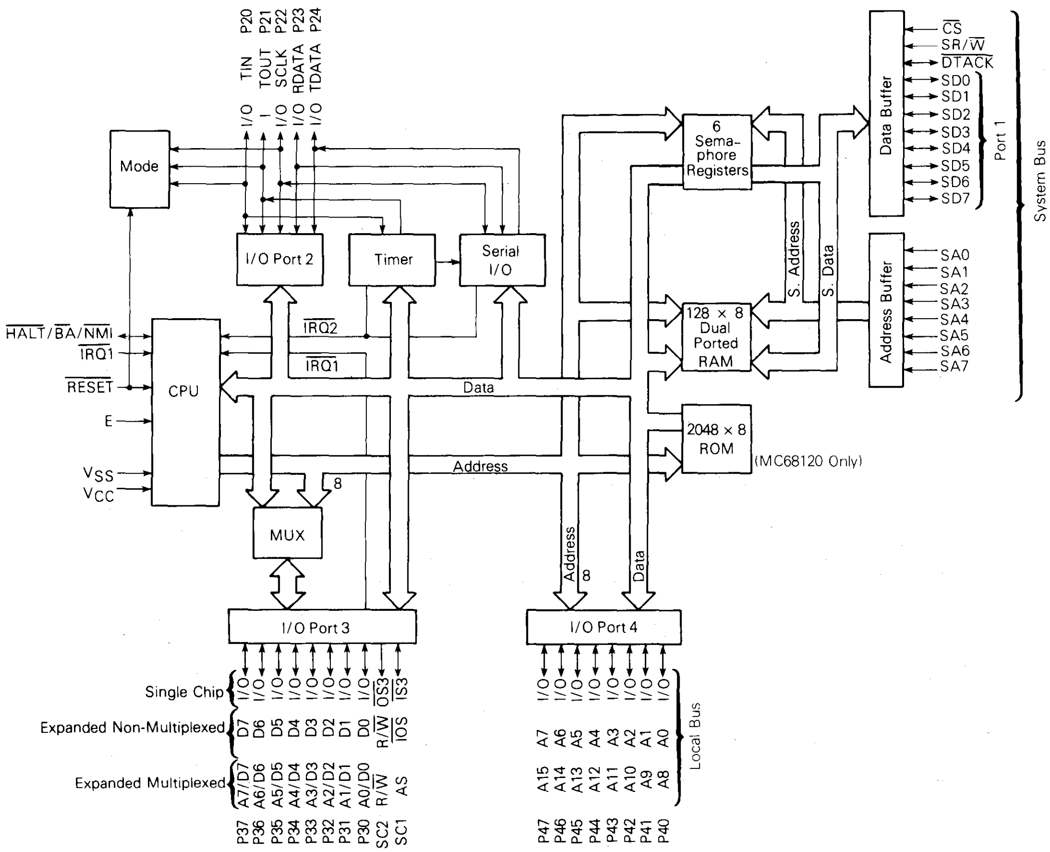


L SUFFIX
CERAMIC PACKAGE
CASE 740

PIN ASSIGNMENT

Signal	Pin	Pin	Signal
V_{SS}	1	48	$\overline{RESET}$
$\overline{IRQ1}$	2	47	P24
$\overline{HALT}$/BA/$\overline{NMI}$	3	46	P23
E	4	45	P22
SR/$\overline{W}$	5	44	P21
$\overline{DTACK}$	6	43	P20
$\overline{CS}$	7	42	SC2
SA7	8	41	SC1
SA6	9	40	P30
SA5	10	39	P31
SA4	11	38	P32
V_{CC}	12	37	P33
SA3	13	36	P34
SA2	14	35	P35
SA1	15	34	P36
SA0	16	33	P37
SD0	17	32	P40
SD1	18	31	P41
SD2	19	30	P42
SD3	20	29	P43
SD4	21	28	P44
SD5	22	27	P45
SD6	23	26	P46
SD7	24	25	P47

MC68120/MC68121 INTELLIGENT PERIPHERAL CONTROLLER — BLOCK DIAGRAM



MAXIMUM RATINGS

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

THERMAL CHARACTERISTICS

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

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

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

POWER CONSIDERATIONS

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

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

Where:

$T_A \equiv$ Ambient Temperature, °C

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

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

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

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

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

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

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

Solving equations 1 and 2 for K gives:

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

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

DC LOCAL BUS ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 0° to 70°C unless otherwise noted)
(Refer to Figures 1 and 2)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	E	V_{EIH}	$V_{CC}-0.75$	–	V_{CC}	V
Input Low Voltage	E	V_{EIL}	$V_{SS}-0.3$	–	$V_{SS}+0.6$	V
Input High Voltage	$\overline{RESET}$ Other Inputs*	V_{IH}	$V_{SS}+4.0$ $V_{SS}+2.0$	– –	V_{CC} V_{CC}	V
Input Low Voltage	All Inputs*	V_{IL}	$V_{SS}-0.3$	–	$V_{SS}+0.8$	V
Input Load Current (V_{in} = 0 to 2.4 V)	Port 4	I_{in}	–	–	0.5	mA
Input Leakage Current (V_{in} = 0 to 5.25 V)	SCI, $\overline{HALT}/\overline{NMI}$, $\overline{IRQ1}$, $\overline{RESET}$	I_{in}	–	1.5	2.5	μA
Three-State (Off State) Input Current (V_{in} = 0.5 to 2.4 V)	SD0-SD7, P20-P24, P30-P37	I_{TSI}	–	2.0	10	μA
Output High Voltage (I_{load} = −65 μA, V_{CC} = min) (I_{load} = −100 μA, V_{CC} = min)	 P40-P47, SC1, SC2 Other Outputs	V_{OH}	 $V_{SS}+2.4$ $V_{SS}+2.4$	 – –	 – –	V
Output Low Voltage (I_{load} = 2.0 mA, V_{CC} = min)	All Outputs	V_{OL}	–	–	$V_{SS}+0.5$	V
Internal Power Dissipation (measured at T_A = 0°C)		P_{INT}	–	–	1200	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f_o = 1.0 MHz)	E P30-P37, P40-P47, SC1 Other Inputs	C_{in}	– – –	– – –	60.0 12.5 10.0	pF

*Except Mode Programming Levels; See Figure 29.

FIGURE 1 — CMOS LOAD

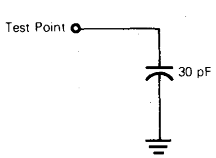
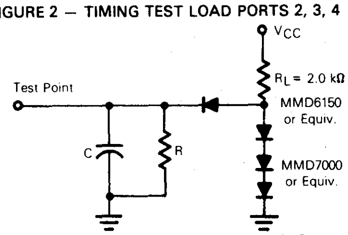


FIGURE 2 — TIMING TEST LOAD PORTS 2, 3, 4



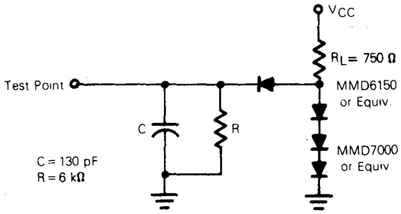
C = 90 pF for P30-P37, P40-P47, $\overline{SC1}$, SC2
= 30 pF for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$
R = 37 kΩ for P40-P47, SC1, SC2
= 24 kΩ for P20-P24, $\overline{HALT}/\overline{BA}/\overline{NMI}$, P30-P37

DC SYSTEM BUS ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0, T_A = 70°C unless otherwise noted) (Refer to Figure 3)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IH}	V_{SS} + 2.0	–	V_{CC}	V
Input Low Voltage	$\overline{CS}$, $\overline{DTACK}$, SA0-SA7, SD0-SD7, SR/$\overline{W}$	V_{IL}	V_{SS} − 0.3	–	V_{SS} + 0.8	V
Output High Voltage (I_{Load} = −400 μA, V_{CC} = min)	$\overline{DTACK}$, SD0-SD7	V_{OH}	V_{SS} + 2.4	–	–	V
Output Low Voltage (I_{Load} = 5.3 mA, V_{CC} = min)	$\overline{DTACK}$, SD0-SD7	V_{OL}	–	–	V_{SS} + 0.5	V

FIGURE 3 — TIMING TEST LOAD SD0-SD7, $\overline{DTACK}$

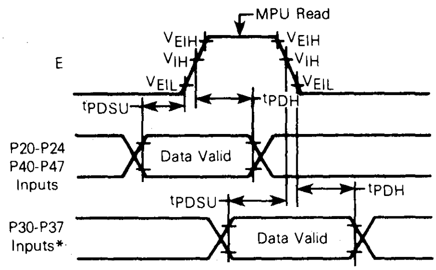


3

PERIPHERAL PORT TIMING (Refer to Figures 4 through 7)

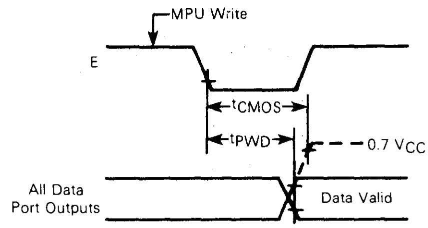
Characteristics	Symbol	Min	Max	Unit
Peripheral Data Setup Time	t_{PDSU}	200	–	ns
Peripheral Data Hold Time	t_{PDH}	200	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition	t_{OSD1}	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition	t_{OSD2}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid (Ports 2, 3, 4)	t_{PWD}	–	350	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	t_{CMOS}	–	2.0	μs
Input Strobe Pulse Width	t_{PWIS}	200	–	ns
Input Data Hold Time	t_{IH}	60	–	ns
Input Data Setup Time	t_{IS}	20	–	ns
Input Capture Pulse Width (Timer Function)	t_{PWIC}	2	–	E_{cyc}

FIGURE 4 — DATA SETUP AND HOLD TIMES (MPU READ LOCAL BUS)



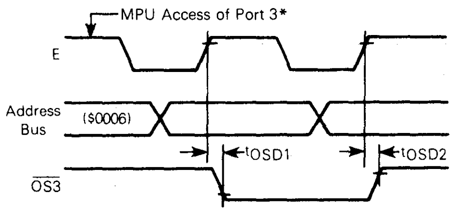
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

FIGURE 5 — DATA SETUP AND HOLD TIMES (MPU WRITE LOCAL BUS)



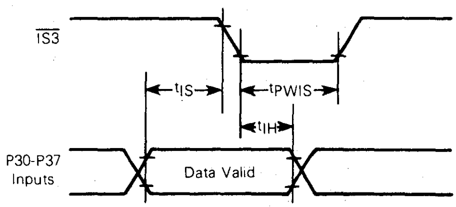
Notes:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

FIGURE 6 — PORT 3 OUTPUT STROBE TIMING (SINGLE CHIP MODE)



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

FIGURE 7 — PORT 3 LATCH TIMING (SINGLE CHIP MODE)

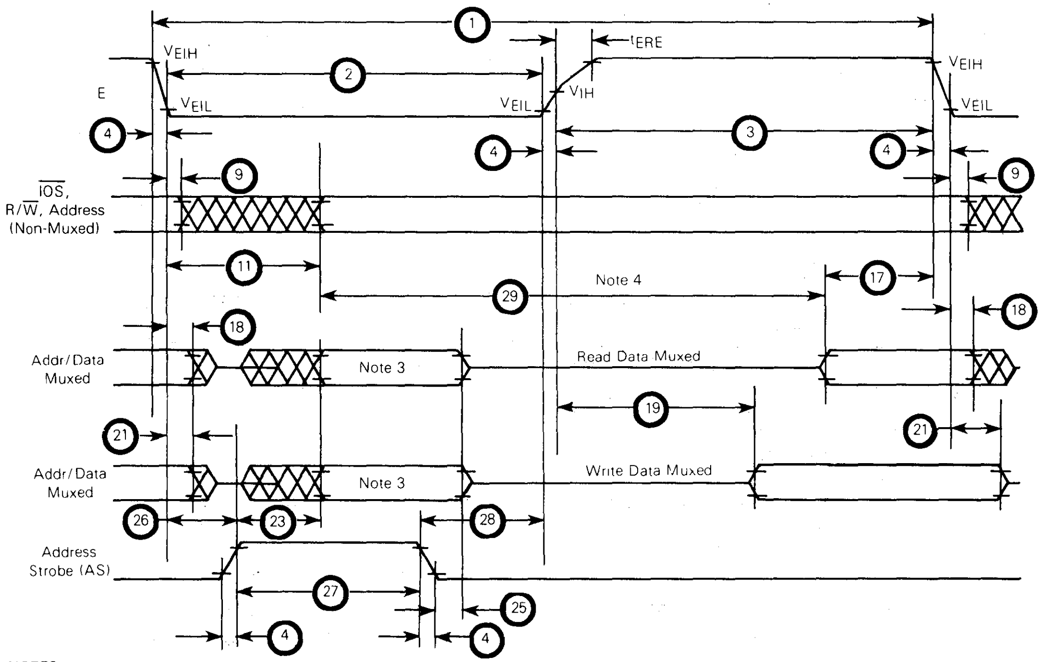


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

LOCAL BUS TIMING (See Notes 1 and 2)

Ident. Number	Characteristics	Symbol	MC68120/ MC68121 Min	MC68120/ MC68121 Max	MC68120-1/ MC68121-1 Min	MC68120-1/ MC68121-1 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	–	80	ns
25	Muxed Address Hold Time	t_{AHL}	20	110	20	110	ns
26	Delay Time E to AS Rise	t_{ASD}	100	–	80	–	ns
27	Pulse Width, AS High	PW_{ASH}	220	–	170	–	ns
28	Delay Time AS to E Rise	t_{ASED}	100	–	80	–	ns
29	Usable Access Time (Note 4)	t_{ACC}	570	–	435	–	ns
	Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns
	Processor Control Setup Time	t_{PCS}	200	–	200	–	ns
	Processor Control Hold Time	t_{PCH}	20	40	20	40	ns

FIGURE 8 — LOCAL BUS TIMING



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

ASYNCHRONOUS SYSTEM BUS TIMING (Refer to Figures 9, 10, 11 and 12)

Characterisic	Symbol	Min	Typ	Max	Unit
Cycle Time	t_{cyc}	0.8	–	2.0	µs
System Address Setup	t_{SAS}	30	–	–	ns
System Address Hold	t_{SAH}	0	–	–	ns
System Data Delay Read Semaphore	t_{SDDR}	0.3	–	0.3+1.5 t_{cyc}*	µs
RAM	t_{SDDR}	–	315	–	ns
System Data Valid	t_{SDV}	0	–	–	ns
System Data Hold Read	t_{SDHR}	0	–	100	ns
System Data Delay Write Semaphore	t_{SDDW}	**	–	**	ns
RAM	t_{SDDW}	–	–	60	ns
System Data Hold Write	t_{SDHW}	0	–	–	ns
Data Acknowledge Semaphore	t_{DAL}	0.5	–	0.5+1.5 t_{cyc}*	µs
RAM	t_{DAL}	–	315	–	ns
Data Acknowledge High	t_{DAH}	–	–	60	ns
Data Acknowledge Three-State	t_{DAT}	–	–	90	ns
Data Acknowledge Low to $\overline{CS}$ High	t_{DCS}	60	–	–	ns

*Actual value dependent upon clock period.
**Data need not be valid on write to Semaphore Registers.

FIGURE 9 — ASYNCHRONOUS READ OF SEMAPHORE REGISTER

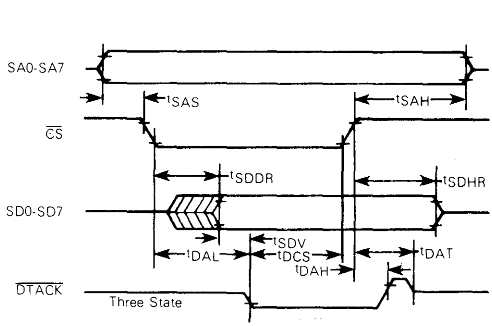


FIGURE 10 — ASYNCHRONOUS WRITE OF SEMAPHORE REGISTER

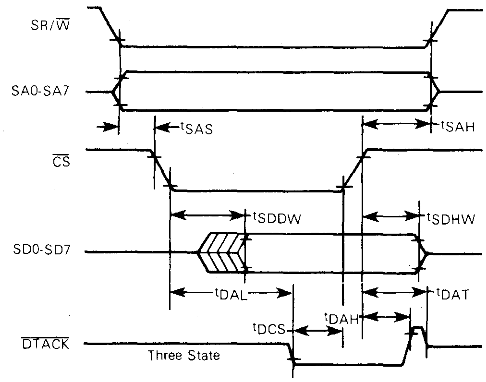


FIGURE 11 — ASYNCHRONOUS READ OF RAM

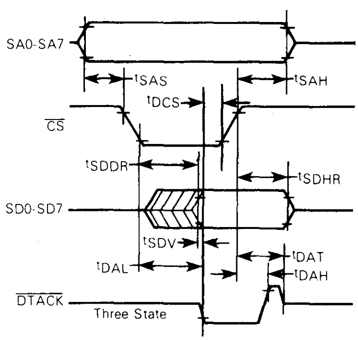
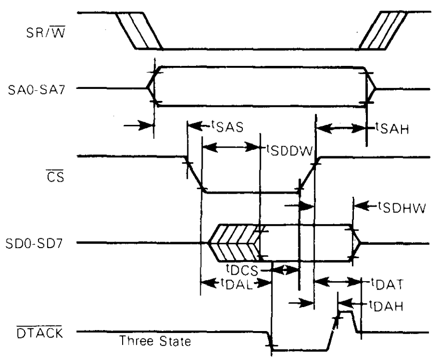


FIGURE 12 — ASYNCHRONOUS WRITE OF RAM

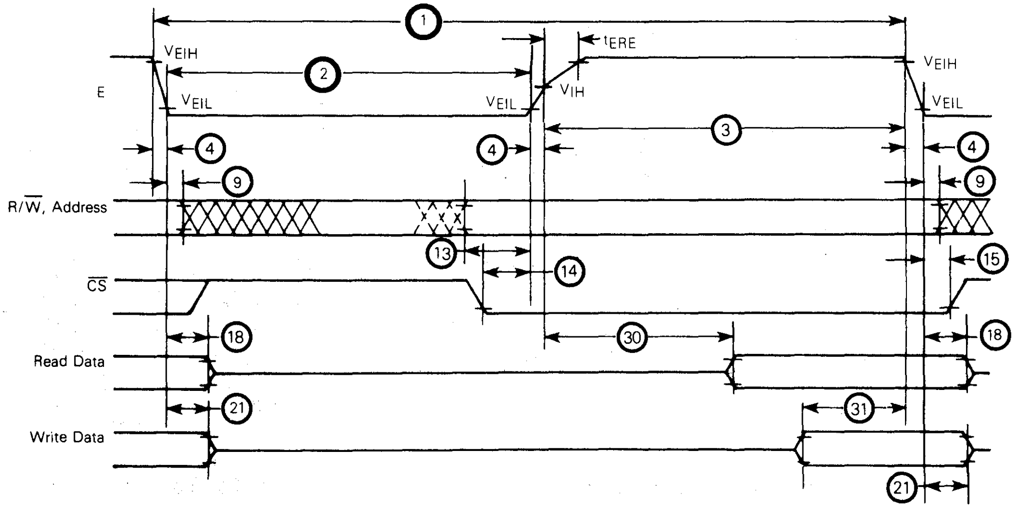


Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

SYNCHRONOUS SYSTEM BUS TIMING (See Notes 1 and 2)

Ident Number	Characteristic	Symbol	MC68120/ MC68121 Min	MC68120/ MC68121 Max	MC68120-1 MC68121-1 Min	MC68120-1 MC68121-1 Max	Unit
1	Cycle Time	t_{cyc}	1.0	10	0.80	10	μs
2	Pulse Width, E Low	PW_{EL}	430	9500	360	9500	ns
3	Pulse Width, E High	PW_{EH}	450	9500	360	9500	ns
4	Clock Rise and Fall Time	t_r, t_f	–	25	–	25	ns
9	Address Hold Time	t_{AH}	10	–	10	–	ns
13	Address Setup Time Before E	t_{AS}	80	–	70	–	ns
14	Chip Select Setup Time Before E	t_{CS}	80	–	70	–	ns
15	Chip Select Hold Time	t_{CH}	10	–	10	–	ns
18	Read Data Hold Time	t_{DHR}	30	100	30	85	ns
21	Write Data Hold Time	t_{DHW}	10	–	10	–	ns
30	Output Data Delay Time	t_{DDR}	–	290	–	250	ns
31	Input Data Setup Time	t_{DSW}	165	–	120	–	ns
	Clock Enable Rise Time Extended	t_{ERE}	–	80	–	80	ns

FIGURE 13 — SYNCHRONOUS SYSTEM BUS TIMING



Notes:
1. Voltage levels shown are $V_L \leq 0.5$ V, $V_H \geq 2.4$ V, unless otherwise specified.
2. Measurement points shown are 0.8 V and 2.0 V, unless otherwise specified.

INTRODUCTION

The MC68120/MC68121 is an 8-bit Intelligent Peripheral Controller (IPC) which can be configured to function in a wide variety of applications. This extraordinary flexibility is provided by its ability to be hardware programmed into eight different operating modes. These operating modes allow the IPC to operate on its local bus and communicate with an external system bus through the internal dual-ported RAM. The operating mode controls the configuration of 18 of the 48 pins on the IPC, the available on-chip resources, the memory map, the location (internal or external) of interrupt vectors, and the type of local bus. The configuration of the remaining 30 pins is not controlled by the operating mode.

The dual-ported RAM provides a vehicle for devices on two separate buses to exchange data without directly affecting the devices on the other bus. The dual-ported RAM is accessible from the MC68120/MC68121 CPU and accessible synchronously or asynchronously to the system bus through Port 1. Semaphore registers are provided as a software tool to arbitrate shared resources such as the dual-ported RAM. The semaphore registers are accessible from both buses in the same way each bus accesses the dual-ported RAM.

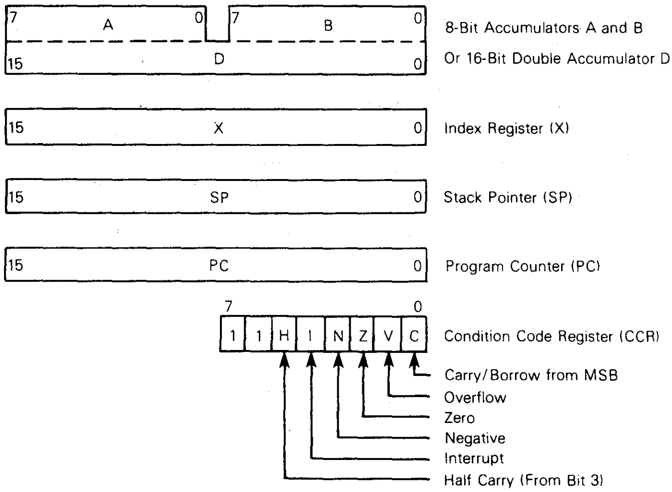
The remaining ports (2, 3, and 4) are I/O ports. Each port is controlled by its Data Direction Register. The CPU has direct access to the port pins of each port through its Data Register. Port pins are labeled as P_{ij} where i identifies one of three ports and j indicates the particular bit. Port 2 is a 5-bit port which may be configured for I/O or for use of the on-chip timer and Serial Communications Interface (SCI). Ports 3 and 4 may be used as 16 bits of I/O or may form a local address and data bus with control lines allowing communications with external memory and peripherals.

The IPC contains an enhanced M6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the MC6800 and directly compatible with the MC6801. The programming model is depicted in Figure 14, where accumulator D is a concatenation of accumulators A and B.

The MC68121 has all of the features of the MC68120 with the exception of on-chip ROM. Thus the MC68121 normally operates in the modes utilizing external ROM (modes 2 and 3). Therefore, modes 0, 1, 4, 5, 6 and 7 should not be used.

3

FIGURE 14 — PROGRAMMING MODEL



DUAL-PORTED RAM AND SEMAPHORE REGISTERS

The dual-ported RAM may be accessed from both the MC68120/MC68121 CPU and the external system bus. The six semaphore registers are tools provided for the programmer's use in arbitrating simultaneous accesses of the same resource.

For the internal CPU, the dual-ported RAM is located from \$0080 through \$00FF in all modes except 3 and 4. In mode 3, the dual-ported RAM has been relocated in high memory from \$C080 through \$C0FF thus allowing use of direct addressing mode on external memory/peripherals. Note that no direct addressing of internal control registers is possible in mode 3. In mode 4, the internal RAM is not fully decoded and appears in locations \$XX80 through \$XXFF. From the external system bus, the dual-ported RAM is found in locations %10000000-11111111, as shown below in Table 1.

TABLE 1 — LOCATION OF SEMAPHORE REGISTERS AND DUAL-PORTED RAM

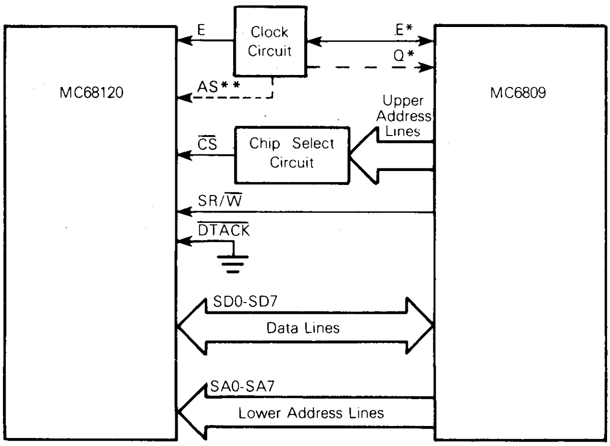
System Bus Address (SA7-SA0)	Feature	IPC Address*
%0000 0000 – 0001 0110	Reserved	------
---- ---- – ---- ----	Internal Registers	\$00-16
0001 0111 – 0001 1100	Semaphore Registers	17-1C
0001 1101 – 0111 1111	Reserved	1D-1F
---- ---- – ---- ----	External Mem./Unusable*	20-7F
1000 0000 – 1111 1111	Dual-Ported RAM	80-FF

% = Binary; \$ = Hexadecimal
* Mode Dependent

The reserved memory areas %0-0001 0110 and %0001 1101-%0111 1111 cannot be written to from the System bus. If read from the System bus these memory locations return a value of \$FF.

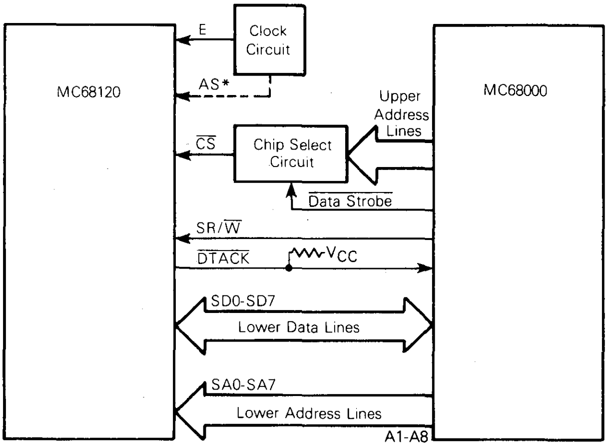
The dual-ported RAM is accessed from the external System bus by way of eight address lines (SA0-SA7) and eight data lines (SD0-SD7). Three control lines provide for synchronous or asynchronous access to the dual-ported RAM through Port 1. Figure 15 shows an example of a synchronous interface (using MC6809) and Figure 16 shows an example of an asynchronous interface (using MC68000). The dual-ported RAM is selected in each case by address lines SA0-SA7 and Chip Select ($\overline{CS}$) from the system bus. The direction of data transfer is selected by the System Read/Write (SR/$\overline{W}$) line. The Data Transfer Acknowledge ($\overline{DTACK}$) signal is the asynchronous handshake required by an MC68000. Refer to $\overline{DTACK}$ under Functional Pin Description for more information. $\overline{DTACK}$ can be used to control a Memory Ready signal on the M6800 Family processor where Memory Ready capability is provided (see Figure 17). The latter would allow the M6800 Family processor to run asynchronously with the MC68120/MC68121. It should be noted that if the Memory Ready signal (on M6800 processors) is to be used with the $\overline{DTACK}$ signal, the system clock must be faster than or equal to the clock driving the IPC. Example clock circuits are shown in Figures 18 and 19.

FIGURE 15 — SYNCHRONOUS SYSTEM BUS ACCESS INTERFACE



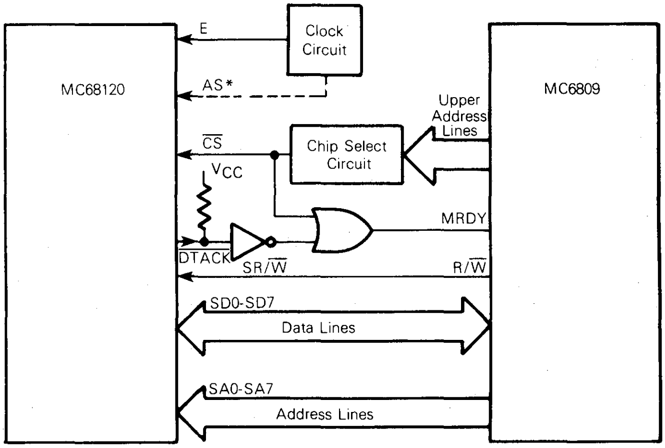
*E and Q are inputs for MC6809E
**Only needed in expanded multiplexed modes.

FIGURE 16 — ASYNCHRONOUS SYSTEM BUS INTERFACE



* Only needed in expanded multiplexed modes.

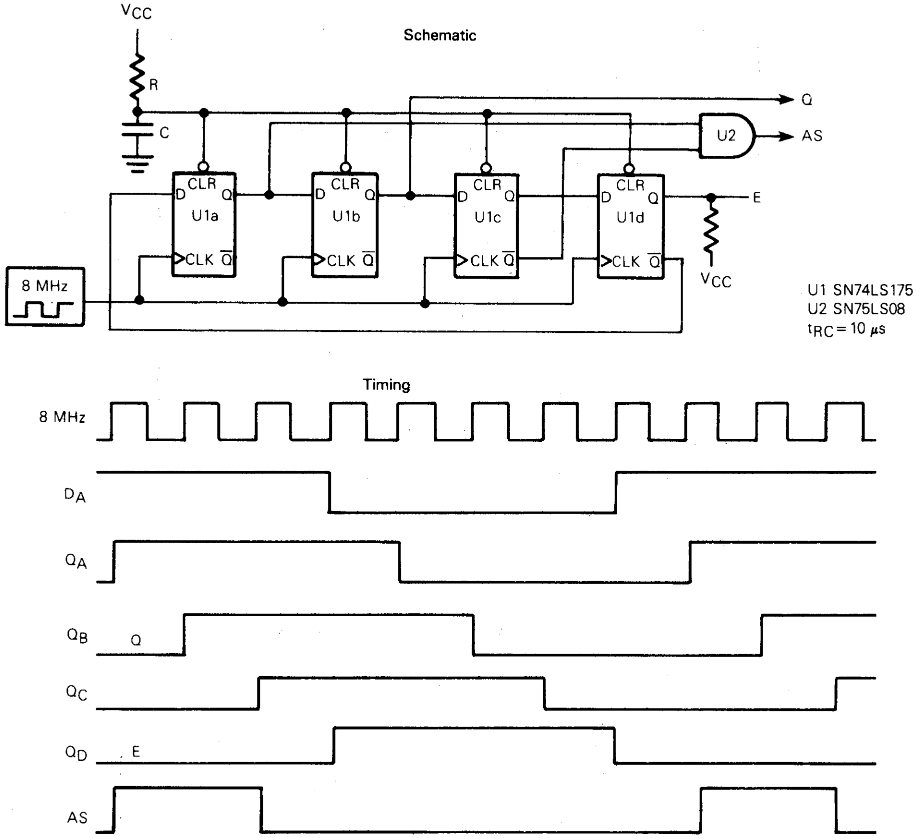
FIGURE 17 — MEMORY READY — DTACK CONFIGURATION



* Only needed in expanded multiplexed modes.

3

FIGURE 18 — CLOCK CIRCUIT EXAMPLE 1 — SCHEMATIC AND TIMING



The semaphore registers allow arbitration between shared resources, which may be part or all of the dual-port RAM, or a peripheral. The semaphore registers may also be used to indicate that non-reentrant code is in use or that a task is in process or is complete. To prevent the writing or reading of erroneous data from the dual-ported RAM, all simultaneous accesses involving a write to the same byte in the dual-ported RAM should be avoided. The responsibility for mutual exclusion resides in software. The semaphore registers are a convenient means for the software to control the simultaneous accesses involving a write to the dual-ported RAM. Each of the six semaphore registers consist of a semaphore bit (SEM, bit 7) and an ownership bit (OWN, bit 6). The remaining six bits (b0-b5) will read all zeros.

SEMAPHORE REGISTER

7	6	5	4	3	2	1	0
SEM	OWN	0	0	0	0	0	0

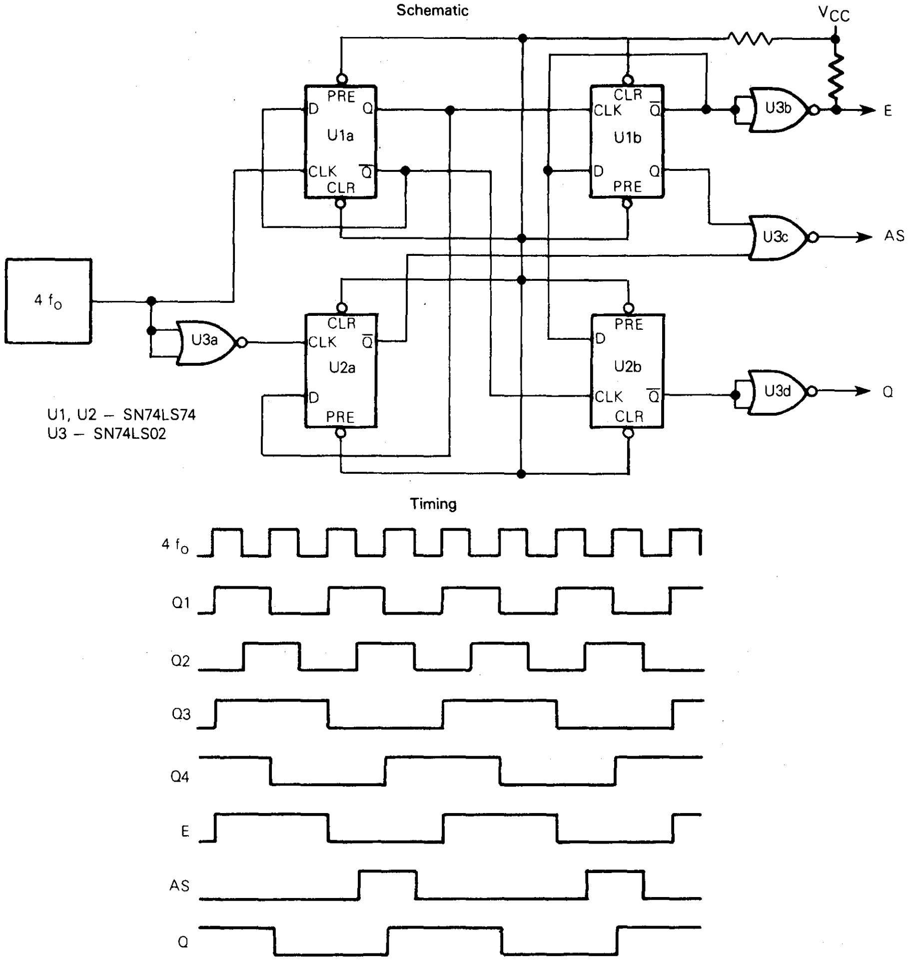
The semaphore bits are test and set bits with hardware arbitration during simultaneous accesses. Basically, the semaphore bit is cleared when written and set when read, during a single processor access. This is shown in Table 2.

TABLE 2 — SINGLE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	$R/\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	W	–	0
1	W	–	0

*0 — Resource Available
1 — Resource Not Available

FIGURE 19 — CLOCK CIRCUIT EXAMPLE 2 — SCHEMATIC AND TIMING



The data written is disregarded and the information obtained from the Read may be interpreted as: 0 — resource available; 1 — resource not available. Thus, any write to a semaphore clears the semaphore bit and makes the associated resource "available."

An access where both the IPC and system processors attempt to read or write the same semaphore register simultaneously is a contested access. During a contested access, the hardware decides which processor reads a clear semaphore bit and which reads a set semaphore bit. Table 3 describes contested operation of a semaphore bit.

The IPC always reads the actual semaphore bit; the system processor reads the semaphore bit in all cases except the simultaneous read of a clear semaphore bit. This arbitration during a simultaneous read ensures that only one processor reads a clear bit and therefore controls the resource; that processor is arbitrarily the IPC.

In Table 3, the first four states are considered proper and they occur in correctly written software. The last four states are improper and only exist in improperly written software.

The ownership bit is a read-only bit that indicates which processor sets the semaphore bit. If the semaphore bit is set, the ownership bit indicates which processor set it. If the semaphore bit is not set, the ownership bit indicates which processor last set the semaphore bit; OWN = 0, the other processor set SEM; OWN = 1, this processor set SEM.

The reset state of the semaphore and ownership bits is defined in Table 4. All of the semaphore bits are set after an MC68120/MC68121 reset. The IPC owns all of them except the second semaphore which is owned by the system processor. This configuration should prevent the system processor from reading a clear semaphore and implying the system processor set it when the IPC $\overline{\text{RESET}}$ is held low.

TABLE 3 — DUAL PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	IPC $R/\overline{W}$	IPC Data Read	System $R/\overline{W}$	System Data Read	Resulting SEM Bit	
0	R	0*	R	1*	1	PROPER
1	R	1*	W	–	0	
1	W	–	R	1*	0	
1	R	1	R	1*	1	
0	W	–	W	–	0	IMPROPER
0	R	0*	W	–	1	
1	W	–	W	–	0	
0	W	–	R	0*	1	

*0 — Resource Available
1 — Resource Not Available

TABLE 4 — RESET STATE OF SEMAPHORE REGISTER

SEM Reg No.	IPC Sem	IPC Own	System Sem	System Own
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

PROGRAM STORAGE MEMORY — ROM

The standard MC68120 comes preprogrammed with a monitor in the ROM. Custom programs are placed in ROM by special order (see Appendix A).

The MC68120 contains 2048 bytes of on-chip, mask programmable read-only memory (ROM) in memory locations \$F800 through \$FFFF. The contents of this ROM allows the IPC to perform a custom function for the user. The interrupt vectors \$FFF0-\$FFFF are decoded to provide vectors at the top of resident ROM. Address \$FFEF is reserved for the checksum value for the ROM. This value is the complement of the "Exclusive OR" of the 2047 bytes of mask programmed ROM. An IPC without ROM is also available as the MC68121. The MC68121 should only be used in modes 2 and 3 to access external ROM after reset.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} AND V_{SS}

V_{CC} and V_{SS} provide power and ground to the IPC. The power supply should provide +5 volts (±5%) to V_{CC} and V_{SS} should be tied to ground. Total power dissipation should not exceed P_D milliwatts.

RESET

The reset function is used for three purposes. The first is to provide the IPC with an orderly and defined start-up procedure from a powerdown condition. The second is to return to start-up conditions without an intervening powerdown condition. The third is to provide a control signal to latch the operating mode.

During reset (low logic level on $\overline{RESET}$ pin), execution of the current instruction is suspended and the CPU enters a "reset state." The register contents are not pushed onto the stack and their contents become undefined during reset. The "reset state" initializes the IPC, as shown in Table 5.

On the positive edge of $\overline{RESET}$, the IPC latches the operating mode from P22, P21 and P20, and then configures Port 3, Port 4, SC1 and SC2. The restart vector is then fetched and transferred to the program counter, then instruction execution begins.

Reset timing is illustrated in Figure 20. The $\overline{RESET}$ line must be held low for a minimum of three E-cycles for the IPC to complete its entire reset sequence. An external RC-network may be used to obtain the required timing.

ENABLE — E

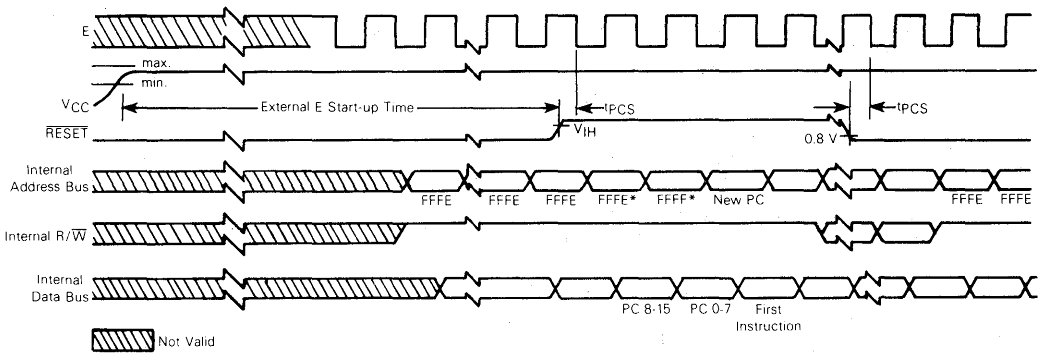
The E clock input is required for timing to synchronize Data Bus transfers. A "CPU E-cycle" (or bus cycle) consists of a negative half-cycle of E followed by a positive half-cycle. For any given bus cycle, the address is valid during the negative half-cycle of E and the selected device must be enabled to the Data Bus during the next positive half-cycle. The data bus is active only while E is high. It should be noted

TABLE 5 — STATE OF IPC DURING RESET

Bits or Registers	Effective State
CPU I-Bit	set (IRQ1 and IRQ2 disabled)
NMI Interrupt Latch	cleared (NMI disabled)
Halt Control Bit	cleared (HALT/BA selected)
All Data Direction Registers	cleared
SCI Rate and Mode Control Register	cleared
Receive Data Register	cleared
Timer Control and Status Register	cleared
Free Running Counter	cleared
Buffer for LSB of Counter	cleared
Port 3 Control and Status Register	cleared
Port 2, 3, 4 Data Registers	undefined after Power-up Reset; and not changed after Reset
SCI Transmit/Receive Control and Status Register	Preset to $20
Output Compare Register	Preset to $FFFF
Semaphore Bits	Preset to 1's
Ownership Bit of Semaphore Register 2	Preset to System Ownership
All other Ownership Bits	Preset to IPC Ownership
All Ports 2 and 3 Lines	High Impedance (inputs)
All Port 4 Lines	High Impedance (inputs) with pullup resistors
SC1*	High Impedance with pullup resistors
SC2	Active High

*If in mode 5, SC1 will go active high; otherwise it will remain in the high impedance state.

FIGURE 20 — RESET TIMING



*Mode 0 – $BFFE, BFFF

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

that this input should have some provision to obtain the specified logical high level which is greater than standard TTL levels.

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

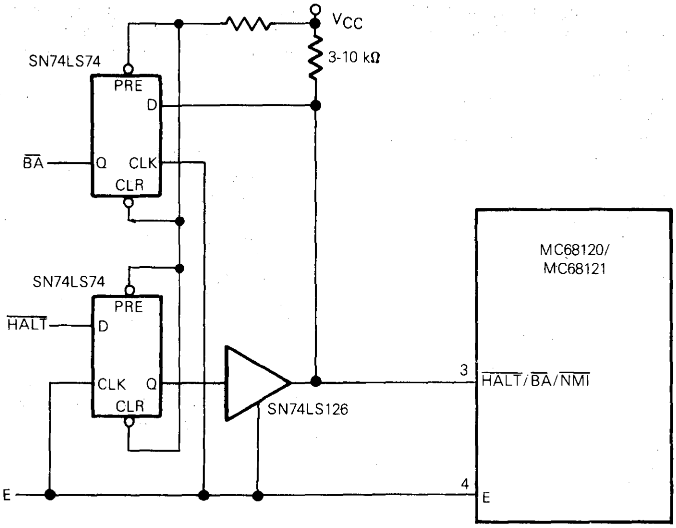
HALT/BUS AVAILABLE/NON-MASKABLE INTERRUPT — HALT/BA/NMI

The HALT/BA/NMI (pin 3) serves one of two functions. These functions are NMI or Halt/BA and the function selected is determined by the Halt Control (HC, bit 2) bit of the Functional Control Register (location $14). If the HC bit is set (to a "1"), then the NMI function is activated. Alternately, if HC is cleared (to a "0" as it is during reset), the Halt/BA function is activated. An external pullup resistor to V_{CC} is required on pin 3 for either function. Typical pullup resistor values range from 3K to 10K depending on the drive capability of the external device.

When the NMI function is implemented, pin 3 is configured as an input. A negative edge on pin 3 then requests an IPC non-maskable interrupt sequence, but the current instruction will be completed before responding to this request. To assure an interrupt under all conditions, NMI must be held low for at least one E-cycle. NMI may be used to cause the IPC to exit the Wait instruction. For interrupt timing specifications, see the interrupt portion of the Operating Mode Section.

When configured to utilize the Halt/BA function of this pin, such as after reset, the circuit of Figure 21 is recommended to detect and supply continuous HALT and BA

FIGURE 21 — $\overline{\text{HALT}}/\overline{\text{BA}}$ DEMULTIPLEXING CIRCUIT

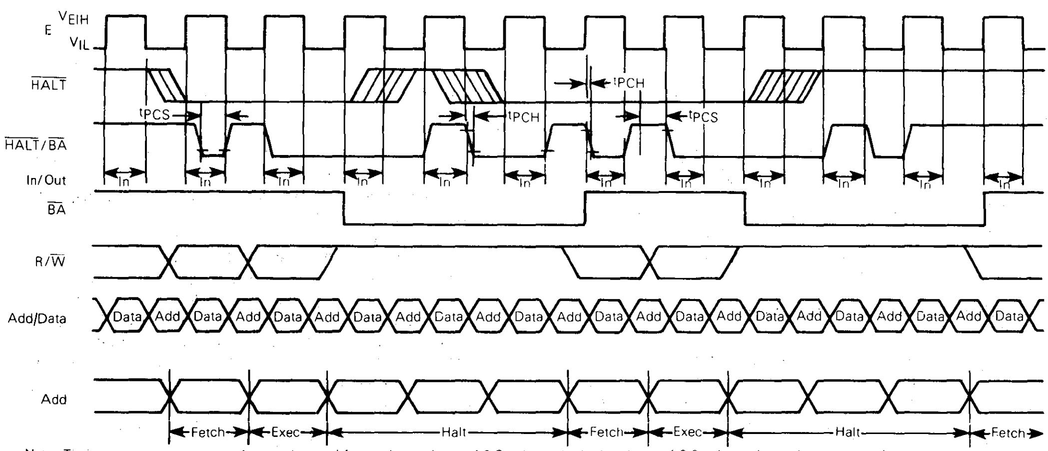


signals. Figure 22 shows the appropriate timing diagram for Halt/BA with the recommended circuit. The pullup resistor shown in the circuit maintains a high logic level when $\overline{\text{HALT}}$ is not active. During a positive half-cycle of E, pin 3 is an input sampled to determine if the Halt State is requested (active low). During the negative half cycle of E, the $\overline{\text{BA}}$ signal is output through pin 3. After the request for Halt State signal is detected and the processor completes its current instruction, the CPU is halted and the active low $\overline{\text{BA}}$ signal is output through pin 3 during the negative half cycle of E. The local bus is then available for other devices to utilize until the Halt State signal has returned to a high level, thus allowing the IPC back on the local bus. During the Halt State, the R/$\overline{\text{W}}$ is high, and the address bus displays the address of the next instruction.

When single instruction operation is desired, in program debug for instance, it is advantageous to single step through instructions. After $\overline{\text{BA}}$ goes low, $\overline{\text{HALT}}$ must be brought high for one E-cycle and returned low again to single step through instructions. Figure 22 illustrates the timing involved while single stepping through a single byte, two bus cycle instruction, such as CLRA.

$\overline{\text{BA}}$ is not output in response to the Wait instruction. If interrupts are to be utilized in removing the processor from a

FIGURE 22 — $\overline{\text{HALT}}/\overline{\text{BA}}$ TIMING DIAGRAM



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

Wait State while in the Halt/BA mode then, $\overline{IRQ1}$ and $\overline{IRQ2}$ are the only interrupts which may do so; therefore, their masks must be cleared before entering the Wait State.

MASKABLE INTERRUPT REQUEST 1 — $\overline{IRQ1}$

This level-sensitive input can be used to request an interrupt sequence. The IPC will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the IPC will begin an interrupt sequence: a vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is continued at the new location. This is explained in greater detail in the Interrupt Section.

$\overline{IRQ1}$ typically requires an external resistor (3K to 10K depending on external devices drive capability) to V_{CC} for wire-OR applications. $\overline{IRQ1}$ has no internal pullup resistor.

STROBE CONTROL 1 AND 2 — SC1 and SC2

The functions of SC1 and SC2 depend on the operating mode. SC1 is configured as an input in all modes except the Expanded Non-Multiplexed Mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90 pF.

Single Chip Modes — In these modes, SC1 and SC2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC1 functions as an input strobe (IS3) and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{IS3}$ are controlled by the Control and Status Register for Port 3 and are discussed in the Port 3 description.

SC2 is configured as an output strobe ($\overline{OS3}$) and can be used to strobe output data or acknowledge input data for Port 3. It is controlled by Output Strobe Select (OSS) in the Port 3 Control and Status Register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the Port 3 Data Register. $\overline{OS3}$ timing is shown in Figure 6.

Expanded Non-Multiplexed Mode — In this mode, both SC1 and SC2 are configured as outputs. SC1 functions as Input/Output Select (IOS) and is asserted (active-low) only when addresses $0100 through $01FF are accessed. SC2 is configured as R/$\overline{W}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{W}$ and E are high.

Expanded Multiplexed Modes — In these modes, SC1 is configured as an input and SC2 is configured as an output. In the expanded multiplexed modes, the IPC has the ability to access a 64K byte address space. SC1 functions as an input, Address Strobe, which controls demultiplexing and enabling of the eight least significant addresses and the data buses.

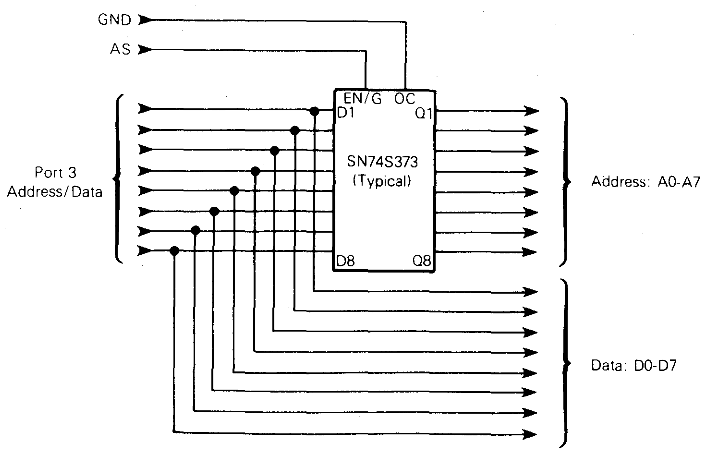
By using a transparent latch such as an SN74LS373 or MC6882, Address Strobe (AS) can also be used to demultiplex the two buses external to the IPC. (See Figure 23.) SC2 provides the local Data Bus control signal called Read/Write (R/$\overline{W}$). SC2 is configured as R/$\overline{W}$ and is used to control the direction of local data bus transfers. An MPU read is enabled when R/$\overline{W}$ and E are high.

SYSTEM BUS INTERFACE

Port 1 is a mode-independent 8-bit data port which permits the external system bus to access the dual-ported RAM and semaphore registers either asynchronously or synchronously with respect to the E clock. In addition to the eight data lines (SD0-SD7), eight address (SA0-SA7) and three control lines (SR/$\overline{W}$, $\overline{CS}$, $\overline{DTACK}$) are used to access the dual-ported RAM and semaphore registers.

Port 1 Data Lines (SD0-SD7) — These data lines are bidirectional data lines which allow data transfer between the dual-ported RAM or the semaphore registers, and the system bus. The data bus output drivers are three-state devices which remain in the high-impedance state except

FIGURE 23 — TYPICAL LATCH ARRANGEMENT



during a read of the IPC dual-ported RAM or semaphore registers by the system processor.

System Address Lines (SA0-SA7) — The address lines together with the Chip Select signal allow any of the 128 bytes of RAM or six semaphore registers to be uniquely selected from the system bus. The address lines must be valid before the $\overline{\text{CS}}$ signal goes low for the asynchronous interface and valid before the E signal goes high for the synchronous interface. The system interface must be deselected between reads or between writes for the asynchronous operation.

System Read/Write (SR/$\overline{\text{W}}$) — This signal is generated by the system bus to control the direction of data transfer on the data bus. With the IPC selected, a low on the SR/$\overline{\text{W}}$ line enables the input buffers, and data is transferred from the system processor to the IPC. When SR/$\overline{\text{W}}$ is high and the chip is selected, the data output buffers are turned on and data is transferred from the IPC to the system bus.

Chip Select ($\overline{\text{CS}}$) — This signal is a TTL compatible input signal, used to activate the system bus interface and allows transfer of data between the IPC and the system processor during synchronous or asynchronous accesses. $\overline{\text{CS}}$ provides the synchronizing signal for the Semaphore registers during access by the system bus.

Data Transfer Acknowledge ($\overline{\text{DTACK}}$) — This bidirectional control line is used to determine synchronous or asynchronous system bus accesses and to provide the data acknowledge signal for asynchronous data transfers.

As an input, it is sampled on the falling edge of $\overline{\text{CS}}$ by the IPC to determine if the system bus is being accessed synchronously or asynchronously with respect to the E clock.

If $\overline{\text{DTACK}}$ is low when sampled, the system bus is synchronous and data will be transferred during E high as shown in Figure 13.

If $\overline{\text{DTACK}}$ is high when sampled, the system bus is asynchronous. In this mode $\overline{\text{DTACK}}$ becomes an output that is asserted low when data is on the bus during a system read or when a data transfer is completed during a system write. Refer to Figures 9 through 12.

$\overline{\text{DTACK}}$ requires an external pullup resistor when the system bus is run asynchronously since it is then a bidirectional handshake line for information transfer on the system data bus.

PORT 2 — P20-P24

Port 2 is a mode independent 5-bit I/O port where each line is configured by its Data Direction Register. During reset, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, or CMOS devices using external pullup resistors. P20, P21 and P22 must always be connected to provide the operating mode.

PORT 2 DATA REGISTER

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

Inputs on P20, P21 and P22 determine the operating mode which is latched into the Program Control Register on the positive edge of $\overline{\text{RESET}}$. The mode may be read from the Port 2 Data Register (PC2 is latched from pin 45).

Port 2 also provides an interface for the Serial Communications Interface and Timer. Bit 1, if configured as an output, is dedicated to the Timer Output Compare function and cannot be used to provide output from the Port 2 Data Register.

PORT 3 — P30-P37

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

Single Chip Modes — In these modes, Port 3 is an 8-bit I/O port where each line is configured by the Port 3 Data Direction Register. Associated with Port 3 are two lines, $\overline{\text{IS3}}$ and $\overline{\text{OS3}}$, which can be used to control Port 3 data transfers.

Three Port 3 options, controlled by the Port 3 Control and Status Register and available only in the Single Chip Modes are: 1) Port 3 input data can be latched using $\overline{\text{IS3}}$ as a control signal, 2) $\overline{\text{OS3}}$ can be generated by either an IPC 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 7.

PORT 3 CONTROL AND STATUS REGISTER

7	6	5	4	3	2	1	0	
IS3 FLAG	IS3 IRQ1 ENABLE	X	OSS	LATCH ENABLE	X	X	X	\$0F

Bits 0-2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{\text{IS3}}$ negative edge. The latch is transparent after a read of the Port 3 Data Register. LATCH ENABLE is cleared by Reset.

Bit 4 OSS (Output Strobe Select). This bit determines whether $\overline{\text{OS3}}$ 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 by Reset.

Bit 5 Not used.

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

Bit 7 $\overline{\text{IS3}}$ 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 $\overline{\text{IS3}}$ FLAG set) followed by a read or write to the Port 3 Data Register or by Reset.

Expanded Non-Multiplexed Mode — In this mode, Port 3 is configured as a bi-directional data bus (D0-D7). The direction of data transfers is controlled by R/$\overline{\text{W}}$ (SC2). Data transfers are clocked by E (Enable).

Expanded Multiplexed Modes — In these modes, Port 3 is configured as a time-multiplexed address (A0-A7) and data bus (D0-D7). Address Strobe (AS) must be input on SC1, and can be used externally to de-multiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent potential bus conflicts.

PORT 4 — P40-P47

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

Single Chip Modes — In these modes, Port 4 functions as an 8-bit I/O port where each line is configured by the Port 4 Data Direction Register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Expanded Non-Multiplexed Mode — In this mode, Port 4 is configured from reset as an 8-bit input port, where the Data Direction Register can be written, to provide any or all of address lines A0-A7. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured.

Expanded Multiplexed Mode — In all these modes except Mode 6, Port 4 functions as half of the address bus and provides A8 to A15. In Mode 6, the port is configured from reset as an 8-bit parallel input port; the Port 4 Data-Direction Register must be written to provide any or all of address lines, A8 to A15. Internal pullup resistors are intended to pull the lines high until the Data Direction Register is configured (bit 0 controls A8, etc.).

OPERATING MODES

The IPC provides eight different operating modes which are selectable by hardware programming and referred to as Modes 0 through 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC1 and SC2 and the address location of the interrupt vectors.

FUNDAMENTAL MODES

The eight modes of the IPC can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single Chip includes Modes 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. A system utilizing three MC68120's, one in each of the fundamental operating modes, is shown in Figure 24. Table 6 summarizes the characteristics of the operating modes.

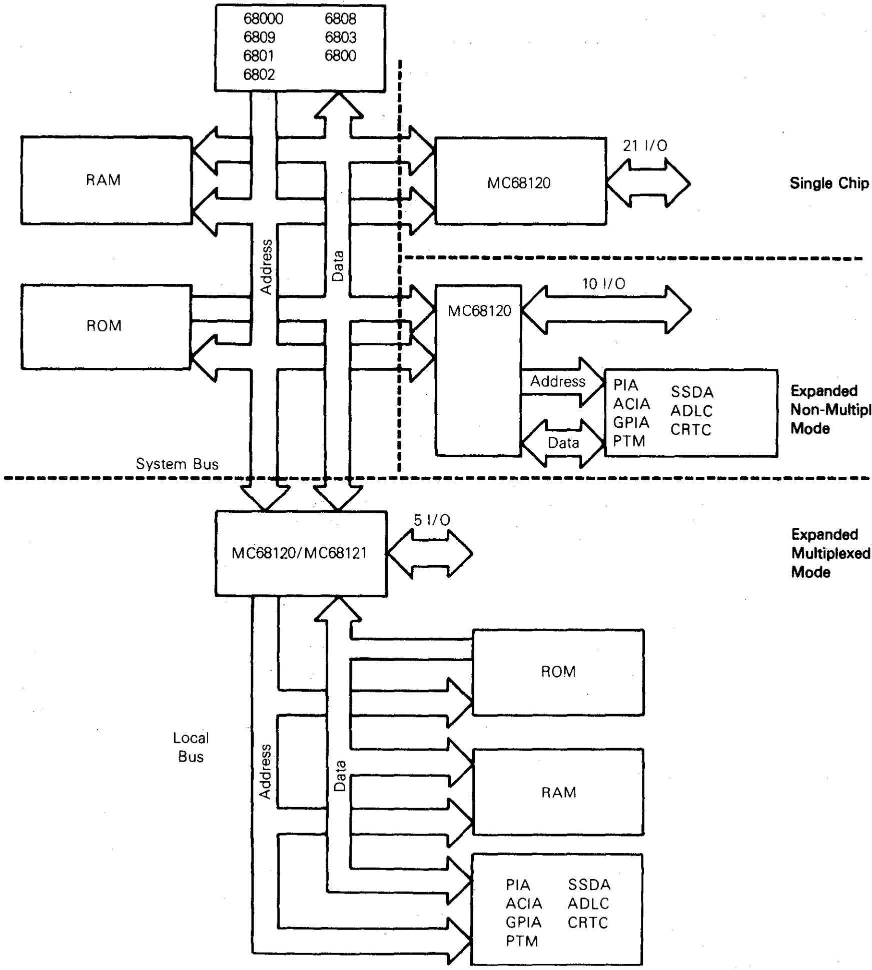
Single Chip Modes (4, 7) — In Single Chip Mode, three of the four IPC ports are configured as parallel input/output data ports, as shown in Figure 25. The IPC functions as a complete microcomputer in these two modes without external address or data buses. A maximum of 21 I/O lines and two Port 3 control lines are provided.

In Single Chip Test Mode (4), the RAM responds to addresses $XX80 (X = don't care) through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using Modes 0, 1, 2, or 6. If the IPC is reset and then programmed into Mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from Mode 4 without going through reset by setting bit 5 of the Port 2 Data Register. This mode is used primarily to test Port 3 and 4 in the Single Chip and Non-Multiplexed Modes.

TABLE 6 — SUMMARY OF IPC OPERATING MODES

Common to all Modes: System Bus Interface Reserved Register Area 6 Semaphore Registers I/O Port 2 Programmable Timer Serial Communications Interface 128 bytes of Dual Ported RAM	Expanded Multiplexed Modes Four Memory Space Options (64K Address Space): (1) MDOS Compatible (2) No ROM (3) External Vector Space (4) ROM with Partial Address Bus* External Memory Space Accessed Through: Port 3 as a Multiplexed Address/Data Bus Port 4 as an Address Bus (High) SC1 is Address Strobe Bus (AS) Input SC2 is Read/Write (R/$\overline{W}$)
Single Chip Mode* 2048 Bytes of ROM (Internal) Port 3 is a Parallel I/O Port with Two Control Lines Port 4 is a Parallel I/O Port SC1 is Input Strobe 3 ($\overline{IS3}$) SC2 is Output Strobe 3 ($\overline{OS3}$)	Test Modes Expanded Multiplexed Test Mode May be Used to Test RAM and ROM* Single Chip and Non-Multiplexed Test Mode* May be Used to Test Ports 3 and 4 as I/O Ports
Expanded Non-Multiplexed Mode* 2048 Bytes of ROM (Internal) 256 Bytes of External Memory Space Port 3 is an 8-Bit Data Bus Port 4 is an Address Bus SC1 is Input/Output Select ($\overline{IOS}$) SC2 is Read/Write (R/$\overline{W}$)	*MC68120 only

FIGURE 24 — IPC FUNDAMENTAL OPERATING MODES



3

Expanded Non-Multiplexed Mode (5) — A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bi-directional data bus and Port 4 is configured as an input data port. Any combination of A0 to A7 may be provided while retaining the remainder as input data lines. Any combination of the eight least-significant address lines may be obtained by writing to the Port 4 Data Direction Register. Internal pullup resistors are provided to pull Port 4 lines high until it is configured.

Figure 26 illustrates the external resources available in the Expanded Non-Multiplexed Mode. The IPC interfaces directly with M6800 Family parts and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and may be used as an address or chip select line.

Expanded-Multiplexed Modes (0, 1, 2, 3, 6) — In the Expanded Multiplexed Modes, the IPC has the ability to access a 64K-byte memory space. Port 3 functions as a time-multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS) and the data bus valid while E is high. In Modes 0 to 3, Port 4 provides address lines A8-A15. However, in Mode 6, Port 4 can provide any subset of A8 to A15 while retaining the remainder as input lines. Writing 1's to the desired bits in the Data Direction Register (DDR) will output the corresponding address lines while the remaining bits will remain inputs (as configured from reset or from 0's written to the DDR). Internal pullup resistors are provided to pull Port 4 lines high until software configures the port. Initialization of Port 4 in Mode six must be done to obtain any upper address lines externally.

FIGURE 25 — SINGLE CHIP MODE

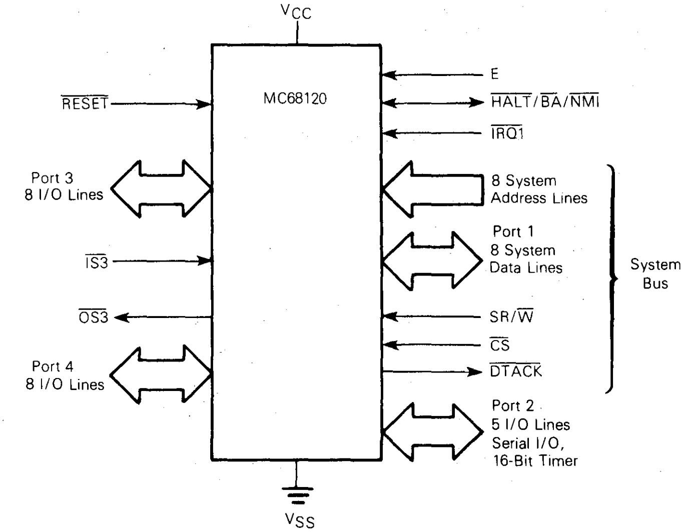


FIGURE 26 — EXPANDED NON-MULTIPLEXED MODE

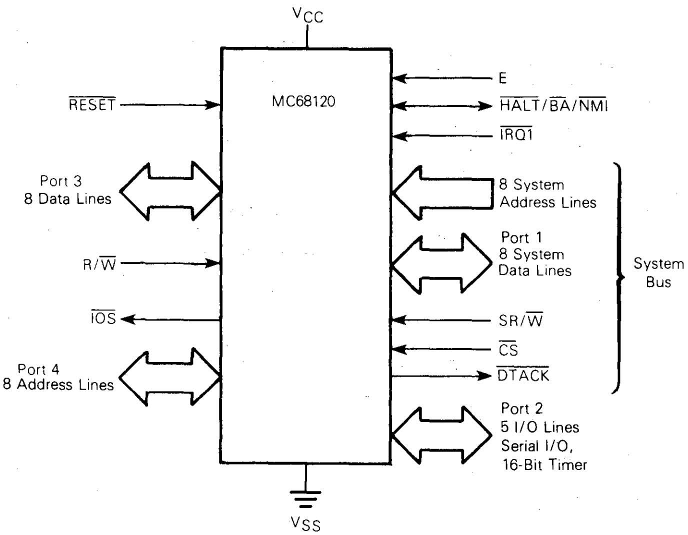
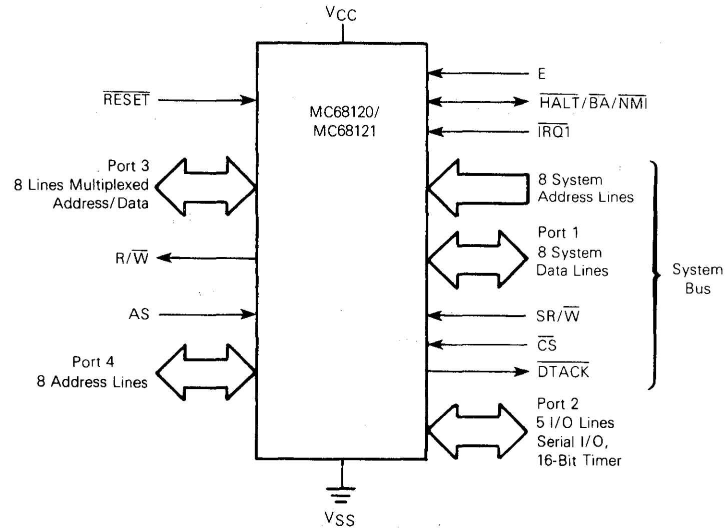


Figure 27 depicts the external resources available in the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in Figure 23. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the reset vector is external at $BFFE and $BFFF after the positive edge of RESET. In addition, the internal and external data buses are connected together so there must be no memory map overlap (to avoid potential bus conflicts). Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with automated test equipment.

FIGURE 27 — EXPANDED MULTIPLEXED MODE



MODE PROGRAMMING

The operating mode is programmed by the levels asserted on P22, P21, and P20 during the positive edge of RESET. These are latched into PC2, PC1, and PC0 of the program control register. The operating mode may be read from the Port 2 Data Register and programming levels and timing must be met as shown in Figure 28 and Table 7. Any mode may be entered from either Mode 0 or Mode 4 without going through reset by writing the appropriate bits to the port 2 data register. A brief outline of the operating modes is shown in Table 8.

Circuitry to provide the programming levels is primarily dependent on the normal system use of the three pins. If configured as outputs, the circuit shown in Figure 29 may be used; otherwise, the three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The IPC provides up to 64K bytes of address space depending upon the operating mode. A memory map for each operating mode is shown in Figure 30. In Modes 1R and 6R, the "R" means the ROM has been relocated by a mask option. The first 32 locations of each map are reserved for the IPC internal register area, as shown in Table 9, with exceptions as indicated.

FIGURE 28 — MODE PROGRAMMING TIMING

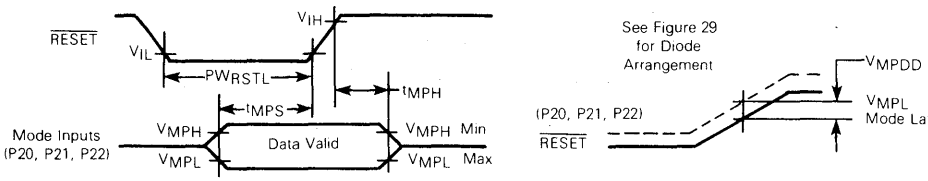


TABLE 7 — MODE PROGRAMMING SPECIFICATIONS (See Figure 30)

Characteristic	Symbol	Min	Typ	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	–	V
Mode Programming Diode Differential (if Diodes are Used)	V_{MPDD}	0.6	–	–	V
$\overline{\text{RESET}}$ 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

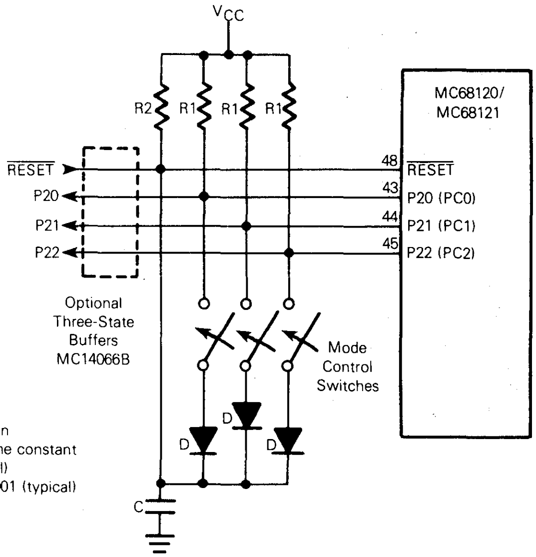
TABLE 8 — MODE SELECTION SUMMARY

Mode	Pin 45 P22 PC2	Pin 44 P21 PC1	Pin 43 P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode(5)
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode(5)
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	I(7)	E	MUX(4)	Multiplexed/RAM(4)
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM(4)
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and ROM(4)
0	L	L	L	I	I	E(3)	MUX(4)	Multiplexed Test(4)

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

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) Interrupt vectors externally located at $BFF0-$BFFF
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register
(7) Internal RAM and registers located at $C0XX (for use with MDOS)

FIGURE 29 — TYPICAL MODE PROGRAMMING CIRCUIT

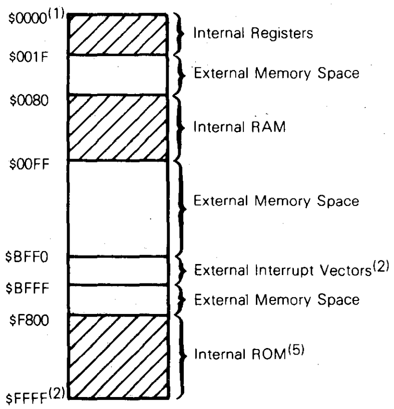


Notes:
1. Mode 7 as shown
2. R2•C = Reset time constant
3. R1 = 10 k (typical)
4. D = 1N914, 1N4001 (typical)

FIGURE 30 — IPC MEMORY MAPS

MC68120 Mode **0**

Multiplexed Test Mode

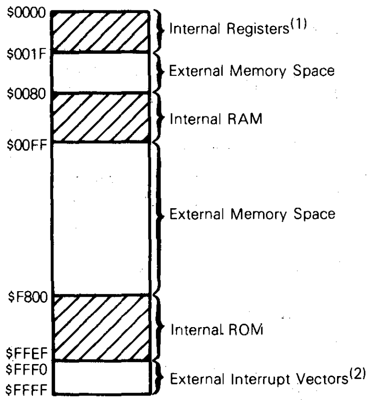


Notes:

1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) The interrupt vectors are externally located at $BFF0-$BFFF.
3) There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{RESET}$ vector.
5) MC68120 only.

MC68120 Mode **1**

Multiplexed/RAM and ROM



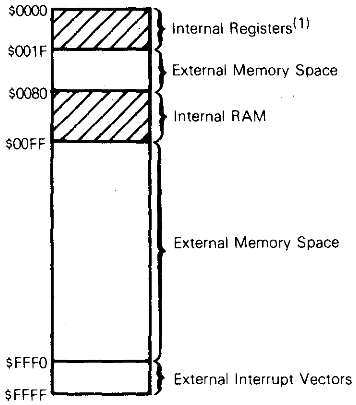
Notes:

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

FIGURE 30 — IPC MEMORY MAPS (CONTINUED)

MC68120/
MC68121
Mode **2**

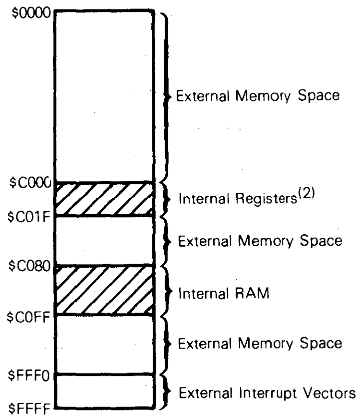
Multiplexed/RAM



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

MC68120/
MC68121
Mode **3**

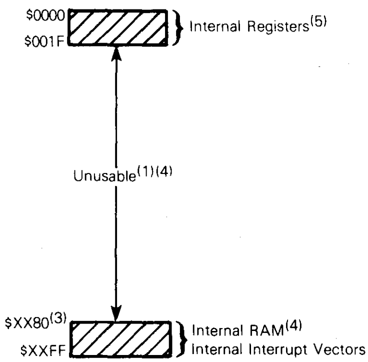
Multiplexed/RAM, MDOS Compatible (1)



Notes:
1) Relocating the internal registers and the internal RAM to high memory allows processor to run MDOS.
2) Excludes the following addresses which may be used externally: $C004, $C005, $C006, $C007, and $C00F.

MC68120
Mode **4**

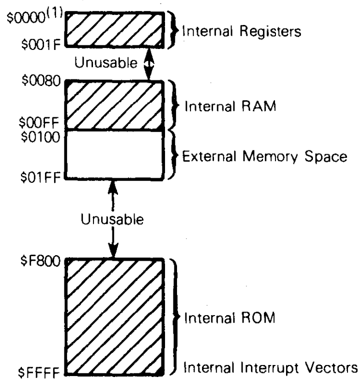
Single Chip Test(2)



Notes:
1) The internal ROM is disabled.
2) Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a "1" into bit 5 (PC0) of Port 2 Data Register.
3) Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4) Internal RAM will appear at $XX80 to $XXFF.
5) MPU Read of Port 3 Data Direction Register will access Port 3 Data Register instead.

MC68120
Mode **5**

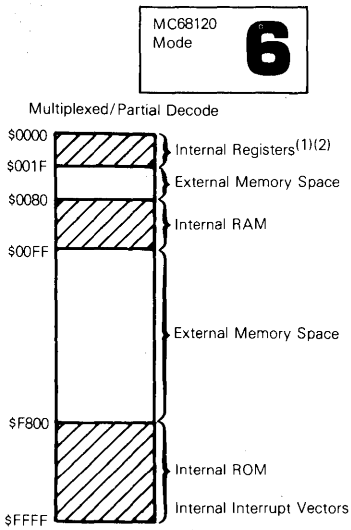
Non-Multiplexed/Partial Decode (2) (3)



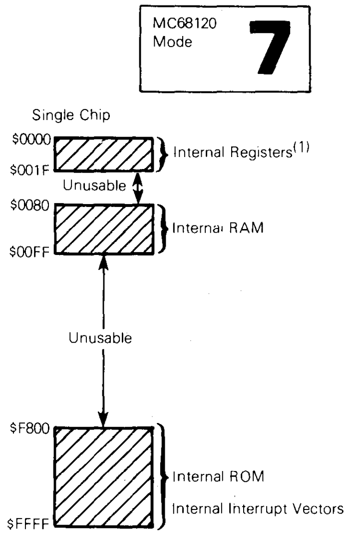
Notes:
1) Excludes the following addresses which may not be used externally: $04, $06, and $0F (no $\overline{\text{IOS}}$).
2) This mode may be entered without going through Reset by using Mode 4 and subsequently writing a "1" into bit 5 (PC0) of Port 2 Data Register.
3) Address lines A0 to A7 will 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.

3

FIGURE 30 — IPC MEMORY MAPS (CONCLUDED)



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



Notes:
1) MPU reads of Port 3's Data Direction Register will access Port 3's Data Register instead.

TABLE 9 — INTERNAL REGISTER AREA

Register	Address**** (Hexadecimal)	Register	Address**** (Hexadecimal)
Reserved	00	SCI Rate and Mode Control Register	10
Port 2 Data Direction Register***	01	Transmit/Receive Control and Status Register	11
Reserved	02	SCI Receive Data Register	12
Port 2 Data Register	03	SCI Transmit Data Register	13
Port 3 Data Direction Register***	04*		
Port 4 Data Direction Register***	05**	Function Control Register	14
Port 3 Data Register	06*	Counter Alternate Address (High Byte)	15
Port 4 Data Register	07**	Counter Alternate Address (Low Byte)	16
Timer Control and Status Register	08	Semaphore 1	17
Counter (High Byte)	09	Semaphore 2	18
Counter (Low Byte)	0A	Semaphore 3	19
Output Compare Register (High Byte)	0B	Semaphore 4	1A
Output Compare Register (Low Byte)	0C	Semaphore 5	1B
Input Capture Register (High Byte)	0D	Semaphore 6	1C
Input Capture Register (Low Byte)	0E	Reserved	1D-1F
Port 3 Control and Status Register	0F*		

*These external addresses in Modes 0, 1, 2, 3, 5, 6 cannot be accessed in Mode 5 (no IOS).
**These are external addresses in Modes 0, 1, 2, 3.
***1 = Output, 0 = Input
****These addresses relocated at $C000-$C01F in Mode 3.

3

INTERRUPTS

The IPC 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 of the IPC. External devices (and $\overline{\text{IS3}}$) use $\overline{\text{IRQ1}}$. An $\overline{\text{IRQ1}}$ interrupt is serviced before an $\overline{\text{IRQ2}}$ interrupt 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 where each is vectored to a separate location. All IPC vector locations are shown in Table 10, from highest (top) to lowest (bottom) priority.

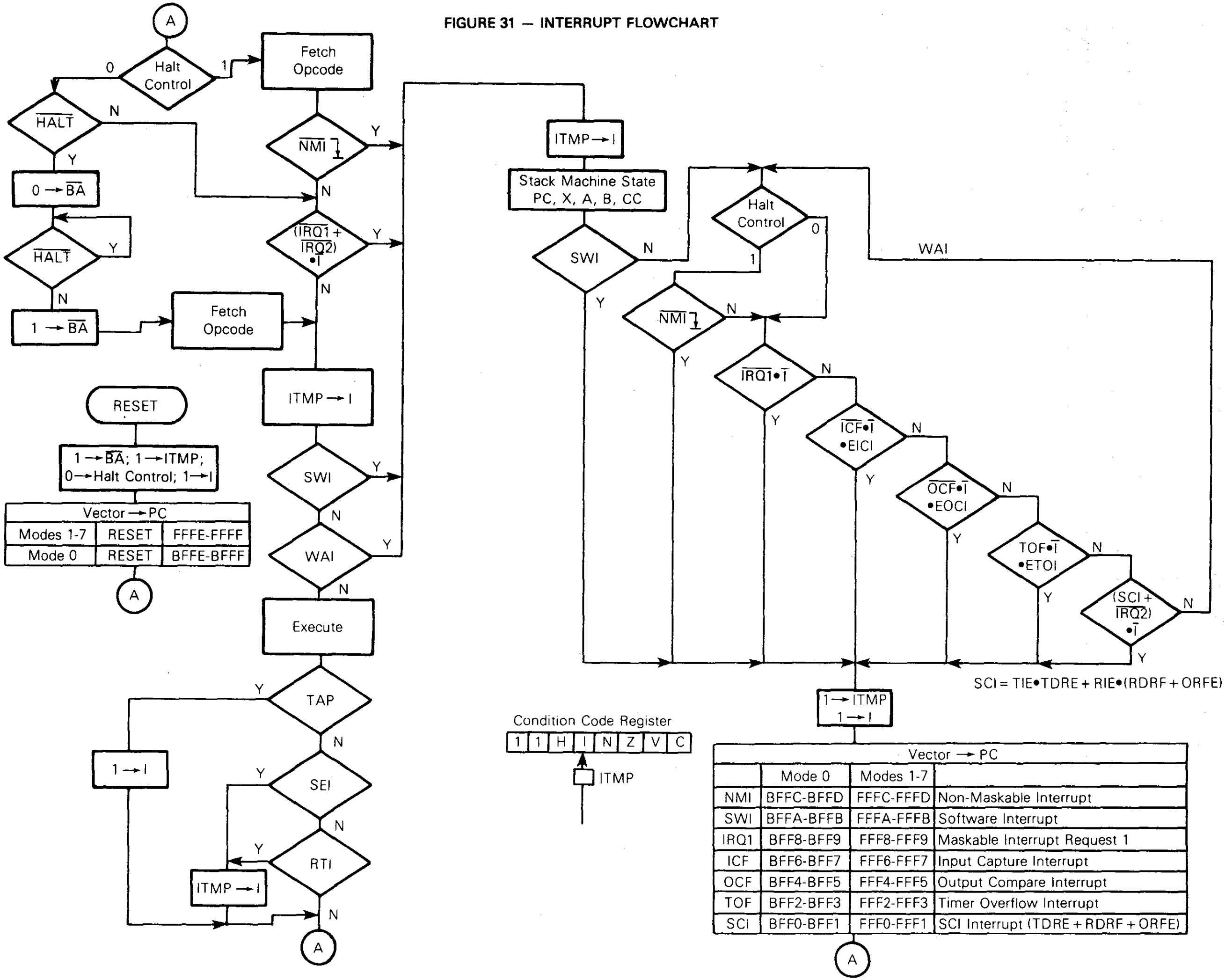
The interrupt flowchart is depicted in Figure 31. The Program Counter, Index Register, Accumulator A, Accumulator B, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. The general interrupt timing sequence is shown in Figure 32. The Interrupt $\overline{\text{HALT}}/\overline{\text{BA}}$ timing is illustrated in Figure 21 and 22.

TABLE 10 — MCU 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 IS3)
FFF6	FFF7	ICF (Input Capture)
FFF4	FFF5	OCF (Output Compare)
FFF2	FFF3	TOF (Timer Overflow)
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

*These locations are relocated at $BFF0-$BFFF in Mode 0.
**Highest priority.

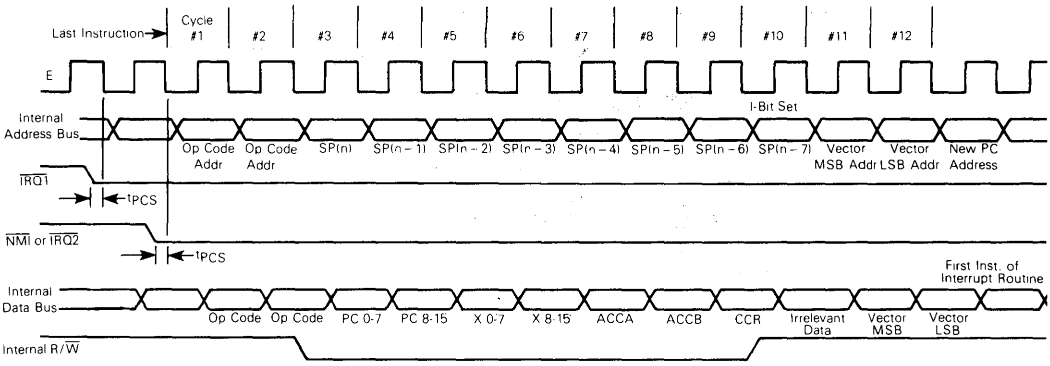
FIGURE 31 — INTERRUPT FLOWCHART



Vector → PC		
Modes 1-7	RESET	FFFE-FFFF
Mode 0	RESET	BFFE-BFFF

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 (TDRE + RDRF + ORFE)

FIGURE 32 — INTERRUPT SEQUENCE



PROGRAMMABLE TIMER

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 33.

TIMER CONTROL AND STATUS REGISTER ($08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable while bits 0-4 can be written. The three most significant bits provide the timer status and they indicate:

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

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

TIMER CONTROL AND STATUS REGISTER
(TSCR)

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

Bit 0 OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P21 if Bit 1 of the Port 2 Data Direction Register is set. It is cleared by reset.

Bit 1 IEDG Input Edge. IEDG is cleared by reset and controls which level transition will trigger a counter transfer to the Input Capture Register:
IEDG = 0 Transfer on a negative edge
IEDG = 1 Transfer on a positive edge

Bit 2 ETOI Enable Timer Overflow Interrupt. When set, an $\overline{\text{IRQ2}}$ interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 3 EOCI Enable Output Compare Interrupt. When set, an $\overline{\text{IRQ2}}$ interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 4 EICI Enable Input Capture Interrupt. When set, an $\overline{\text{IRQ2}}$ interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by reset.

Bit 5 TOF Timer Overflow Flag. TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) followed by reading the highest byte of the counter ($09), or by reset. Reading the counter at $15 will not clear TOF.

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

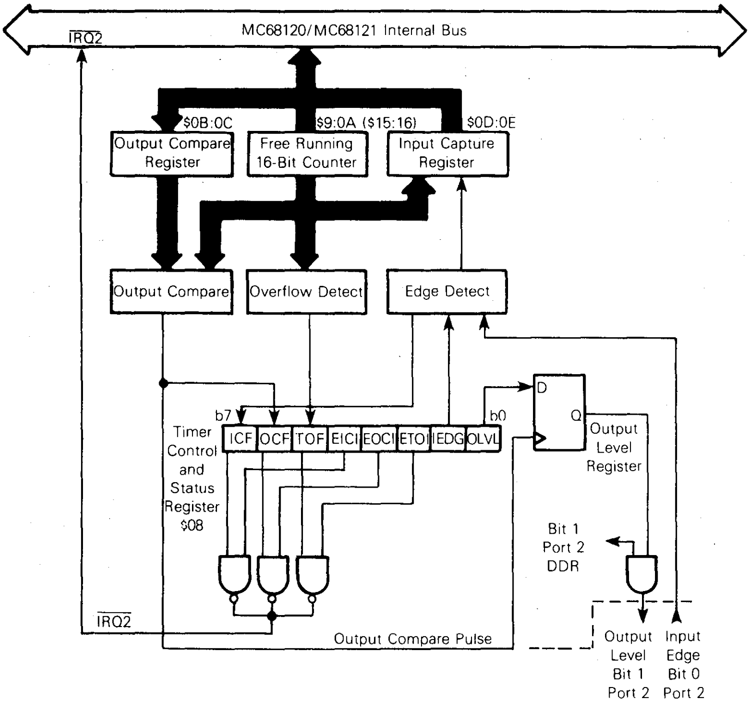
Bit 7 ICF Input Capture Flag. ICF is set to indicate a proper level transition. It is cleared by reading the TCSR (with ICF set) and then reading the Input Capture Register High Byte ($0D), or by reset.

COUNTER ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during reset and is a read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all 1's. The counter may also be read at location $15 and $16 to avoid the clearing of the TOF.

3

FIGURE 33 — PROGRAMMABLE TIMER — BLOCK DIAGRAM



OUTPUT COMPARE REGISTER ($0B:0C)

The Output Compare Register is a 16-bit Read/Write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E-cycle. When a match is found, OCF is set and OLVL is clocked to an output level register. If Port 2, bit 1 is configured as an output, OLVL will appear at P21. The Output Compare Register and OLVL can then be changed for the next compare. The compare function is inhibited for one cycle after a write to the high byte of the counter ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF by reset.

INPUT CAPTURE REGISTER ($0D:0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P20, even when configured as an output. An input capture can occur independently of ICF: the input capture register always contains the most current value regardless of whether ICF was previously set or not. Counter transfer is inhibited, however, between accesses of a double byte IPC read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

SERIAL COMMUNICATIONS INTERFACE (SCI)

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

WAKE-UP FEATURE

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to allow uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until the data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or by reset. Software must provide the required idle string between consecutive messages and prevent it within messages.

PROGRAMMABLE OPTIONS

The following features of the SCI are programmable:

- format: standard mark/space (NRZ) or Bi-phase

- clock: external or internal clock source
- Baud rate: one of four per E-clock frequency, or one-eighth of the external clock input to P22
- wake-up features: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P22

SERIAL COMMUNICATIONS REGISTERS

The Serial Communications Interface includes four addressable registers as depicted in Figure 34. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and read-only Receive Register. The shift registers are not accessible by software.

Rate and Mode Control Register ($10) — The Rate and Mode Control Register (RMCR) controls the SCI Baud rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared by reset. The two least significant bits control the Baud rate of the internal clock and the remaining two bits control the format and clock source.

RATE AND MODE CONTROL REGISTER (RMCR)

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

Bit 1: Bit 0 SS1:SS0 Speed Select. These two bits select the Baud rate when using the internal clock. Four rates may be selected which are a function of the IPC input frequency (E). Table 11 lists bit times and rates for three selected IPC frequencies.

Bit 3: Bit 2 CC1:CC0 Clock Control and Format Select. These two bits control the format and select the serial clock source. If CC1 is set, the Data Direction Register (DDR) value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 12 defines the format, clock source, and use of P22.

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

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

FIGURE 34 — SCI REGISTERS

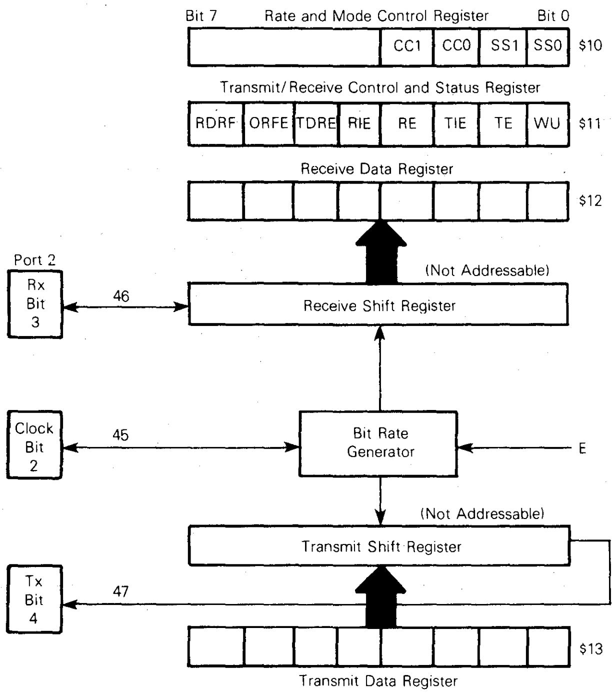


TABLE 11 — SCI BIT TIMES AND RATES

SS1:SS0	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	÷16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	÷128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	÷1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	÷4096	6.67ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

TABLE 12 — SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
0 0	Bi-Phase	Internal	Not Used
0 1	NRZ	Internal	Not Used
1 0	NRZ	Internal	Output
1 1	NRZ	External	Input

Transmit/Receive Control and Status Register ($11) — The Transmit/Receive Control and Status Register (TRCSR) controls the transmitter, receiver, wake-up features, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while only bits 0 to 4 are writable. The register is initialized to $20 by reset.

TRANSMIT/RECEIVE CONTROL AND STATUS REGISTER (TRCSR)

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

Bit 0 WU "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by reset. WU will not set if the line is idle.

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

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

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

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

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

Bit 6 ORFE Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun occurs when a new byte is ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun condition. ORFE is cleared by reading the TRCSR (with ORFE set) then reading the Receive Data Register, or by reset.

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

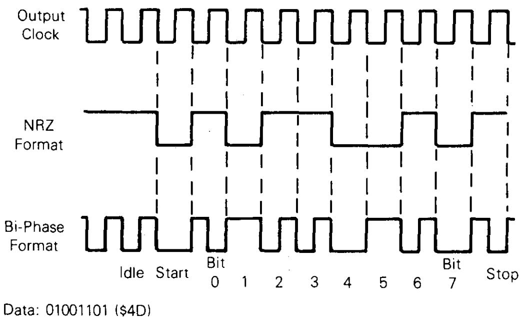
SERIAL OPERATIONS

The SCI is initialized by writing the control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register. When TE is set, the output of the Transmit Shift Register is connected to P24 and serial output is initiated by the transmission of a 9-bit preamble of 1's.

At this point one of two situations exist: 1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or 2) if a byte has been written to the Transmit Data Register (TDRE = 0), the byte will be transferred to the Transmit Shift Register (synchronized with the bit rate clock), TDRE will be set, and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. In Bi-phase format, the output toggles at the start of each bit and at half time when a "1" is sent. SCI data formats are illustrated in Figure 35. In receiving Bi-phase, a "1" is input when two transitions occur in less than 3/4 bit-time, and a "0" is input when more than 3/4 bit-time passes after a transition on P23.

FIGURE 35 — SCI DATA FORMATS



3

INSTRUCTION SET

The MC68120/MC68121 is upward source and object code compatible with the MC6800 processor and directly compatible with the M6801 Family processors.

PROGRAMMING MODEL

A programming model for the MC68120/MC68121 is shown in Figure 14. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

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

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

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

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

Condition Code Register — The Condition Code Register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits b6 and b7, are read as ones.

ADDRESSING MODES

The MC68120/MC68121 provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 13, 14, 15 and 16 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 17. With an input frequency (E) of 1 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 18 and a description of selected instructions is shown in Figure 38.

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

Direct Addressing — The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access (refer to Table 1). In most applications, this 256-byte area is reserved for frequently referenced data. Note that no direct addressing of internal control registers is possible in Mode 3.

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

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

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

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

TABLE 13 — INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	Mnemonic	Immed			Direct			Index			Extend			Inherent			Boolean/ Arithmetic Operation	Condition Codes					
																		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 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 1 of 2)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	●	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	●	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	ASLA													48	2	1		●	●	↕	↕	↕	↕
	ASLB													58	2	1		●	●	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●

TABLE 14 — ACCUMULATOR AND MEMORY INSTRUCTIONS (Sheet 2 of 2)

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

The Condition Code Register notes are listed after table 16.

TABLE 15 — JUMP AND BRANCH INSTRUCTIONS

		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 36	●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3					●	●	●	●	●	●
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1		●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1	See Special Operations — Figure 36	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

TABLE 16 — CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

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

CONDITION CODE SYMBOLS

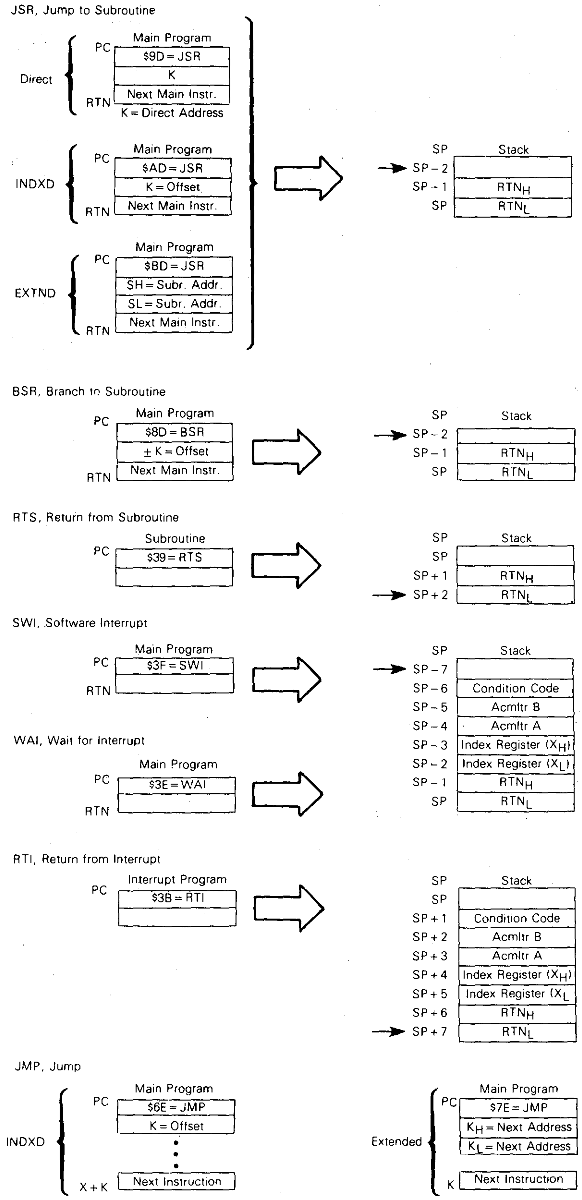
- H Half-carry from bit 3
- I Interrupt mask
- N Negative (sign bit)
- Z Zero (byte)
- V Overflow, 2's complement
- C Carry/Borrow from MSB
- R Reset Always
- S Set Always
- ↕ Affected
- ● Not Affected

TABLE 17 — INSTRUCTION EXECUTION TIMES IN E CYCLES

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

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

FIGURE 36 — SPECIAL OPERATIONS



Legend:

RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address

➔ = Stack pointer after execution
K = 8-bit unsigned value

CYCLE-BY-CYCLE OPERATION SUMMARY

Table 18 provides a detailed description of the information present on the Address Bus, Data Bus, and the R/$\overline{W}$ line during cycle of each instructions.

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

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

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 19. There are 220 valid machine codes, 34 unassigned codes and 2 reserved for test purposes.

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 1 of 5)

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

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 2 of 5)

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

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 3 of 5)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	$R/\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	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 —

3

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 4 of 5)

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

— Continued —

TABLE 18 — CYCLE BY CYCLE OPERATION (Sheet 5 of 5)

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

3

TABLE 19 — CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#	OP	MNEM	MODE	~	#
00	•				34	DES	INHER	3	1	68	ASL	INDXD	6	2	9C	CPX	DIR	5	2	D0	SUBB	DIR	3	2
01	NOP	INHER	2	1	35	TXS		3	1	69	ROL		6	2	9D	JSR		5	2	D1	CMPB		3	2
02	•				36	PSHA		3	1	6A	DEC		6	2	9E	LDS		4	2	D2	SBCB		3	2
03	•				37	PSHB		3	1	6B	•				9F	STS	DIR	4	2	D3	ADDD		5	2
04	LSRD		3	1	38	PULX		5	1	6C	INC		6	2	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	FB	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	CE	LDX	IMMED	3	3		*UNDEFINED OP CODE			
33	PULB	INHER	4	1	67	ASR	INDXD	6	2	9B	ADDA	DIR	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.

APPENDIX A
MC68120 CUSTOM ORDERING INFORMATION

A.0

Address $FFEF is Reserved for the Checksum value for the ROM, to be generated at the factory.

A.1 CUSTOM MC68120 ORDERING INFORMATION

The custom MC68120 specifications may be transmitted to Motorola in any of the following media:

A) EPROM(s)

B) MDOS diskette

The specification should be formatted and packaged, as indicated in the appropriate paragraph below, and mailed prepaid and insured with a cover letter (see Figure A-1) to:

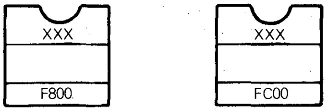
Motorola Inc.
MPU Marketing
3501 Ed Bluestein Blvd.
Austin, Texas 78721

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

A.2 EPROMs

MCM2708 and MCM2716 type EPROMs, programmed with the custom program (positive logic notation for address and data), may be submitted for pattern generation. The MC2708s must be clearly marked to indicate which PROM corresponds to which address space ($F800-$FBFF; $FC00-$FFFF). See Figure A-2 for recommended marking procedure.

FIGURE A-2



XXX = Customer ID

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

A.3 MDOS DISKETTE

The file name and start/end location should be written on the label.

FIGURE A-1

CUSTOMER NAME ____________________

ADDRESS ____________________

STATE __________ CITY __________ ZIP __________

PHONE __________ EXTENSION __________

CONTACT MS/MR ____________________

CUSTOMER PART # ____________________

PATTERN MEDIA
☐ 2708 EPROM
☐ 2716 EPROM
☐ Diskette (MDOS)

TEMPERATURE RANGE
☐ 0° to 70°C

PACKAGE TYPE
☐ Ceramic

MARKING
☐ Standard
☐ Special

(Note 1) ____________________

NOTE: (1) Other Media Require Prior Factory Approval

SIGNATURE ____________________

TITLE ____________________

MOTOROLA

MC146805E2

Advance Information

CMOS

(HIGH PERFORMANCE SILICON GATE)

8-BIT MICROPROCESSOR

3

8-BIT MICROPROCESSOR UNIT

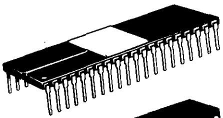
The MC146805E2 Microprocessor Unit (MPU) belongs to the M6805 Family of Microcomputers. This 8-bit fully static and expandable microprocessor contains a CPU, on-chip RAM, I/O, and TIMER. It is a low-power, low-cost processor designed for low-end to mid-range applications in the consumer, automotive, 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

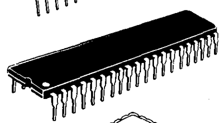
SOFTWARE FEATURES

- 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

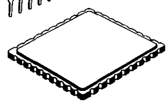


L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734



P SUFFIX
PLASTIC PACKAGE
CASE 711



Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT

$\overline{RESET}$	1 (2)	(1) 40	V_{DD}
$\overline{IRQ}$	2 (3)	(40) 39	OSC1
LI	3 (4)	(39) 38	OSC2
DS	4 (5)	(38) 37	TIMER
R/$\overline{W}$	5 (6)	(37) 36	PB0
AS	6 (7)	(36) 35	PB1
PA7	7 (8)	(35) 34	PB2
PA6	8 (9)	(34) 33	PB3
PA5	9 (10)	(33) 32	PB4
PA4	10 (11)	(32) 31	PB5
PA3	11 (12)	(31) 30	PB6
PA2	12 (13)	(30) 29	PB7
PA1	13 (14)	(29) 28	B0
PA0	14 (15)	(28) 27	B1
A12	15 (16)	(27) 26	B2
A11	16 (17)	(26) 25	B3
A10	17 (18)	(25) 24	B4
A9	18 (19)	(24) 23	B5
A8	19 (20)	(23) 22	B6
V_{SS}	20 (21)	(22) 21	B7

Pin numbers in parentheses represent equivalent Z suffix chip carrier pins.

GENERIC INFORMATION

Package Type	Frequency (MHz)	Temperature	Generic Number
Ceramic L Suffix	1.0 1.0	0°C to 70°C −40°C to 85°C	MC146805E2L MC146805E2CL
Cerdip S Suffix	1.0 1.0	0°C to 70°C −40°C to 85°C	MC146805E2S MC146805E2CS
Plastic P Suffix	1.0 1.0	0°C to 70°C −40°C to 85°C	MC146805E2P MC146805E2CP
Leadless Chip Carrier Z Suffix	1.0 1.0	0°C to 70°C −40°C to 85°C	MC146805E2Z MC146805E2CZ

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

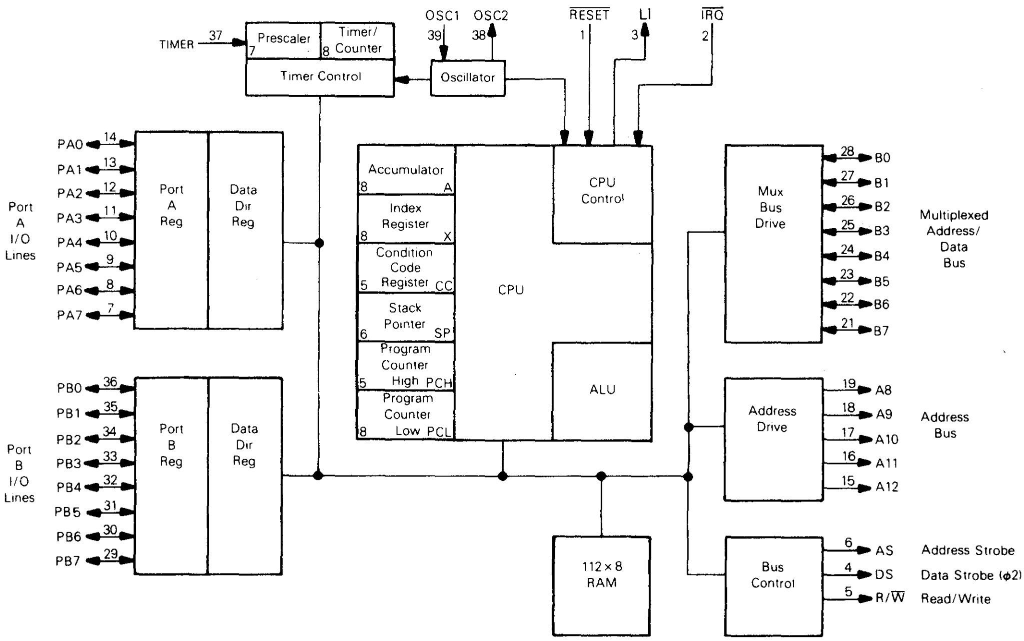
THERMAL CHARACTERISTICS

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}).

3

FIGURE 1 — MICROPROCESSOR BLOCK DIAGRAM



DC ELECTRICAL CHARACTERISTICS @ 3.0 V (V_{DD} = 3.0 Vdc, V_{SS} = 0, T_A = T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10.0\ \mu A$)	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
Total Supply Current (C_L = 50 pF – No dc Loads, t_{cyc} = 5 μs) Run (V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V)	I_{DD}	–	1.3	mA
Wait (Test Conditions – See Note Below)	I_{DD}	–	200	μA
Stop (Test Conditions – See Note Below)	I_{DD}	–	100	μA
Output High Voltage (I_{Load} = 0.25 mA) A8-A12, B0-B7, DS, AS, R/$\overline{W}$	V_{OH}	2.7	–	V
(I_{Load} = 0.1 mA) PA0-PA7, PB0-PB7	V_{OH}	2.7	–	V
Output Low Voltage (I_{Load} = 0.25 mA) A8-A12, B0-B7, PB0-PB7, DS, AS, R/$\overline{W}$, PA0-PA7	V_{OL}	–	0.3	V
Input High Voltage PA0-PA7, PB0-PB7, B0-B7	V_{IH}	2.1	–	V
TIMER, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	2.5	–	V
OSC1	V_{IH}	2.1	–	V
Input Low Voltage (All Inputs)	V_{IL}	–	0.5	V
Frequency of Operation Crystal	f_{osc}	–	1.0	MHz
External Clock	f_{osc}	dc	1.0	MHz
Input Current $\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	μA
Hi-Z Output Leakage PA0-PA7, PB0-PB7, B0-B7	I_{TSL}	–	±10	μA
Capacitance $\overline{RESET}$, $\overline{IRQ}$, TIMER	C_{in}	–	8.0	pF
Capacitance DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode I_{DD} is affected linearly by this capacitance.

DC ELECTRICAL CHARACTERISTICS @ 5.0 V (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0, T_A = T_L to T_H, unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \le 10.0\ \mu A$)	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, $\overline{IRQ}$, $\overline{RESET}$	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				
$\overline{RESET}$, $\overline{IRQ}$, TIMER, OSC1	I_{in}	–	±1	μA
Hi-Z Output Leakage				
PA0-PA7, PB0-PB7, B0-B7	I_{TSI}	–	±10	μA
Capacitance				
$\overline{RESET}$, $\overline{IRQ}$, TIMER	C_{in}	–	8.0	pF
Capacitance				
DS, AS, R/$\overline{W}$, A8-A12, PA0-PA7, PB0-PB7, B0-B7	C_{out}	–	12.0	pF

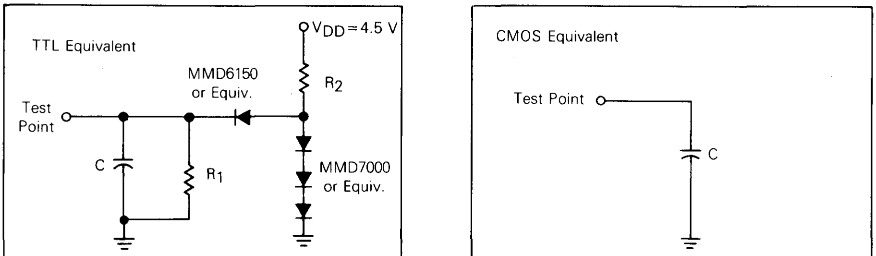
NOTE: Test conditions for Quiescent Current Values are:
Port A and B programmed as inputs.
V_{IL} = 0.2 V for PA0-PA7, PB0-PB7, and B0-B7.
V_{IH} = V_{DD} – 0.2 V for $\overline{RESET}$, $\overline{IRQ}$, and TIMER.
OSC1 input is a squarewave from V_{SS} + 0.2 V to V_{DD} – 0.2 V.
OSC2 output load (including tester) is 35 pF maximum.
Wait mode (I_{DD}) is affected linearly by this capacitance.

TABLE 1 — CONTROL TIMING ($V_{SS}=0$, $T_A=T_L$ to T_H)

Characteristics	Symbol	$V_{DD}=3.0$ V $f_{osc}=1$ MHz Min	Typ	Max	$V_{DD}=5.0$ V ± 10% $f_{osc}=5.0$ MHz 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.

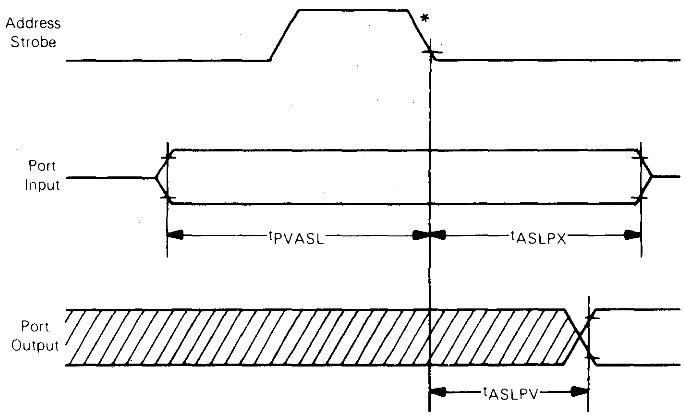
FIGURE 2 — EQUIVALENT TEST LOADS



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, $V_{DD}=5.0\pm10\%$
$T_A=T_L$ to T_H, C_L on Port=50 pF, $f_{osc}=5$ MHz)



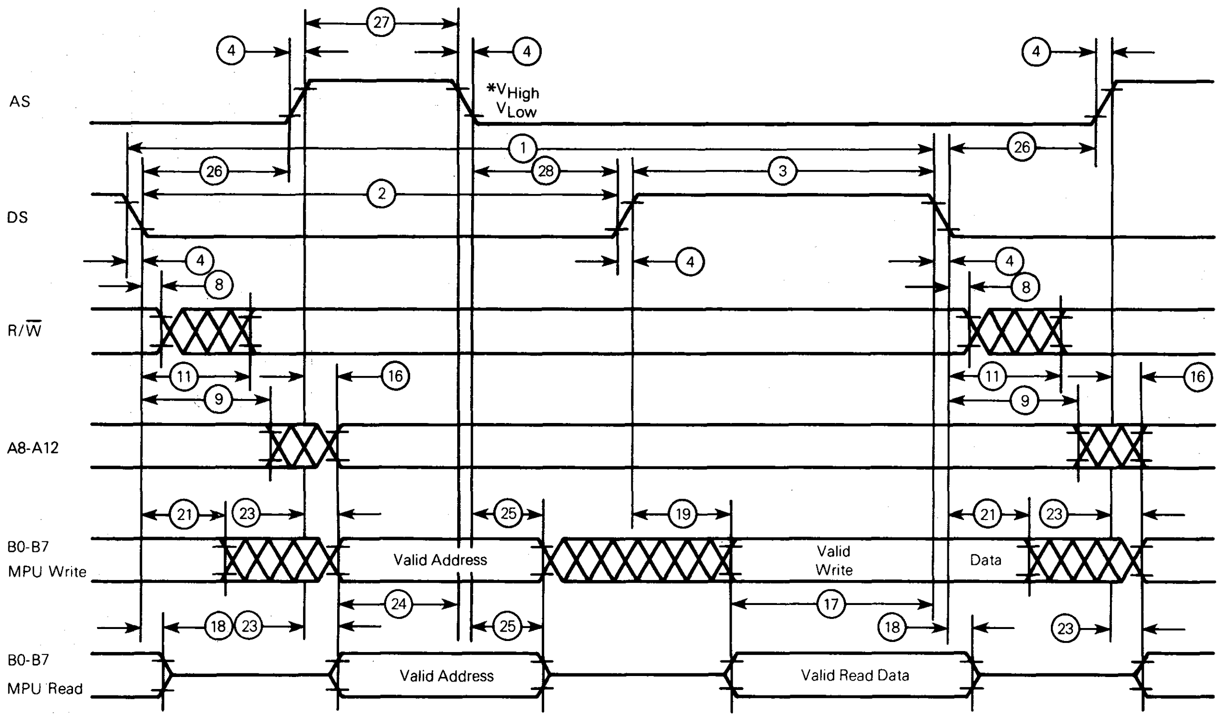
* The address strobe of the first cycle of the next instruction.

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 $\pm10\%$, 1 TTL and 130 pF Load		Unit
			Min	Max	Min	Max	
1	Cycle Time	t_{cyc}	5000	dc	1000	dc	ns
2	Pulse Width, DS Low	PW_{EL}	2800	–	560	–	ns
3	Pulse Width, DS High	PW_{EH}	1800	–	375	–	ns
4	Clock Transition	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold	t_{RWH}	10	–	10	–	ns
9	Non-Muxed Address Hold	t_{AH}	800	–	100	–	ns
11	R/$\overline{W}$ Delay from DS Fall	t_{AD}	–	500	–	300	ns
16	Non-Muxed Address Delay from AS Rise	t_{ADH}	0	200	0	100	ns
17	MPU Read Data Setup	t_{DSR}	200	–	115	–	ns
18	Read Data Hold	t_{DHR}	0	800	0	160	ns
19	MPU Data Delay, Write	t_{DDW}	–	0	–	120	ns
21	Write Data Hold	t_{DHW}	800	–	55	–	ns
23	Muxed Address Delay from AS Rise	t_{BHD}	0	250	0	120	ns
24	Muxed Address Valid to AS Fall	t_{ASL}	600	–	55	–	ns
25	Muxed Address Hold	t_{AHL}	250	750	60	180	ns
26	Delay DS Fall to AS Rise	t_{ASD}	800	–	160	–	ns
27	Pulse Width, AS High	PW_{ASH}	850	–	175	–	ns
28	Delay, AS Fall to DS Rise	t_{ASED}	800	–	160	–	ns

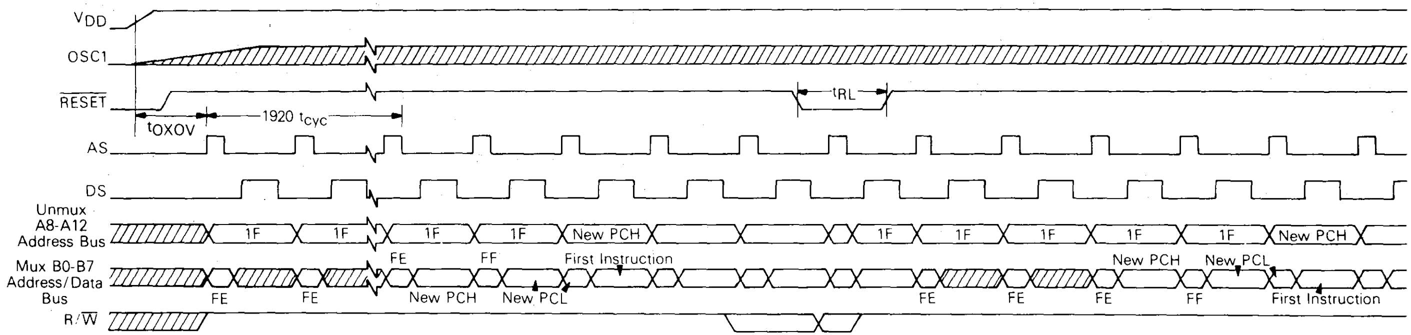
3

FIGURE 4 — MC146805E2 BUS TIMING

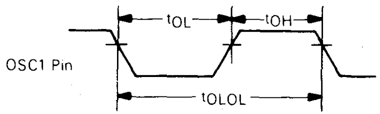


* $V_{High} = 2.0$ V, $V_{Low} = 0.5$ V for $V_{DD} = 3$ V for outputs only.
$V_{High} = V_{DD} - 2.0$ V, $V_{Low} = 0.8$ V for $V_{DD} = 5$ V $\pm$ 10% for outputs only.

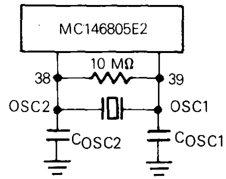
FIGURE 5 — POWER-ON RESET AND $\overline{RESET}$ TIMING



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

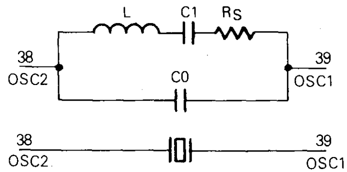
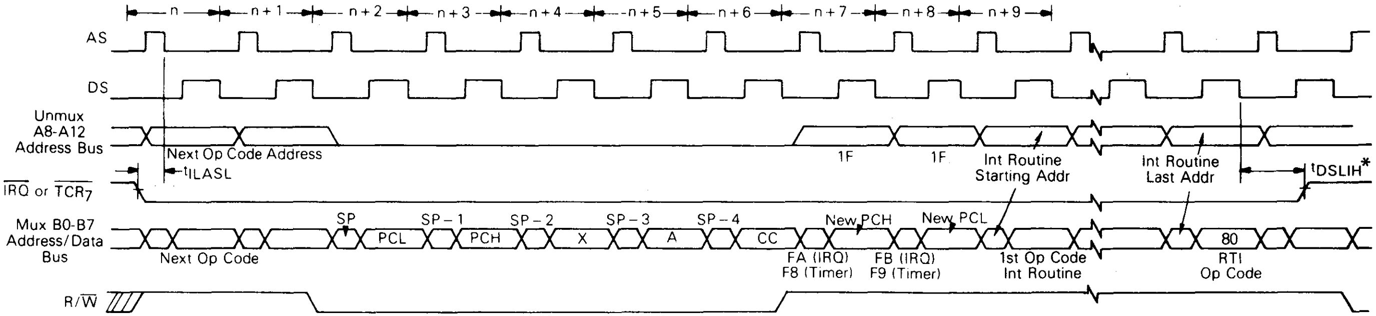


FIGURE 6 — $\overline{IRQ}$ AND $\overline{TCR_7}$ INTERRUPT TIMING



*t_{DSLIH} — The interrupting device must release the $\overline{IRQ}$ line within this time to prevent subsequent recognition of the same interrupt.

FIGURE 7 — TIMER INTERRUPT AFTER WAIT INSTRUCTION: TIMING

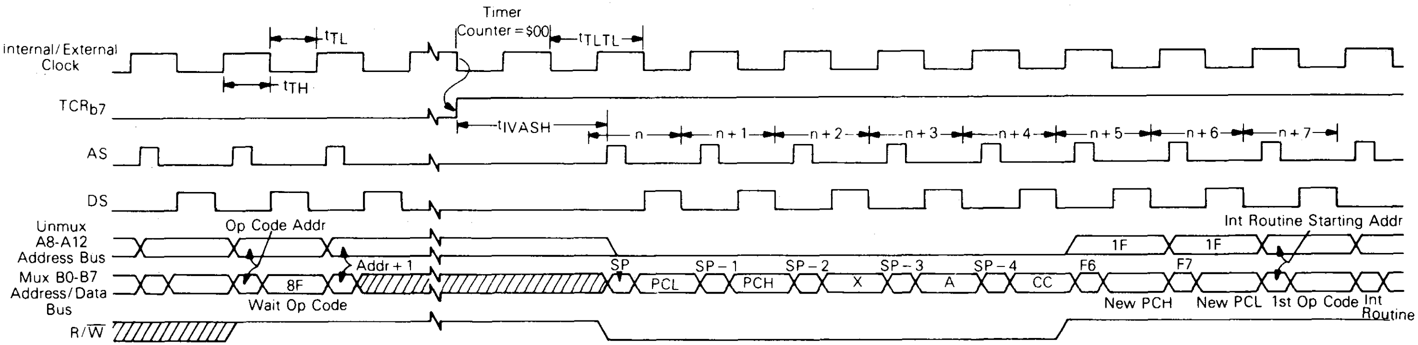
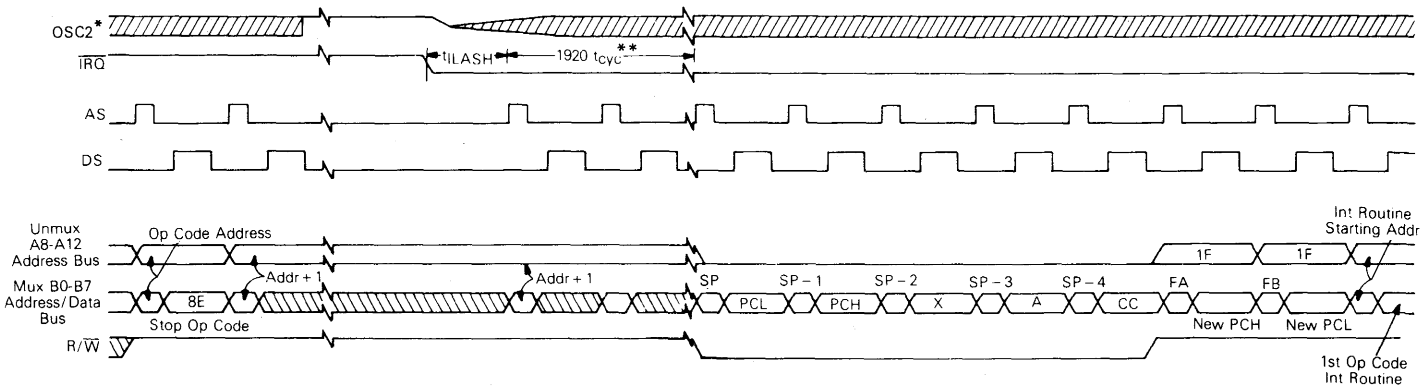


FIGURE 8 — INTERRUPT RECOVERY FROM STOP INSTRUCTION: TIMING



*Represents the internal gating of the OSC1 input pin.

**t_{cyc} is one instruction cycle (for $f_{osc} = 5$ MHz, $t_{cyc} = 1$ μs)

3

FUNCTIONAL PIN DESCRIPTION

V_{DD} AND V_{SS}

V_{DD} and V_{SS} provide power to the chip. V_{DD} provides power and V_{SS} is ground.

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

$\overline{IRQ}$ is both a level-sensitive and edge-sensitive input which can be used to request an interrupt sequence. The MPU completes the current instruction before it responds to the request. If $\overline{IRQ}$ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MPU begins an interrupt sequence at the end of the current instruction. The interrupt circuit recognizes both a "wire ORed" level as well as pulses on the $\overline{IRQ}$ line (see Interrupt section for more details). $\overline{IRQ}$ requires an external resistor to V_{DD} for "wire OR" operation.

$\overline{RESET}$

The $\overline{RESET}$ input is not required for start-up but can be used to reset the MPU internal state and provide an orderly software start-up procedure. Refer to the Reset section for a detailed description.

TIMER

The TIMER input is used for clocking the on-chip timer. Refer to Timer section for a detailed description.

AS (ADDRESS STROBE)

Address strobe (AS) is an output strobe used to indicate the presence of an address on the 8-bit multiplexed bus. The AS line is used to demultiplex the eight least significant address bits from the data bus. A latch controlled by address strobe should capture addresses on the negative edge. This output is capable of driving one standard TTL load and 130 pF and is available at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP states.

DS (DATA STROBE)

This output is used to transfer data to or from a peripheral or memory. DS occurs anytime the MPU does a data read or write. DS also occurs when the MPU does a data transfer to or from the MPU internal memory. Refer to Table 2 and Figure 4 for timing characteristics. This output is capable of driving one standard TTL load and 130 pF. DS is a continuous signal at $f_{OSC} \div 5$ when the MPU is not in the WAIT or STOP state. Some bus cycles are redundant reads of opcode bytes.

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

The R/$\overline{W}$ output is used to indicate the direction of data transfer for both internal memory and I/O registers, and external peripheral devices and memories. This output is used to indicate to a selected peripheral whether the MPU is going to read or write data on the next data strobe (R/$\overline{W}$ low = processor write; R/$\overline{W}$ high = processor read). The R/$\overline{W}$ output is capable of driving one standard TTL load and 130 pF. The normal standby state is read (high).

A8-A12 (HIGH ORDER ADDRESS LINES)

The A8-A12 output lines constitute the higher order 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/$\overline{W}$ 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.

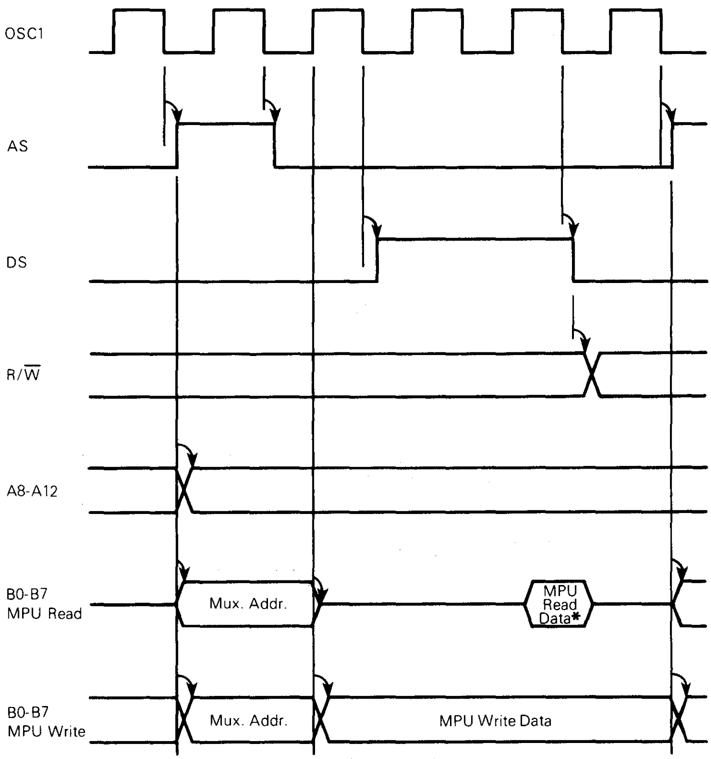
LI (LOAD INSTRUCTION)

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

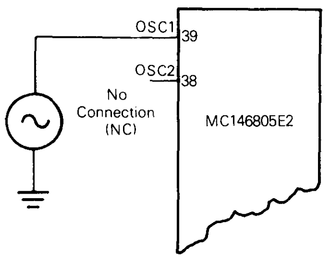
These eight pins constitute input/output port A. Each line is individually programmed to be either an input or output under software control via its data direction register as shown in Figure 11(b). An I/O pin is programmed as an output when the corresponding DDR bit is set to a "1", and as an input 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

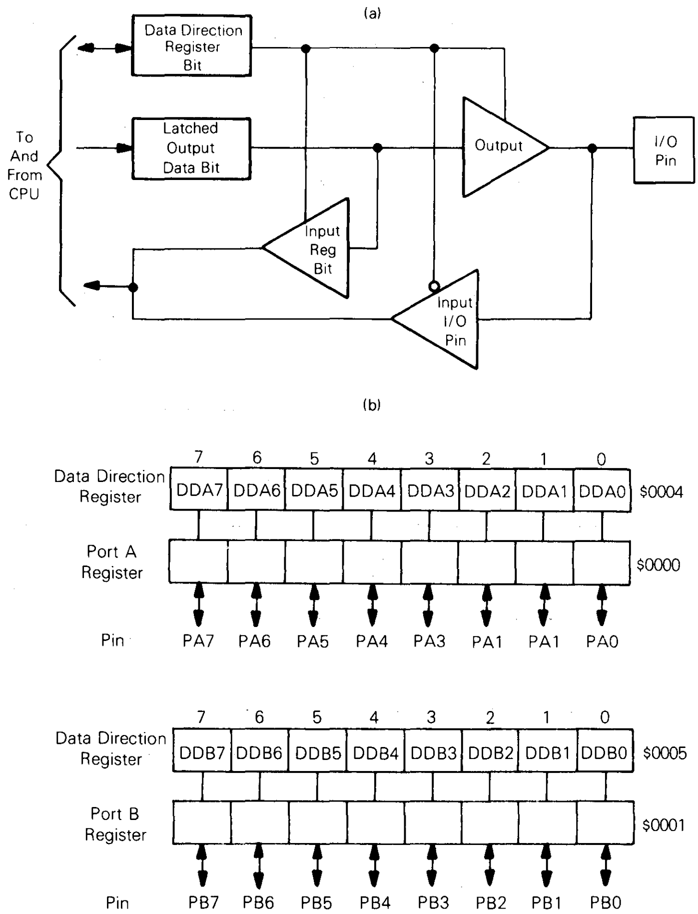


TABLE 3 — I/O PIN FUNCTIONS

$R/\overline{W}$	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

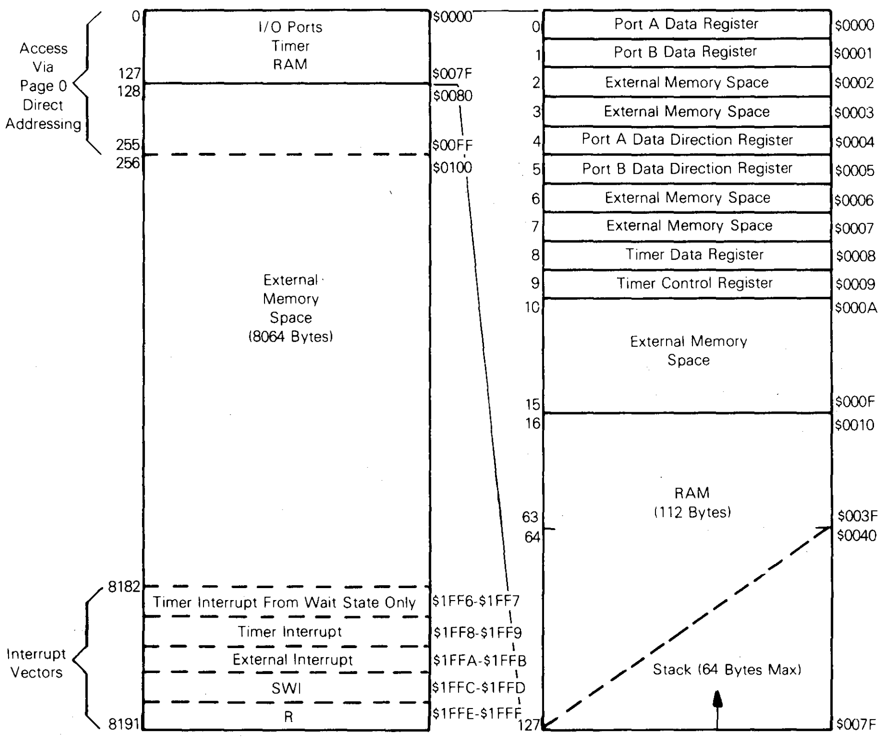


FIGURE 13 — PROGRAMMING MODEL

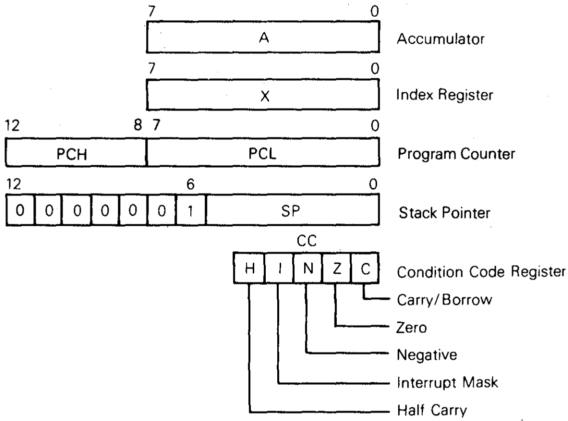
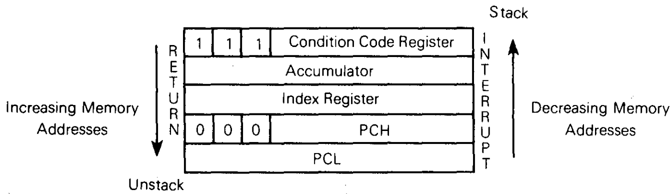


FIGURE 14 — STACKING ORDER



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.

$\overline{RESET}$ (PIN #1)

The $\overline{RESET}$ input pin is used to reset the MPU and provide an orderly software start-up procedure. When using the external reset mode, the $\overline{RESET}$ pin must stay low for a minimum of one t_{RL}. The $\overline{RESET}$ 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 $\overline{RESET}$, hardware interrupts do not cause the current instruction excution to be halted, but are considered pending until the current instruction execution is complete.

When the current instruction is complete, the processor checks all pending hardware interrupts and if unmasked, proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Note that masked interrupts are latched for later interrupt service.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed as any other instruction. Refer to Figure 15 for the interrupt and instruction processing sequence.

TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then each time the timer decrements to zero (transitions from $01 to $00) an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the I bit in the condition code register is set. This masks further interrupts until the present one is serviced. The processor now vectors to the timer interrupt service routine. The address for this service routine is specified by the contents of $1FF8 and $1FF9 unless the processor is in a WAIT mode, in which case users of mask versions BP4XX-XX and AW9XXXX should refer to the appendix for additional information regarding exceptions to this function. The contents of $1FF6 and $1FF7 specify the service routine. Also, software must be used to clear the timer interrupt request bit (TCR7). At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

EXTERNAL INTERRUPT

If the interrupt mask bit of the condition code register is cleared and the external interrupt pin $\overline{IRQ}$ 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{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 $\overline{IRQ}$ 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/$\overline{W}$ line goes to a high state.

3

FIGURE 15 — $\overline{\text{RESET}}$ AND INTERRUPT PROCESSING FLOWCHART

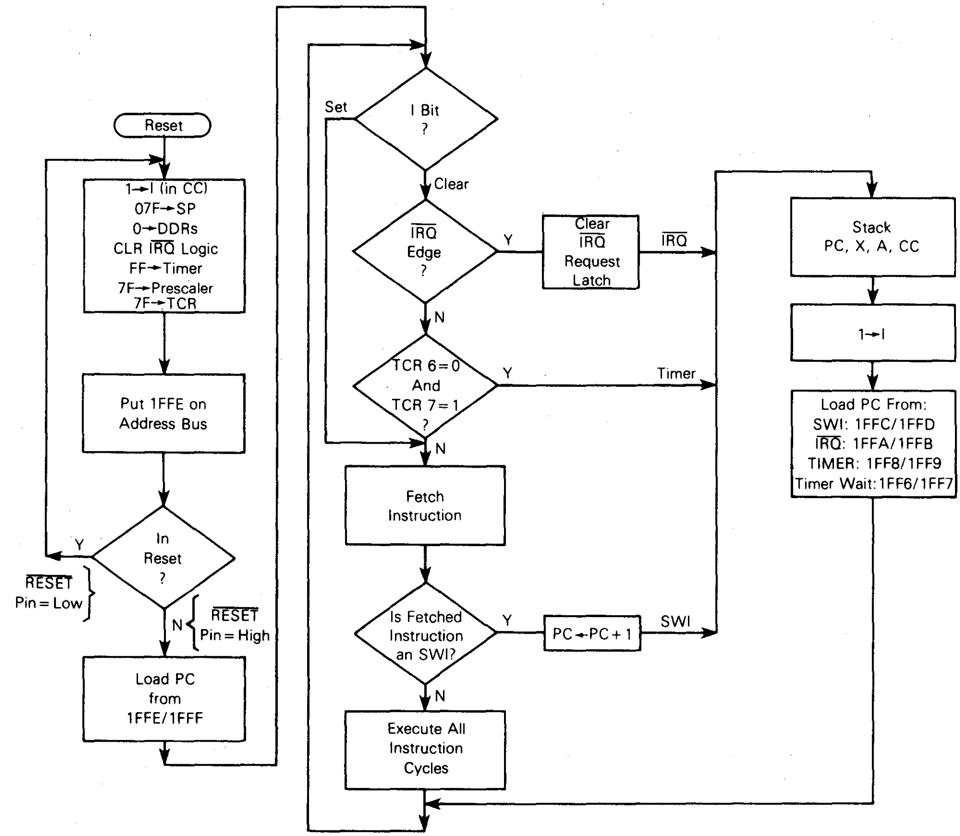
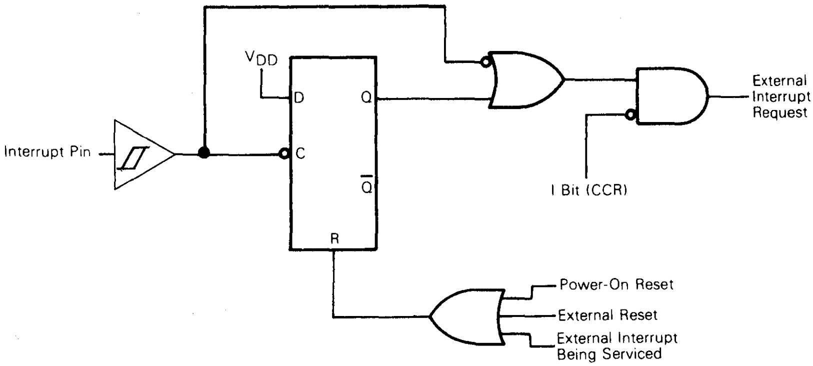


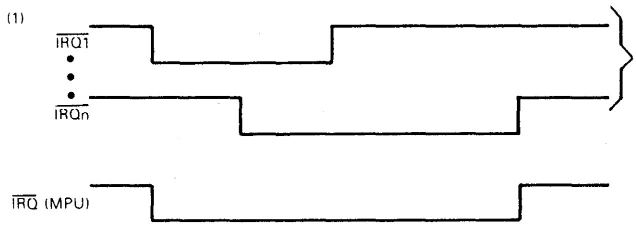
FIGURE 16 — EXTERNAL INTERRUPT

(a) Interrupt Functional Diagram



(b) Interrupt Mode Diagram

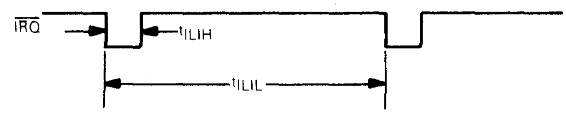
(1)



Wire ORed Condition

If after servicing an interrupt the $\overline{IRQ}$ remains low, then the next interrupt is recognized.

(2)



Pulse Condition

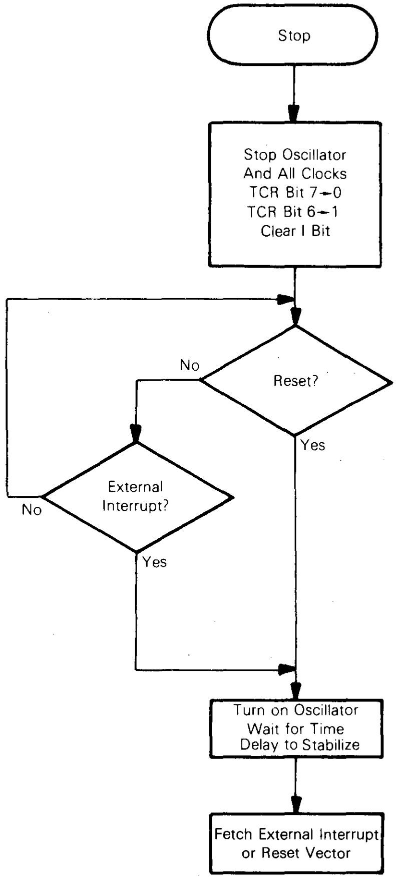
The minimum pulse width (t_{ILIH}) is one t_{cyc}. The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 20 t_{cyc} cycles.

3

The multiplexed address/data bus goes to the data input state (as shown in Figure 8). The high order address lines remain at the address of the next instruction. The MPU remains in the STOP mode until an external interrupt or reset occurs.

During the STOP mode, timer control register (TCR) bits 6 and 7 are altered to remove any pending timer interrupt requests and to disable any further timer interrupts. External interrupts are enabled in the condition code register. All other registers and memory remain unaltered. All I/O lines remain unchanged.

FIGURE 17 — STOP FUNCTION FLOWCHART



WAIT

The WAIT instruction places the MC146805E2 in a low power consumption mode, but the WAIT mode consumes somewhat more power than the STOP mode; refer to Table 1. In the WAIT function, the internal clock is disabled from all internal circuitry except the timer circuit; refer to Figure 18. Thus, all internal processing is halted except the timer which is allowed to count in a normal sequence. The R/$\overline{W}$ line goes to a high state, the multiplexed address/data bus goes to the data input state, and the DS and AS lines go to the low state (as shown in Figure 7). The high order address lines remain at the address of the next instruction. The MPU remains in this state until an external interrupt, timer interrupt, or a reset occurs.

During the WAIT mode, the I bit in the condition code register is cleared to enable interrupts. All other registers, memory, and I/O lines remain in their last state. The timer may be enabled to allow a periodic exit from the WAIT mode. If an external and a timer interrupt occur at the same time, the external interrupt is serviced first; then, if the timer interrupt request is not cleared in the external interrupt routine, the normal timer interrupt (not the timer WAIT interrupt) is serviced since the MPU is no longer in the WAIT mode.

TIMER

The MPU timer contains a single 8-bit software programmable counter (timer data register) with 7-bit software selectable prescaler. Figure 19 shows a block diagram of the timer. The counter may be preset under program control and decrements towards zero. When the counter decrements to zero, the timer interrupt request bit, i.e., bit 7 of the timer control register (TCR), is set. Then if the timer interrupt is not masked, i.e., bit 6 of the TCR and the I bit in the condition code register are both cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack, and then fetches the timer interrupt vector from locations \$1FF8 and \$1FF9 in order to begin servicing the interrupt. If the MPU is interrupted by the timer while in the WAIT mode, the interrupt vector fetch would be from locations \$1FF6 and \$1FF7.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external input clocks since the timer interrupt request bit was set. The counter may be read at any time by the processor without disturbing the count. The content of the counter becomes stable prior to the read portion of a cycle and does not change during the read. The timer interrupt request bit remains set until cleared by the software. If a read occurs before the timer interrupt is serviced, the interrupt is lost. TCR7 may also be used as a scanned status bit in a non-interrupt mode of operation (TCR6 = 1).

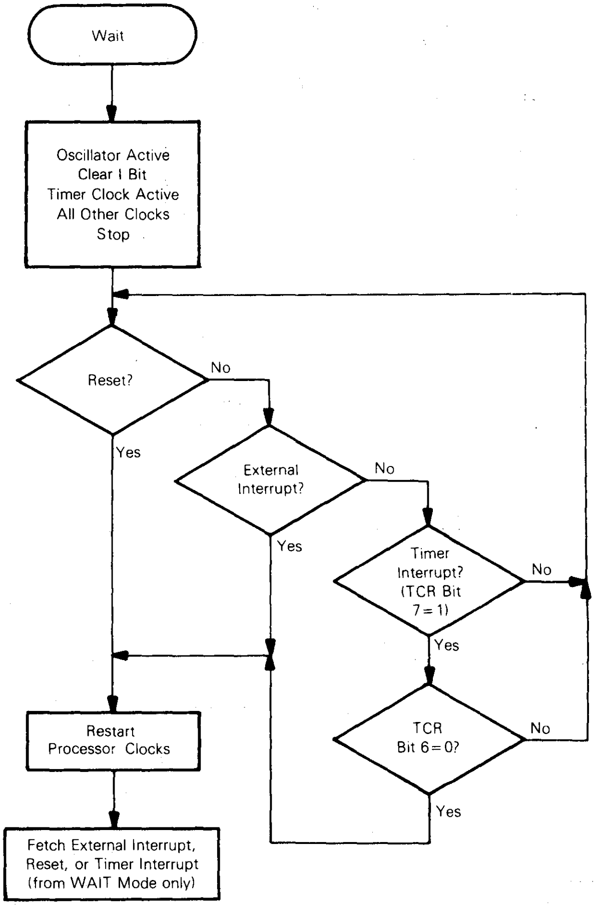
The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1, and bit 2 of the TCR are programmed to choose the appropriate prescaler output which is used as the counter input. The processor cannot write into or read from the prescaler; however, its contents are cleared to all "0s" by the write operation into TCR when bit 3 of the written data equals 1, which allows for truncation-free counting.

The timer input can be configured for three different operating modes, plus a disable mode, depending on the value written to the TCR4, TCR5 control bits. Refer to the Timer Control Register section.

TIMER INPUT MODE 1

If TCR4 and TCR5 are both programmed to a "0", the input to the timer is from an internal clock and the external TIMER input is disabled. The internal clock mode can be

FIGURE 18 — WAIT FUNCTION FLOWCHART



used for periodic interrupt generation, as well as a reference in frequency and event measurement. The internal clock is the instruction cycle clock and is coincident with address strobe (AS) except during a WAIT instruction. During a WAIT instruction the AS pin goes to a low state but the internal clock to the timer continues to run at its normal rate.

TIMER INPUT MODE 2

With TCR4=1 and TCR5=0, the internal clock and the TIMER input pin are ANDed to form the timer input signal. This mode can be used to measure external pulse widths. The external timer input pulse simply turns on the internal clock for the duration of the pulse. The resolution of the count in this mode is ± 1 clock and therefore accuracy improves with longer input pulse widths.

TIMER INPUT MODE 3

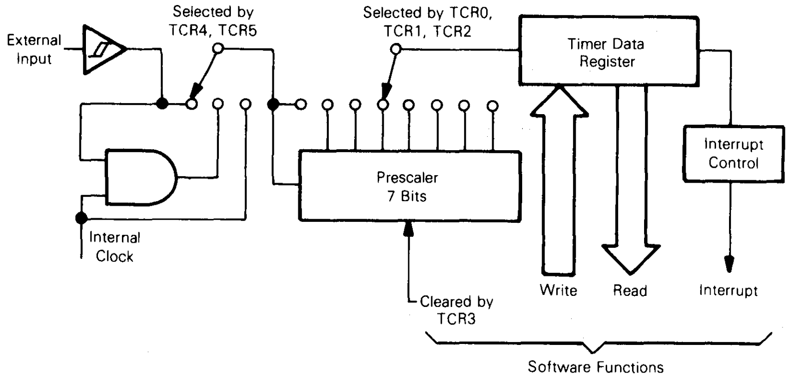
If TCR4=0 and TCR5=1, then all inputs to the timer are disabled.

TIMER INPUT MODE 4

If TCR4=1 and TCR5=1, the internal clock input to the timer is disabled and the TIMER input pin becomes the input to the timer. The external TIMER pin can, in this mode, be used to count external events as well as external frequencies for generating periodic interrupts.

Figure 19 shows a block diagram of the timer subsystem.

FIGURE 19 — TIMER BLOCK DIAGRAM



NOTES:
1. Prescaler and timer data register are clocked on the falling edge of the internal clock (AS) or external input.
2. Timer data register is written to during data strobe (DS) and counts down continuously.

TIMER CONTROL REGISTER (TCR)

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

All bits in this register except bit 3 are read/write bits.

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

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.

TCR3 — Timer Prescaler Reset bit: writing a "1" to this bit resets the prescaler to zero. A read of this location always indicates a "0" (unaffected by $\overline{\text{RESET}}$).

TCR2, TCR1, TCR0 — Prescaler address bits: decoded to select one of eight outputs of the prescaler (unaffected by $\overline{\text{RESET}}$).

Prescaler

TCR2	TCR1	TCR0	Result
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

INSTRUCTION SET

The MPU has a set of 61 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 4.

READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to Table 5.

BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met, otherwise no operation is performed. Branch instructions are two byte instructions. Refer to Table 6.

BIT MANIPULATION INSTRUCTIONS

The MPU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 7.

CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8.

ALPHABETICAL LISTING

The complete instruction set is given in alphabetical order in Table 9.

OPCODE MAP SUMMARY

Table 10 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MPU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory. Table 9 shows the addressing modes for each instruction, with the effects each instruction has on the condition code register. An opcode map is shown in Table 10.

The term "effective address" or EA is used in describing the various addressing modes, and is defined as the address to or from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate "contents of," an arrow indicates "is replaced by," and a colon indicates concatenation of two bytes.

INHERENT

In inherent instructions all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

IMMEDIATE

In immediate addressing, the operand is contained in the byte immediatley following the opcode. Immediate addressing is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

$$EA = PC + 1;\ PC \leftarrow PC + 2$$

DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction. This includes all on-chip RAM and I/O registers and up to 128 bytes of off-chip ROM. Direct addressing is efficient in both memory and speed.

$$EA = (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow 0;\ \text{Address Bus Low} \leftarrow (PC + 1)$$

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.

$$EA = X + (PC + 1);\ PC \leftarrow PC + 2$$
$$\text{Address Bus High} \leftarrow K;\ \text{Address Bus Low} \leftarrow X + (PC + 1)$$
$$\text{where: } K = \text{The carry from the addition of } X + (PC + 1)$$

3

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

Function	Mnemonic	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	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			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IFF Bit n is Set	BRSET n (n=0...7)	–	–	–	2•n	3	5
Branch IFF Bit n is Clear	BRCLR n (n=0...7)	–	–	–	01+2•n	3	5
Set Bit n	BSET n (n=0...7)	10+2•n	2	5	–	–	–
Clear Bit n	BCLR n (n=0...7)	11+2•n	2	5	–	–	–

TABLE 8 — CONTROL INSTRUCTIONS

Function	Mnemonic	Inherent		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	10
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

TABLE 9 — INSTRUCTION SET

Mnemonic	Addressing Modes										Condition Codes				
	Inherent	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
ADD		X	X	X		X	X	X			Λ	●	Λ	Λ	Λ
AND		X	X	X		X	X	X			●	●	Λ	Λ	●
ASL	X		X			X	X				●	●	Λ	Λ	Λ
ASR	X		X			X	X				●	●	Λ	Λ	Λ
BCC					X						●	●	●	●	●
BCLR									X		●	●	●	●	●
BCS					X						●	●	●	●	●
BEQ					X						●	●	●	●	●
BHCC					X						●	●	●	●	●
BHCS					X						●	●	●	●	●
BHI					X						●	●	●	●	●
BHS					X						●	●	●	●	●
BIH					X						●	●	●	●	●
BIL					X						●	●	●	●	●
BIT		X	X	X		X	X	X			●	●	Λ	Λ	●
BLO					X						●	●	●	●	●
BLS					X						●	●	●	●	●
BMC					X						●	●	●	●	●
BMI					X						●	●	●	●	●
BMS					X						●	●	●	●	●
BNE					X						●	●	●	●	●
BPL					X						●	●	●	●	●
BRA					X						●	●	●	●	●
BRN					X						●	●	●	●	●
BRCLR										X	●	●	●	●	Λ
BRSET										X	●	●	●	●	Λ
BSET									X		●	●	●	●	●
BSR					X						●	●	●	●	●
CLC	X										●	●	●	●	0
CLI	X										●	0	●	●	●
CLR	X		X			X	X				●	●	0	1	●
CMP		X	X	X		X	X	X			●	●	Λ	Λ	Λ
COM	X		X			X	X				●	●	Λ	Λ	1
CPX		X	X	X		X	X	X			●	●	Λ	Λ	Λ
DEC	X		X			X	X				●	●	Λ	Λ	●
EOR		X	X	X		X	X	X			●	●	Λ	Λ	●
INC	X		X			X	X				●	●	Λ	Λ	●
JMP			X	X		X	X	X			●	●	●	●	●
JSR			X	X		X	X	X			●	●	●	●	●
LDA		X	X	X		X	X	X			●	●	Λ	Λ	●
LDX		X	X	X		X	X	X			●	●	Λ	Λ	●
LSL	X		X			X	X				●	●	Λ	Λ	Λ
LSR	X		X			X	X				●	●	0	Λ	Λ
NEG	X		X			X	X				●	●	Λ	Λ	Λ
NOP	X										●	●	●	●	●
ORA		X	X	X		X	X	X			●	●	Λ	Λ	●
ROL	X		X			X	X				●	●	Λ	Λ	Λ
ROR	X		X			X	X				●	●	Λ	Λ	Λ
RSP	X										●	●	●	●	●
RTI	X										?	?	?	?	?
RTS	X										●	●	●	●	●
SBC		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SEC	X										●	●	●	●	1
SEI	X										●	1	●	●	●
STA			X	X		X	X	X			●	●	Λ	Λ	●
STOP	X										●	0	●	●	●
STX			X	X		X	X	X			●	●	Λ	Λ	●
SUB		X	X	X		X	X	X			●	●	Λ	Λ	Λ
SWI	X										●	1	●	●	●
TAX	X										●	●	●	●	●
TST	X		X			X	X				●	●	Λ	Λ	●
TXA	X										●	●	●	●	●
WAIT	X										●	0	●	●	●

Condition Code Symbols

H	Half Carry (From Bit 3)	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

3

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 DIH	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

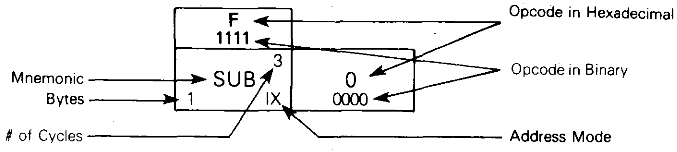


FIGURE 20 — CONNECTION TO CMOS PERIPHERALS

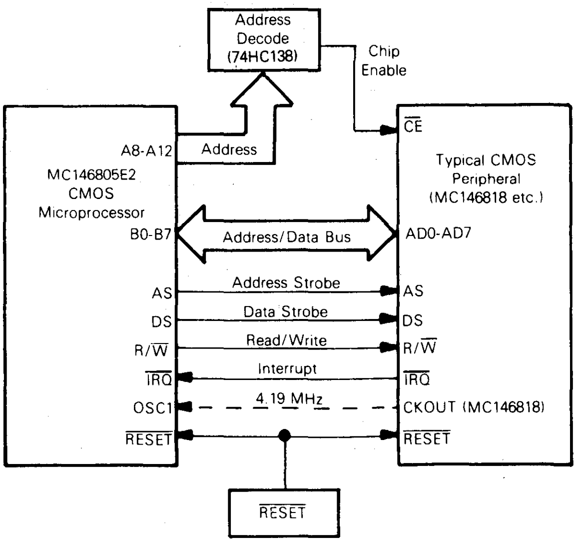
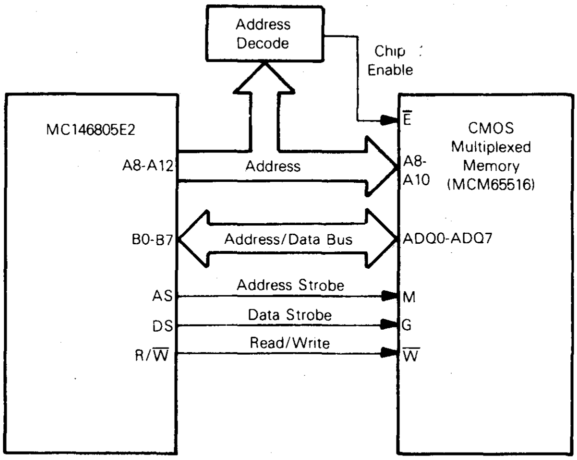
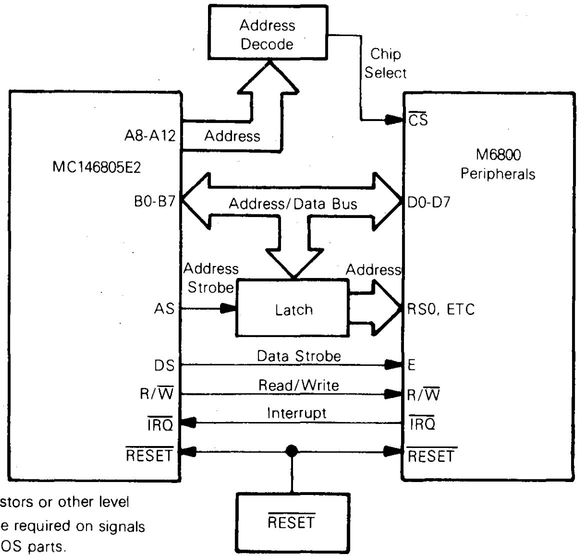


FIGURE 21 — CONNECTION TO CMOS MULTIPLEXED MEMORIES



3

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

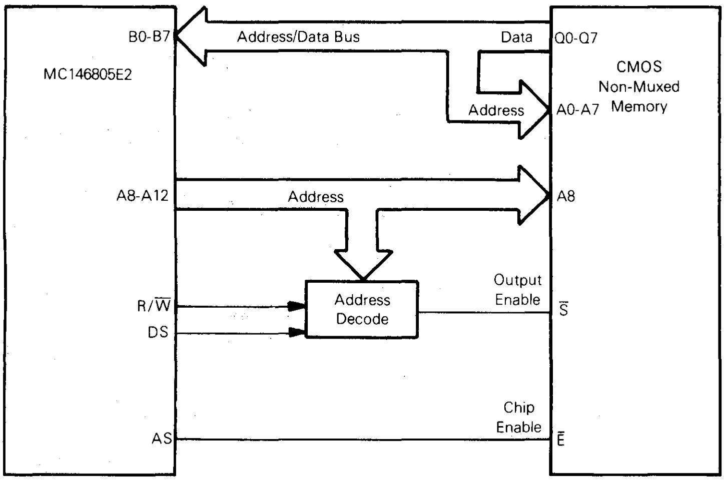


FIGURE 24 — CONNECTION TO STATIC CMOS RAMS

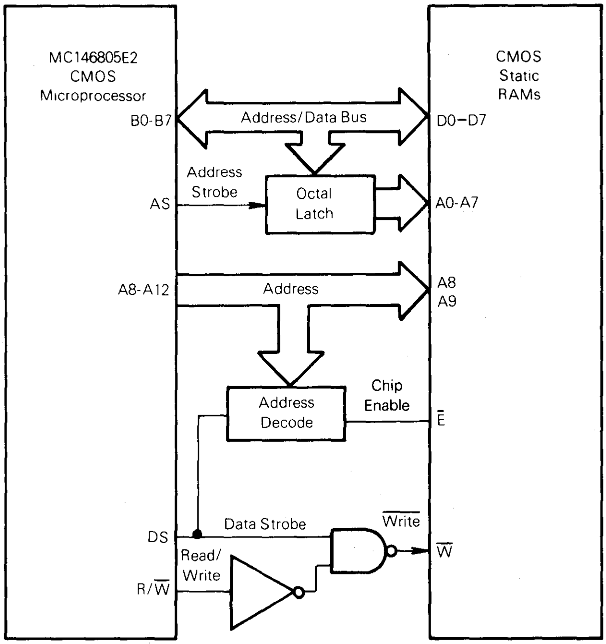
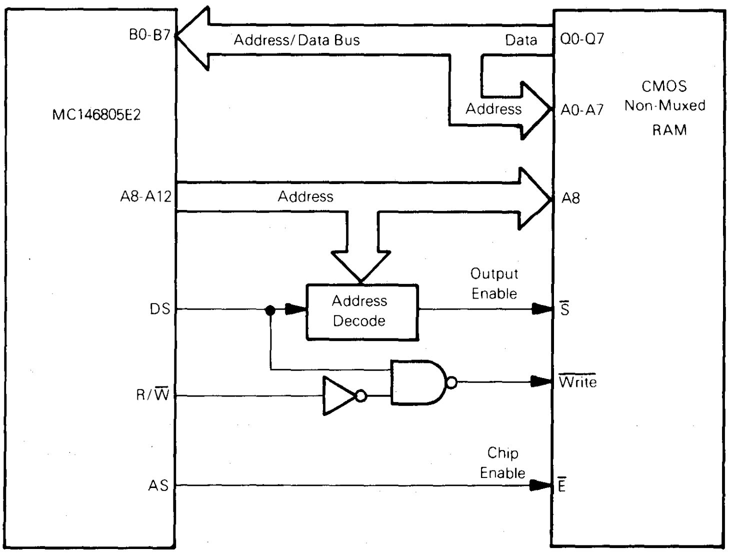


FIGURE 25 — CONNECTION TO LATCHED NON-MULTIPLEXED CMOS RAM



3

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 2	Op Code Address Op Code Address + 1	1 1	1 0	Op Code Jump Address
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	1 0 0	Op Code Address of Operand Operand Data
TST	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Op Code Address + 2	1 1 1 1	1 0 0 0	Op Code Address of Operand Operand Data Op Code Next Instruction
STA STX	4	1 2 3 4	Op Code Address Op Code Adrress + 1 Op Code Address + 1 Address of Operand	1 1 1 0	1 0 0 0	Op Code Address of Operand Address of Operand Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address Operand Address	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand Current Operand Data Current Operand Data New Operand Data
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Starting Address Stack Pointer Stack Pointer − 1	1 1 1 0 0	1 0 0 0 0	Op Code Subroutine Address (LO Byte) 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)
Extended						
JMP	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	1 0 0	Op Code Jump Address (HI Byte) Jump Address (LO Byte)
ADC BIT ORA ADD CMP LDX AND EOR SBC CPX LDA SUB	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address of Operand	1 1 1 1	1 0 0 0	Op Code Address Operand (HI Byte) Address Operand (LO Byte) Operand Data
STA STX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Op Code Address + 2 Op Code Address + 2 Address of Operand	1 1 1 1 0	1 0 0 0 0	Op Code Address of Operand (HI Byte) Address of Operand (LO Byte) Address of Operand (LO Byte) Operand Data
JSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Op Code Address + 2 Subroutine Starting Address Stack Pointer Stack Pointer − 1	1 1 1 1 0 0	1 0 0 0 0 0	Op Code Address of Subroutine (HI Byte) Address of Subroutine (LO Byte) 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)
Indexed, No-Offset						
JMP	2	1 2	Op Code Address Op Code Address + 1	1 1	1 0	Op Code Op Code Next Instruction
ADC EOR CPX ADD LDA LDX AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Index Register	1 1 1	1 0 0	Op Code Op Code Next Instruction Operand Data
TST	4	1 2 3 4	Op Code Address Op Code Address + 1 Index Register Op Code Address + 1	1 1 1 1	1 0 0 0	Op Code Op Code Next Instruction Operand Data Op Code Next Instruction
STA STX	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 1 Index Register	1 1 1 0	1 0 0 0	Op Code Op Code Next Instruction Op Code Next Instruction Operand Data
LSL LSR DEC ASR NEG INC CLR ROL COM ROR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Index Register Index Register Index Register	1 1 1 1 0	1 0 0 0 0	Op Code Op Code Next Instruction Current Operand Data Current Operand Data New Operand Data
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Index Register Stack Pointer Stack Pointer − 1	1 1 1 0 0	1 0 0 0 0	Op Code Op Code Next Instruction 1st Subroutine Op Code Return Address (LO Byte) Return Address (HI Byte)

3

TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION (CONTINUED)

Address Mode / Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Pin	LI Pin	Data Bus
Indexed 8-Bit Offset						
JMP	3	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
ADC EOR CPX ADD LDA LDX AND ORA CMP SUB BIT SBC	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
STA STX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Op Code Address + 1	1	0	Offset
		5	Index Register + Offset	0	0	Operand Data
TST	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Operand Data
		5	Op Code Address + 2	1	0	Op Code Next Instruction
LSL LSR ASR NEG CLR ROL COM ROR DEC INC	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	Current Operand Data
		5	Index Register + Offset	1	0	Current Operand Data
		6	Index Register + Offset	0	0	New Operand Data
JSR	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset
		3	Op Code Address + 1	1	0	Offset
		4	Index Register + Offset	1	0	1st Subroutine Op Code
		5	Stack Pointer	0	0	Return Address LO Byte
		6	Stack Pointer − 1	0	0	Return Address HI Byte
Indexed, 16-Bit Offset						
JMP	4	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
ADC CMP SUB ADD EOR SBC AND ORA CPX LDA BIT LDX	5	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	Operand Data
STA STX	6	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Op Code Address + 2	1	0	Offset (LO Byte)
		6	Index Register + Offset	0	0	Operand Data
JSR	7	1	Op Code Address	1	1	Op Code
		2	Op Code Address + 1	1	0	Offset (HI Byte)
		3	Op Code Address + 2	1	0	Offset (LO Byte)
		4	Op Code Address + 2	1	0	Offset (LO Byte)
		5	Index Register + Offset	1	0	1st Subroutine Op Code
		6	Stack Pointer	0	0	Return Address (LO Byte)
		7	Stack Pointer − 1	0	0	Return Address (HO Byte)

TABLE 11 — SUMMARY OF CYCLE-BY-CYCLE OPERATION (CONTINUED)

Instructions	Cycles	Cycle #	Address Bus	$\overline{\text{RESET}}$ Pin	R/$\overline{\text{W}}$ Pin	LI Pin	Data Bus
Other Functions							
Hardware $\overline{\text{RESET}}$	5		$1FFE	0	1	0	Irrelevant Data
			$1FFE	0	1	0	Irrelevant Data
		1	$1FFE	1	1	0	Irrelevant Data
		2	$1FFE	1	1	0	Irrelevant Data
		3	$1FFE	1	1	0	Vector High
		4	$1FFF	1	1	0	Vector Low
		5	Reset Vector	1	1	0	Op Code
Power on Reset	1922	1	$1FFE	1	1	0	Irrelevant Data
		⋮	⋮	⋮	⋮	⋮	⋮
		1919	$1FFE	1	1	0	Irrelevant Data
		1920	$1FFE	1	1	0	Vector High
		1921	$1FFF	1	1	0	Vector Low
		1922	Reset Vector	1	1	0	Op Code
Instruction	**Cycles**	**Cycles #**	**Address Bus**	**$\overline{\text{IRQ}}$ Pin**	**R/$\overline{\text{W}}$ Pin**	**LI Pin**	**Data Bus**
$\overline{\text{IRQ}}$ Interrupt (Timer Vector $1FF8, $1FF9)	10		Last Cycle of Previous Instruction	0	X	0	X
		1	Next Op Code Address	0	1	0	Irrelevant Data
		2	Next Op Code Address	X	1	0	Irrelevant Data
		3	SP	X	0	0	Return Address (LO Byte)
		4	SP − 1	X	0	0	Return Address (HI Byte)
		5	SP − 2	X	0	0	Contents Index Reg
		6	SP − 3	X	0	0	Contents Accumulator
		7	SP − 4	X	0	0	Contents CC Register
		8	$1FFA	X	1	0	Vector High
		9	$1FFB	X	1	0	Vector Low
		10	$\overline{\text{IRQ}}$ Vector	X	1	0	Int Routine First

APPENDIX

MC146805E2 INTERRUPT CLARIFICATION

Under certain circumstances, the MC146805E2 (BP4XXXX and AW9XXXX) 8-bit Microprocessor Unit $\overline{\text{IRQ}}$ interrupt does not conform to the operation described in this Advanced Information Sheet.

1. The level sensitive $\overline{\text{IRQ}}$ mode, which is by far the most frequently used, is **FULLY OPERATIONAL:** thus, most MC146805E2 applications are unaffected. However, the edge-triggered $\overline{\text{IRQ}}$ interrupt mode **MIGHT NOT BE SERVICED** under certain programming circumstances; therefore, it is recommended that the edge-triggered mode not be used.

2. An interrupt-vector address **CAN BE** improperly generated in some circumstances. There is a possibility that when an external interrupt ($\overline{\text{IRQ}}$) and timer interrupt occur during the WAIT mode (following wait instruction), address locations $1FF2 and $1FF3 are selected instead of vector locations $1FF6 and $1FF7. There are three specific examples listed below; two of these require no action and the third has a recommended solution.

   a. Those not using the WAIT mode need not take any action.

   b. If the WAIT mode is used without external interrupt ($\overline{\text{IRQ}}$ pin held high), no precautions are required.

   c. When $\overline{\text{IRQ}}$ can be active (low) during the WAIT mode, the vector in locations $1FF6 and $1FF7 (the WAIT mode timer interrupt vector) should be duplicated in $1FF2 and $1FF3. In this way the circumstances that caused selection of the second vector do not disturb normal program execution.

On future MC146805E2 parts, no special actions will be necessary. If you have questions, contact your Motorola distributor or Motorola sales office, or contact Motorola Microprocessor Applications Engineering in Austin, Texas.

MOTOROLA

MC146805E3

Product Preview

8-BIT MICROPROCESSOR UNIT

The MC146805E3 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. Operation is identical to the MC146805E2 except that this device includes a 64K memory addressing capability.

The MC146805E3 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.

HARWARE 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 Programming 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 64K Bytes of External Memory
- Single 3- to 6-Volt Supply
- On-Chip Oscillator

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

GENERAL DESCRIPTION

The MC146805E3 MPU, an expanded version of the MC146805E2 MPU, includes a 64K memory addressing capability. The following paragraph explains the modifications made to the MC146805E2 and reference should be made to the *MC146805E2 Advance Information Data Sheet* (ADI-850-R2) for detailed information.

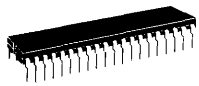
> Port A bits 5, 6, and 7 have been replaced by high-byte address bits 13, 14, and 15. The new address pins will behave identically to the current high address pins (A8-A12). Port A bits 5 through 7 will be seen as "read only" bits and will be read as zeros facilitating "all zero" or "any one" testing. Port A data direction bits 5 through 7 will be seen as "read only" bits and will be read as ones, indicating that they are outputs.

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

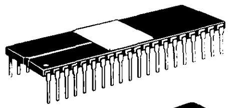
CMOS

(HIGH PERFORMANCE SILICON GATE)

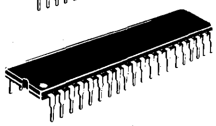
8-BIT MICROPROCESSOR



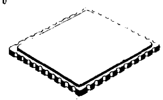
S SUFFIX
CERDIP PACKAGE
CASE 734



L SUFFIX
CERAMIC PACKAGE
CASE 715



P SUFFIX
PLASTIC PACKAGE
CASE 711



Z SUFFIX
CHIP CARRIER
CASE 761

PIN ASSIGNMENT

$\overline{RESET}$	1	40	V_{DD}
$\overline{IRQ}$	2	39	OSC1
LI	3	38	OSC2
DS	4	37	TIMER
R/$\overline{W}$	5	36	PB0
AS	6	35	PB1
A15	7	34	PB2
A14	8	33	PB3
A13	9	32	PB4
PA4	10	31	PB5
PA3	11	30	PB6
PA2	12	29	PB7
PA1	13	28	B0
PA0	14	27	B1
A12	15	26	B2
A11	16	25	B3
A10	17	24	B4
A9	18	23	B5
A8	19	22	B6
V_{SS}	20	21	B7

3

MOTOROLA

MC146818 Addendum

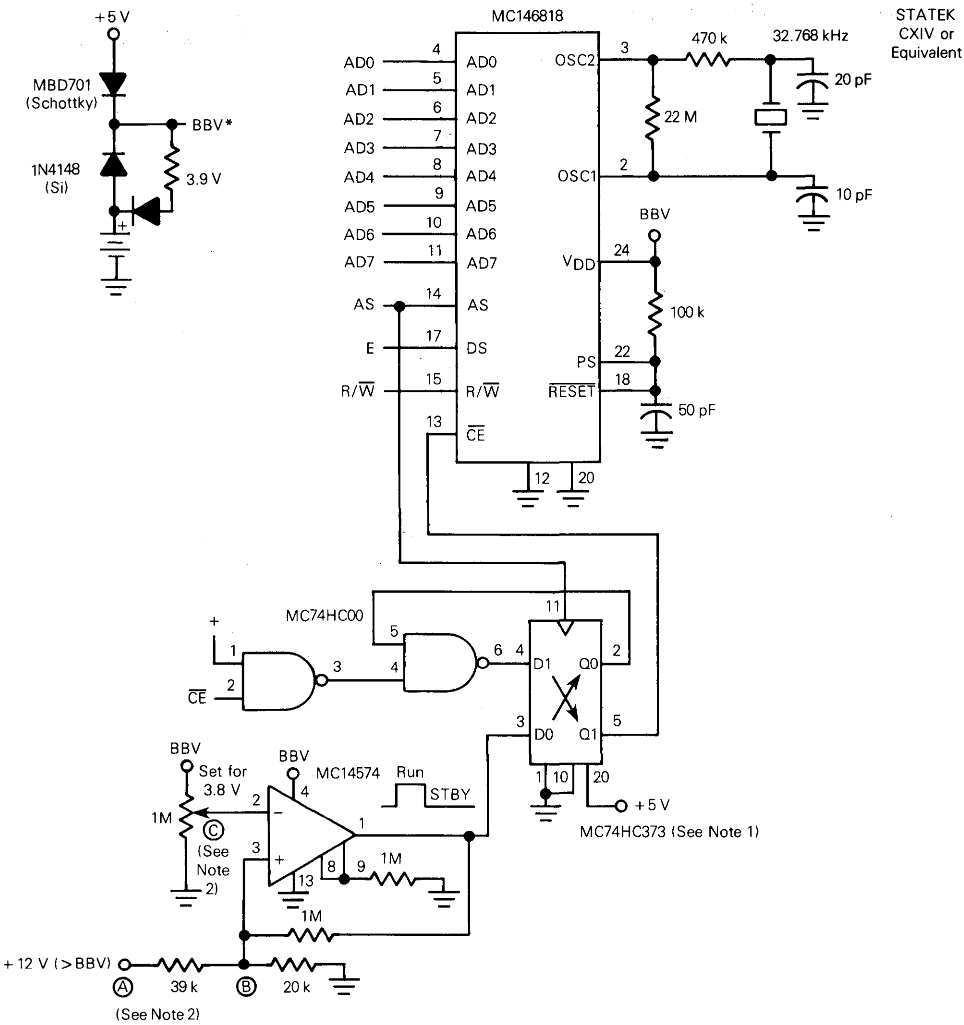
Advance Information

REAL-TIME CLOCK PLUS RAM (RTC)
Advance Information Data Sheet
ADI-856-R1

The following information is an addition to **POWER-DOWN CONSIDERATIONS** found on page 11 of the MC146818 Advance Information Data Sheet (ADI-856-R1).

MC146818s with the date code of 3N46XXXX and GC6XXXX require a synchronization of the $\overline{CE}$ pin with address strobe. The following circuit will satisfy that condition, and also show a typical application of power-down circuitry.

If $\overline{CE}$ is grounded at all times (no power down required) the following circuit need not be used.



*BBV = Battery Backup Voltage

NOTES:

1. All unused inputs of the MC74HC373 must be grounded.
2. If point A equals 12 V point B should be equal to 4.06 V. If point A equals 10 V point B should be equal to 3.38 V with C set for 3.18 V.

MOTOROLA

MC146818

Advance Information

REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 μW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5 μs to 500 ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input At Time Base Frequency ÷1 or ÷4
- 24-Pin Dual-In-Line Package
- Chip Carrier Also Available

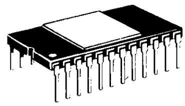
This document contains information on a new product. Specifications and information herein are subject to change without notice.

CMOS

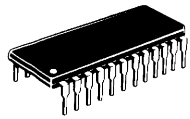
(HIGH-PERFORMANCE SILICON-GATE COMPLEMENTARY MOS)

REAL-TIME CLOCK PLUS RAM

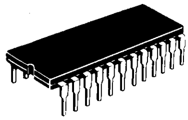
L SUFFIX
CERAMIC PACKAGE
CASE 716



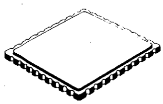
P SUFFIX
PLASTIC PACKAGE
CASE 709



S SUFFIX
CERDIP PACKAGE
CASE 623



Z SUFFIX
CHIP CARRIER
CASE 761



PIN ASSIGNMENT

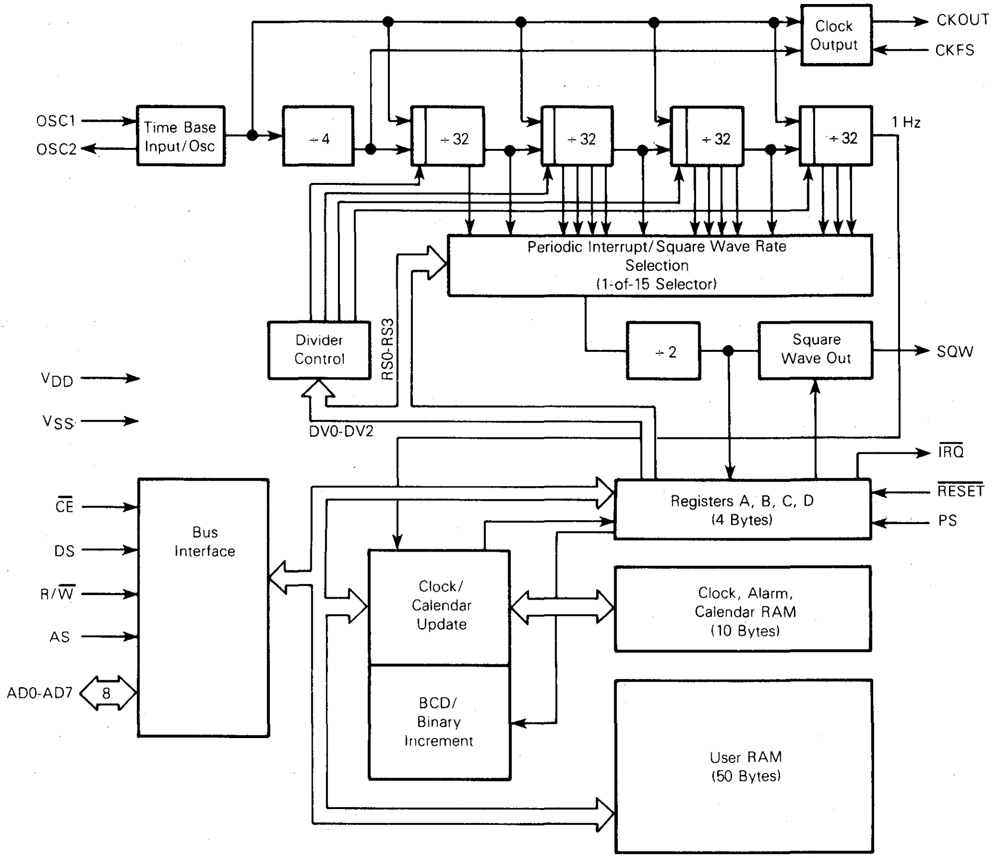
Signal	Pin	Pin	Signal
NC	1	(39) 24	V_{DD}
OSC1	2 (3)	(38) 23	SQW
OSC2	3 (4)	(37) 22	PS
AD0	4 (8)	(34) 21	CKOUT
AD1	5 (9)	(33) 20	CKFS
AD2	6 (10)	(32) 19	$\overline{IRQ}$
AD3	7 (11)	(31) 18	$\overline{RESET}$
AD4	8 (12)	(30) 17	DS
AD5	9 (13)	16	NC
AD6	10 (18)	(24) 15	R/$\overline{W}$
AD7	11 (19)	(23) 14	AS
V_{SS}	12 (20)	(22) 13	$\overline{CE}$

Pin numbers in parentheses represent equivalent Z suffix chip carrier pins. Pins that have not been designated for the chip carrier are not connected.

3

3

FIGURE 1 — BLOCK DIAGRAM



MAXIMUM RATINGS (Voltages referenced to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages Except OSC1	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range MC146818 MC146818C (V_{DD} = 3.0 to 5.5 V operation)	T_A	T_L to T_H 0 to 70 −40 to 85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic Cerdip Ceramic	θ_{JA}	 120 65 50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ 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} = 3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}	32.768	32.768	kHz
Output Voltage $I_{Load} < 10\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
I_{DD} – Bus Idle CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{CE} = V_{DD}-0.2$; C_L (OSC2) = 10 pF f_{osc} = 32.768 kHz	I_{DD3}	–	50	μA
I_{DD} – Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = $V_{DD}-0.2$ V; No Clock	I_{DD4}	–	50	μA
Output High Voltage (L_{Load} = −0.25 mA, All Outputs)	V_{OH}	2.7	–	V
Output Low Voltage (I_{Load} = 0.25 mA, All Outputs)	V_{OL}	–	0.3	V
Input High Voltage — AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, $\overline{RESET}$, CKFS, PS, OSC1	V_{IH}	2.1 2.5	V_{DD} V_{DD}	V
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.5	V
Input Current — All Inputs	I_{in}	–	±1	μA
Three-State Leakage — $\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	μA

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristics	Symbol	Min	Max	Unit
Frequency of Operation	f_{osc}	32.768	4194.304	kHz
Output Voltage $I_{Load} < 10\ \mu A$	V_{OL} V_{OH}	– $V_{DD}-0.1$	0.1 –	V
I_{DD} – Bus Idle (External Clock) CKOUT = f_{osc}, C_L = 15 pF; SQW Disabled, $\overline{CE} = V_{DD}-0.2$; C_L (OSC2) = 10 pF f_{osc} = 4.194304 MHz f_{osc} = 1.048516 MHz f_{osc} = 32.768 kHz	 I_{DD1} I_{DD2} I_{DD3}	 – – –	 3 800 50	 mA μA μA
I_{DD} – Quiescent f_{osc} = DC; OSC1 = DC; All Other Inputs = $V_{DD}-0.2$ V; No Clock	I_{DD4}	–	50	μA
Output High Voltage (I_{Load} = −1.6 mA, AD0-AD7, CKOUT) (I_{Load} = −1.0 mA, SQW)	V_{OH}	4.1	–	V
Output Low Voltage (I_{Load} = 1.6 mA, AD0-AD7, CKOUT) (I_{Load} = 1.0 mA, $\overline{IRQ}$ and SQW)	V_{OL}	–	0.4	V
Input High Voltage — CKFS, AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$, PS $\overline{RESET}$ OSC1	V_{IH}	$V_{DD}-2.0$ $V_{DD}-0.8$ $V_{DD}-1.0$	V_{DD} V_{DD} V_{DD}	V
Input Low Voltage — AD0-AD7, DS, AS, R/$\overline{W}$, $\overline{CE}$ CKFS, PS, $\overline{RESET}$ OSC1	V_{IL}	V_{SS} V_{SS} V_{SS}	0.8 0.8 0.8	V
Input Current — All Inputs	I_{in}	–	±1	μA
Three-State Leakage — $\overline{IRQ}$, AD0-AD7	I_{TSL}	–	±10	μA

3

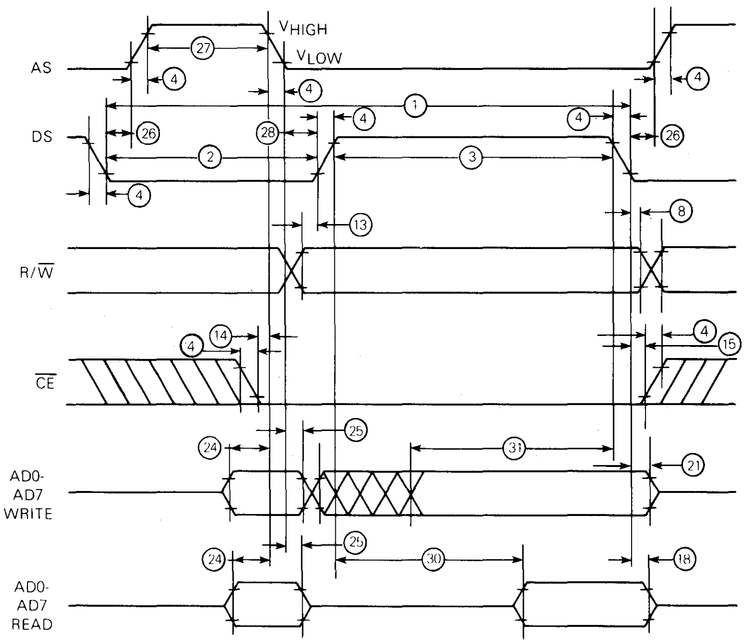
3

BUS TIMING

Ident. Number	Characteristics	Symbol	V_{DD} = 3.0 V 50 pF Load Min	Max	V_{DD} = 5.0 V ± 10% 2 TTL and 130 pF Load Min	Max	Unit
1	Cycle Time	t_{cyc}	5000	–	953	dc	ns
2	Pulse Width, DS/E Low or $\overline{RD}/\overline{WR}$ High	PW_{EL}	1000	–	300	–	ns
3	Pulse Width, DS/E High or $\overline{RD}/\overline{WR}$ Low	PW_{EH}	1500	–	325	–	ns
4	Input Rise and Fall Time	t_r, t_f	–	100	–	30	ns
8	R/$\overline{W}$ Hold Time	t_{RWH}	10	–	10	–	ns
13	R/$\overline{W}$ Setup Time Before DS/E	t_{RWS}	200	–	80	–	ns
14	Chip Enable Setup Time Before AS/ALE Fall	t_{CS}	200	*	55	*	ns
15	Chip Enable Hold Time	t_{CH}	10	–	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	1000	10	100	ns
21	Write Data Hold Time	t_{DHW}	100	–	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	200	–	50	–	ns
25	Muxed Address Hold Time	t_{AHL}	100	–	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	500	–	50	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	600	–	135	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	500	–	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	1300	–	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	1500	–	200	–	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.
*Refer to **IMPORTANT NOTICES** appearing on page 20 of this data sheet.

FIGURE 2 — MC146818 BUS TIMING



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

FIGURE 3 — BUS READ TIMING COMPETITOR MULTIPLEXED BUS

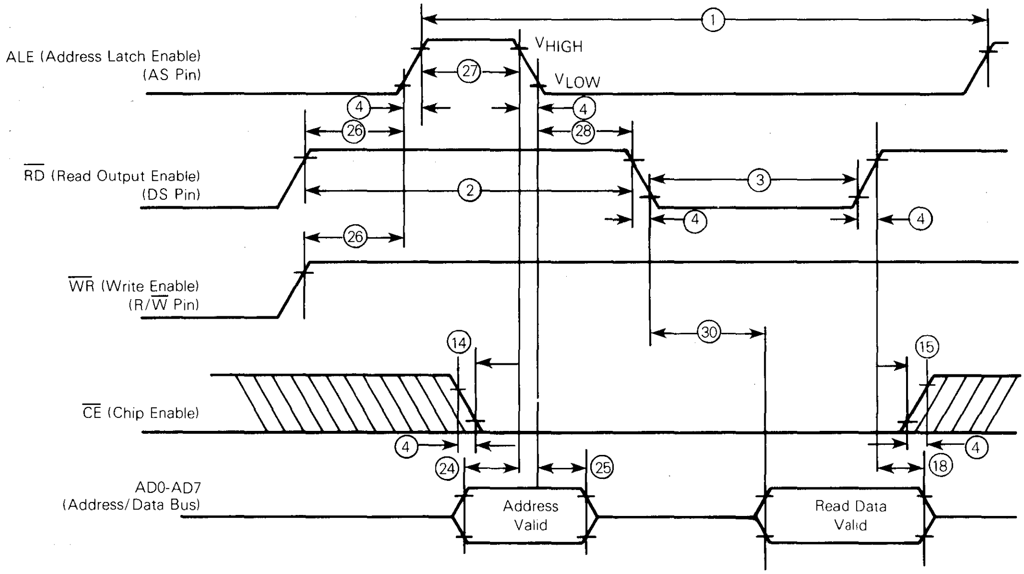
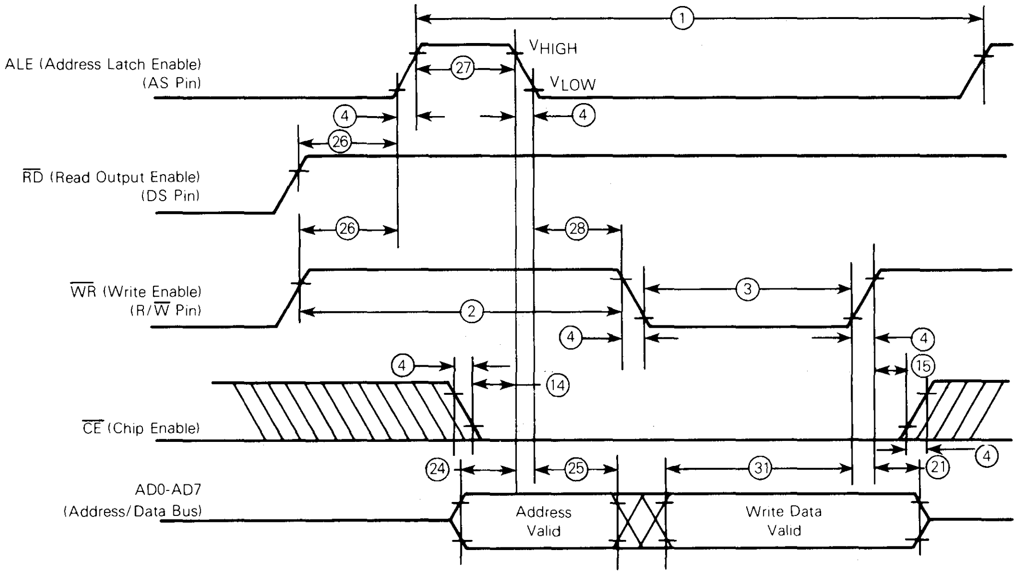


FIGURE 4 — BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V $\pm$ 10%

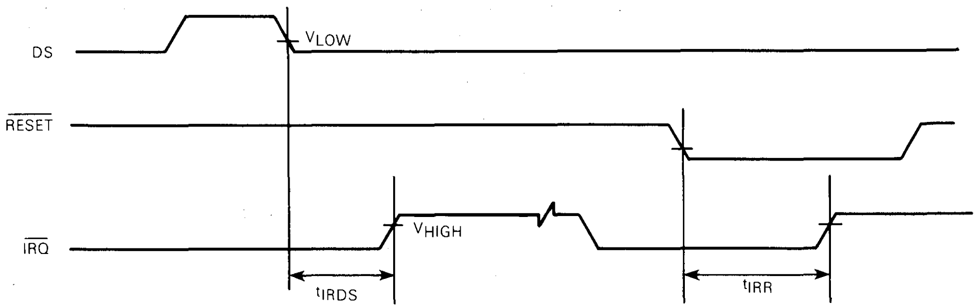
3

TABLE 1 — SWITCHING CHARACTERISTICS (V_{DD}=5.0 Vdc ±10%, V_{SS}=0 Vdc, $T_A=T_L$ to T_H)

Description	Symbol	Min	Max	Unit
Oscillator Startup	t_{RC}	—	100	ms
Reset Pulse Width	t_{RWL}	5	—	µs
Reset Delay Time	t_{RLH}	5	—	µs
Power Sense Pulse Width	t_{PWL}	5	—	µs
Power Sense Delay Time	t_{PLH}	5	—	µs
$\overline{IRQ}$ Release from DS	t_{IRDS}	—	2	µs
$\overline{IRQ}$ Release from $\overline{RESET}$	t_{IRR}	—	2	µs
VRT Bit Delay	t_{VRTD}	—	2	µs

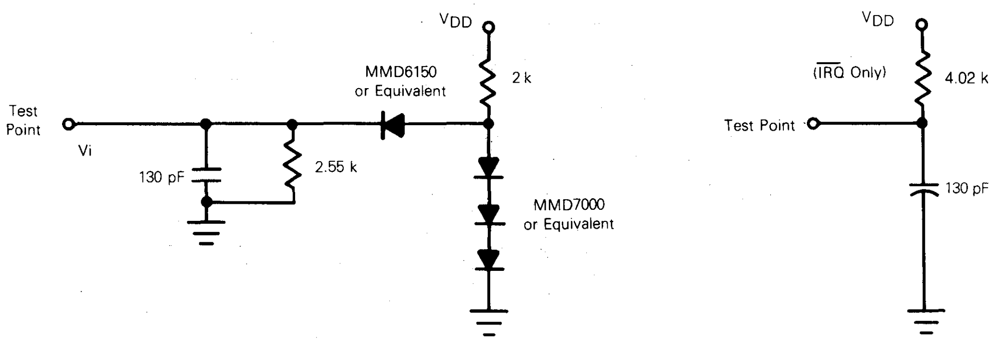
3

FIGURE 5 — $\overline{IRQ}$ RELEASE DELAY



NOTE: $V_{HIGH}=V_{DD}-2.0$ V, V_{LOW}=0.8 V, for V_{DD}=5.0 V ±10%

FIGURE 6 — TTL EQUIVALENT TEST LOAD



All Outputs Except OSC2 (See Figure 10)

FIGURE 7 — POWER-UP

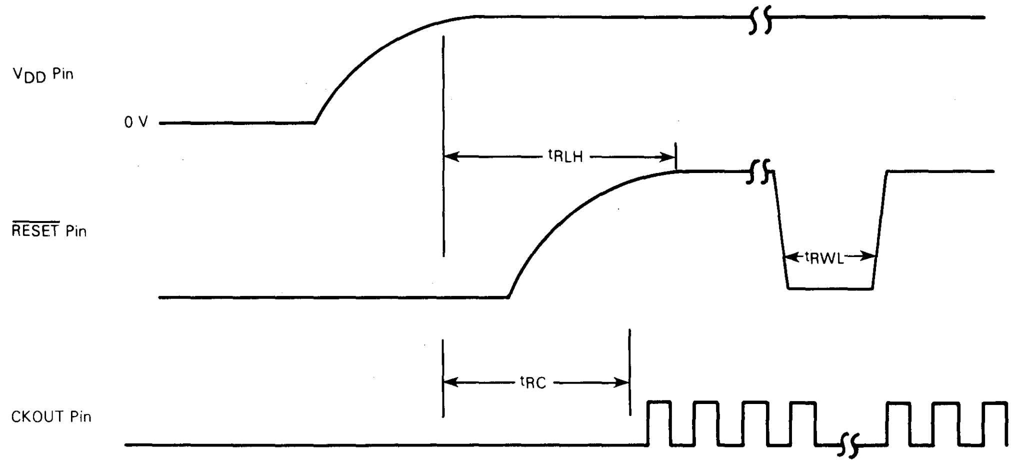
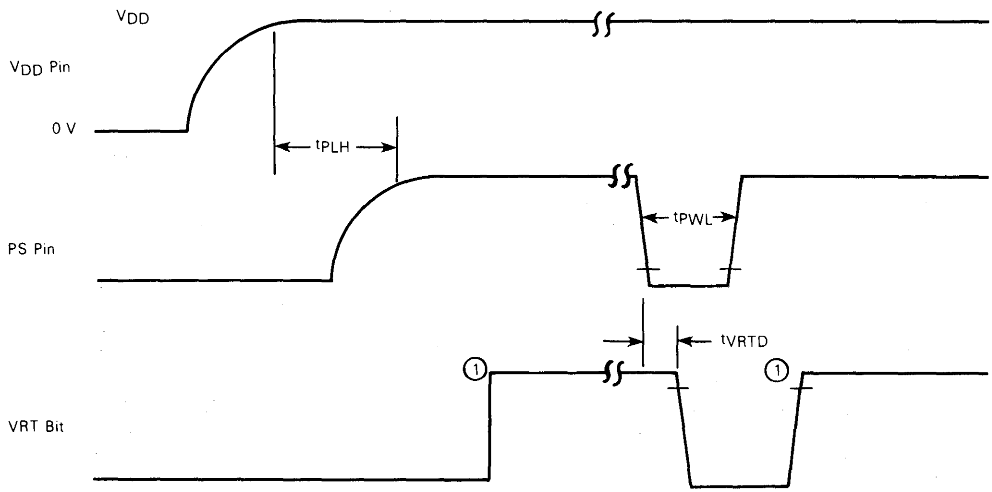


FIGURE 8 — CONDITIONS THAT CLEAR VRT BIT



1 The VRT bit is set to a "1" by reading Register d. The VRT bit can only be cleared by pulling the PS pin low (see **REGISTER D ($0D)**).

MOTEL

The MOTEL circuit is a new concept that permits the MC146818 to be directly interfaced with many types of microprocessors. No external logic is needed to adapt to the differences in bus control signals from common multiplexed bus microprocessors.

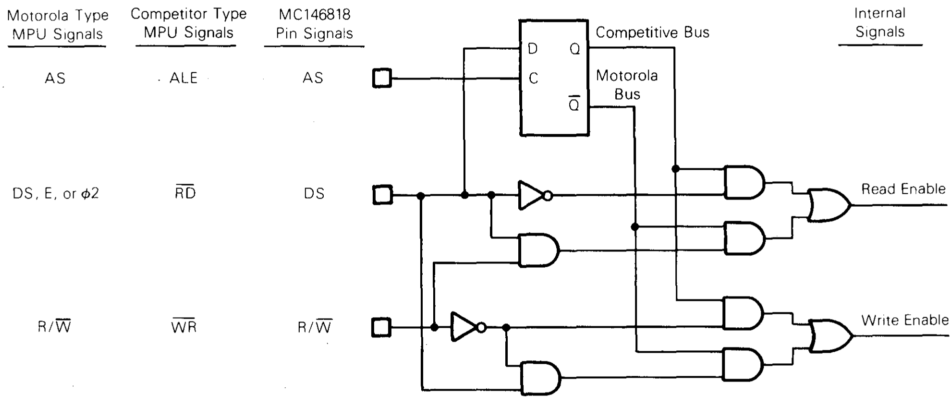
Practically all microprocessors interface with one of two synchronous bus structures. One bus was originated by the Motorola MC6800 and the other by the Intel 8080 and its companion part, the 8228.

The MOTEL circuit (for MOTorola and IntEL bus compatibility) is built into peripheral and memory ICs to permit direct connection to either type of bus. An industry standard bus structure is now available. The MOTEL concept is shown logically in Figure 9.

MOTEL selects one of two interpretations of two pins. In the Motorola case, DS and R/$\overline{W}$ are gated together to produce the internal read enable. The internal write enable is a similar gating of the inverse of R/$\overline{W}$. With competitor buses, the inversion of $\overline{RD}$ and $\overline{WR}$ create functionally identical internal read and write enable signals.

The MC146818 automatically selects the processor type by using AS/ALE to latch the state of the DS/$\overline{RD}$ pin. Since DS is always low and $\overline{RD}$ is always high during AS and ALE, the latch automatically indicates which processor type is connected.

FIGURE 9 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



SIGNAL DESCRIPTIONS

The block diagram in Figure 1, shows the pin connection with the major internal functions of the MC146818 Real-Time Clock plus RAM. The following paragraphs describe the function of each pin.

V_{DD}, V_{SS}

DC power is provided to the part on these two pins, V_{DD} being the more positive voltage. The minimum and maximum voltages are listed in the Electrical Characteristics tables.

OSC1, OSC2 — TIME BASE, INPUTS

The time base for the time functions may be an external signal or the crystal oscillator. External square waves at 4.194304 MHz, 1.048576 MHz, or 32.768 kHz may be connected to OSC1 as shown in Figure 10. The internal time-base frequency to be used is chosen in Register A.

The on-chip oscillator is designed for a parallel resonant AT cut crystal at 4.194304 MHz or 1.048576 MHz frequencies. The crystal connections are shown in Figure 11 and the crystal characteristics in Figure 12.

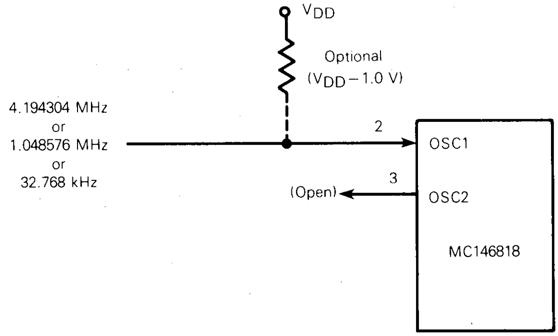
CKOUT — CLOCK OUT, OUTPUT

The CKOUT pin is an output at the time-base frequency divided by 1 or 4. A major use for CKOUT is as the input clock to the microprocessor; thereby saving the cost of a second crystal. The frequency of CKOUT depends upon the time-base frequency and the state of the CKFS pin as shown in Table 2.

CKFS — CLOCK OUT FREQUENCY SELECT, INPUT

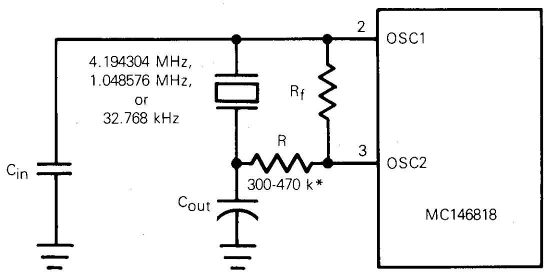
When the CKFS pin is tied to V_{DD} it causes CKOUT to be the same frequency as the time base at the OSC1 pin. When CKFS is tied to V_{SS}, CKOUT is the OSC1 time-base frequency divided by four. Table 2 summarizes the effect of CKFS.

FIGURE 10 — EXTERNAL TIME-BASE CONNECTION



3

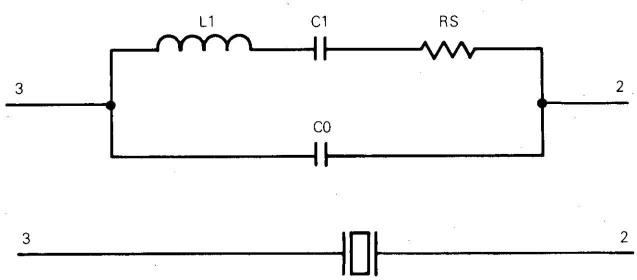
FIGURE 11 — CRYSTAL OSCILLATOR CONNECTION



*32.768 kHz Only — Consult Crystal Manufacturer's Specification

FIGURE 12 — CRYSTAL PARAMETERS

Crystal Equivalent Circuit



f_{osc}	4.194304 MHz	1.048576 MHz	32.768 kHz
RS (Maximum)	75 Ω	700 Ω	50 k
C0 (Maximum)	7 pF	5 pF	1.7 pF
C1	0.012 pF	0.008 pF	0.003 pF
Q	50 k	35 k	30 k
C_{in}/C_{out}	15-30 pF	15-40 pF	10-22 pF
R	–	–	300-470 k
R_f	10 M	10 M	22 M

TABLE 2 — CLOCK OUTPUT FREQUENCIES

Time Base (OSC1) Frequency	Clock Frequency Select Pin (CKFS)	Clock Frequency Output Pin (CKOUT)
4.194304 MHz	High	4.194304 MHz
4.194304 MHz	Low	1.048576 MHz
1.048576 MHz	High	1.048576 MHz
1.048576 MHz	Low	262.144 kHz
32.768 kHz	High	32.768 kHz
32.768 kHz	Low	8.192 kHz

SQW — SQUARE WAVE, OUTPUT

The SQW pin can output a signal from one of the 15 taps provided by the 22 internal-divider stages. The frequency of the SQW may be altered by programming Register A, as shown in Table 5. The SQW signal may be turned on and off using the SQWE bit in Register B.

AD0-AD7 — MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion for data. Address-then-data multiplexing does not slow the access time of the MC146818 since the bus reversal from address to data is occurring during the internal RAM access time.

The address must be valid just prior to the fall of AS/ALE at which time the MC146818 latches the address from AD0 to AD5. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{\text{WR}}$ pulses. In a read cycle, the MC146818 outputs eight bits of data during the latter portion of the DS or $\overline{\text{RD}}$ 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 $\overline{\text{RD}}$ 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 $\overline{\text{RD}}$, $\overline{\text{MEMR}}$, or $\overline{\text{I/OR}}$ 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/$\overline{\text{W}}$ — READ/WRITE, INPUT

The MOTEL circuit treats the R/$\overline{\text{W}}$ pin in one of two ways. When a Motorola type processor 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/$\overline{\text{W}}$ during DS

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 type processors. The MOTEL circuit in this mode gives R/$\overline{\text{W}}$ pin the same meaning as the write (W) pulse on many generic RAMs.

$\overline{\text{CE}}$ — CHIP ENABLE, INPUT

The chip-enable ($\overline{\text{CE}}$) signal must be asserted (low) for a bus cycle in which the MC146818 is to be accessed. $\overline{\text{CE}}$ is not latched and must be stable during DS and AS (Motorola case of MOTEL) 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 MC146818. When $\overline{\text{CE}}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{\text{CE}}$ is high, all address, data, DS, and R/$\overline{\text{W}}$ inputs from the processor are disconnected within the MC146818. This permits the MC146818 to be isolated from a powered-down processor. When $\overline{\text{CE}}$ 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 $\overline{\text{CE}}$ when the main power is off. When $\overline{\text{CE}}$ is not used, it should be grounded.

$\overline{\text{IRQ}}$ — INTERRUPT REQUEST, OUTPUT

The $\overline{\text{IRQ}}$ pin is an active low output of the MC146818 that may be used as an interrupt input to a processor. The $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ pin, the processor program normally reads Register C. The $\overline{\text{RESET}}$ pin also clears pending interrupts.

When no interrupt conditions are present, the $\overline{\text{IRQ}}$ level is in the high-impedance state. Multiple interrupting devices may thus be connected to an $\overline{\text{IRQ}}$ bus with one pullup at the processor.

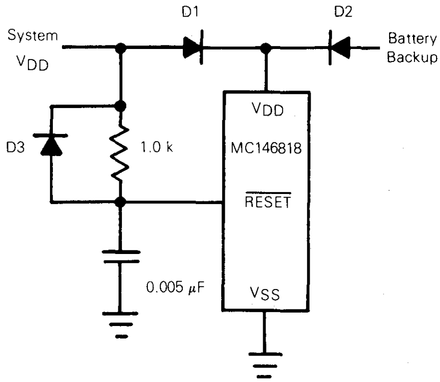
$\overline{\text{RESET}}$ — RESET, INPUT

The $\overline{\text{RESET}}$ pin does not affect the clock, calendar, or RAM functions. On powerup, the $\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ pin circuit.

When $\overline{\text{RESET}}$ is low the following occurs:

a) Periodic Interrupt Enable (PIE) bit is cleared to zero,
b) Alarm Interrupt Enable (AIE) bit is cleared to zero,
c) Update ended Interrupt Enable (UIE) bit is cleared to zero,
d) Update ended Interrupt Flag (UF) bit is cleared to zero,
e) Interrupt Request status Flag (IRQF) bit is cleared to zero,
f) Periodic Interrupt Flag (PF) bit is cleared to zero,
g) The part is not accessible.

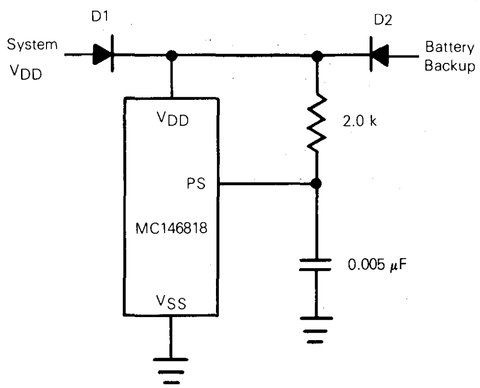
FIGURE 13 — TYPICAL POWERUP DELAY CIRCUIT FOR $\overline{RESET}$



D1 = MBD701 (Schottky) or Equivalent
D2 = D3 = 1N4148 or Equivalent

Note: If the RTC is isolated from the MPU or MCU power by a diode drop, care must be taken to meet V_{in} requirements.

FIGURE 14 — TYPICAL POWERUP DELAY CIRCUIT FOR POWER SENSE



D1 = MBD701 (Schottky) or Equivalent
D2 = 1N4148 or Equivalent

g) Alarm Interrupt Flag (AF) bit is cleared to zero,
h) $\overline{IRQ}$ pin is in high-impedance state, and
i) Square Wave output Enable (SQWE) bit is cleared to zero.

PS — POWER SENSE, INPUT

The power-sense pin is used in the control of the valid RAM and time (VRT) bit in Register D. When the PS pin is low the VRT bit is cleared to zero.

When using the VRT feature during powerup, the PS pin must be externally held low for the specified t_{PLH} time. As power is applied, the VRT bit remains low indicating that the contents of the RAM, time registers, and calendar are not guaranteed. PS must go high after powerup to allow the VRT bit to be set by a read of register D.

POWER-DOWN CONSIDERATIONS

In most systems, the MC146818 must continue to keep time when system power is removed. In such systems, a conversion from system power to an alternate power supply, usually a battery, must be made. During the transition from system to battery power, the designer of a battery backed-up RTC system must protect data integrity, minimize power consumption, and ensure hardware reliability.

The chip enable ($\overline{CE}$) pin controls all bus inputs (R/$\overline{W}$, DS, AS, AD0-AD7). $\overline{CE}$, when negated, disallows any unintended modification of the RTC data by the bus. $\overline{CE}$ also reduces power consumption by reducing the number of transitions seen internally.

Power consumption may be further reduced by removing resistive and capacitive loads from the clock out (CKOUT) pin and the squarewave (SQW) pin.

During and after the power source conversion, the V_{IN} maximum specification must never be exceeded. Failure to meet the V_{IN} maximum specification can cause a virtual SCR to appear which may result in excessive current drain and destruction of the part.

ADDRESS MAP

Figure 15 shows the address map of the MC146818. The memory consists of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes are directly readable and writable by the processor program except for the following: 1) Registers C and D are read only, 2) bit 7 of Register A is read only, and 3) the high-order bit of the seconds byte is read only. The contents of four control and status registers (A, B, C, and D) are described in **REGISTERS.**

TIME, CALENDAR, AND ALARM LOCATIONS

The processor program obtains time and calendar information by reading the appropriate locations. The program may initialize the time, calendar, and alarm by writing to these RAM locations. The contents of the 10 time, calendar, and alarm bytes may be either binary or binary-coded decimal (BCD).

Before initializing the internal registers, the SET bit in Register B should be set to a "1" to prevent time/calendar updates from occurring. The program initializes the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of Register B. All 10 time, calendar, and alarm bytes must use the same data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialized the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitializing the 10 data bytes.

Table 3 shows the binary and BCD formats of the 10 time, calendar, and alarm locations. The 24/12 bit in Register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitializing the hour locations. When the 12-hour format is selected the high-order bit of the hours byte represents PM when it is a "1".

The time, calendar, and alarm bytes are not always accessable by the processor program. Once-per-second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 bytes are read at this time, the data outputs are undefined. The update lockout time is 248 μs at the 4.194304 MHz and 1.048567 MHz time bases and 1948 μs for the 32.768 kHz time base. The Update Cycle section shows how to accommodate the update cycle in the processor program.

FIGURE 15 — ADDRESS MAP

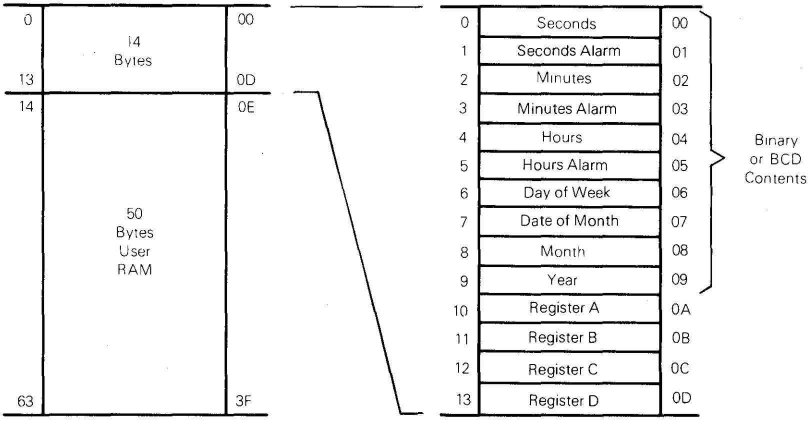


TABLE 3 — TIME, CALENDAR, AND ALARM DATA MODES

Address Location	Function	Decimal Range	Range: Binary Data Mode	Range: BCD Data Mode	Example*: Binary Data Mode	Example*: BCD Data Mode
0	Seconds	0-59	\$00-\$3B	\$00-\$59	15	21
1	Seconds Alarm	0-59	\$00-\$3B	\$00-\$59	15	21
2	Minutes	0-59	\$00-\$3B	\$00-\$59	3A	58
3	Minutes Alarm	0-59	\$00-\$3B	\$00-\$59	3A	58
4	Hours (12 Hour Mode)	1-12	\$01-\$0C (AM) and \$81-\$8C (PM)	\$01-\$12 (AM) and \$81-\$92 (PM)	05	05
	Hours (24 Hour Mode)	0-23	\$00-\$17	\$00-\$23	05	05
5	Hours Alarm (12 Hour Mode)	1-12	\$01-\$0C (AM) and \$81-\$8C (PM)	\$01-\$12 (AM) and \$81-\$92 (PM)	05	05
	Hours Alarm (24 Hour Mode)	0-23	\$00-\$17	\$00-23	05	05
6	Day of the Week Sunday = 1	1-7	\$01-\$07	\$01-\$07	05	05
7	Date of the Month	1-31	\$01-\$1F	\$01-\$31	0F	15
8	Month	1-12	\$01-\$0C	\$01-\$12	02	02
9	Year	0-99	\$00-\$63	\$00-\$99	4F	79

*Example: 5:58:21 Thursday 15 February 1979 (time is AM)

The three alarm bytes may be used in two ways. First, when the program inserts an alarm time in the appropriate hours, minutes, and seconds alarm locations, the alarm interrrupt is initiated at the specified time each day if the alarm enable bit is high. The second usage is to insert a "don't care" state in one or more of three alarm bytes. The "don't care" code is any hexadecimal byte from C0 to FF. That is, the two most-significant bits of each byte, when set to "1", create a "don't care" situation. An alarm interrupt each hour is created with a "don't care" code in the hours alarm location. Similarly, an alarm is generated every minute with "don't care" codes in the hours and minutes alarm bytes. The "don't care" codes in all three alarm bytes create an interrupt every second.

STATIC CMOS RAM

The 50 general purpose RAM bytes are not dedicated within the MC146818. They can be used by the processor program, and are fully available during the update cycle.

When time and calendar information must use battery back-up, very frequently there is other non-volatile data that must be retained when main power is removed. The 50 user RAM bytes serve the need for low-power CMOS battery-backed storage, and extend the RAM available to the program.

When further CMOS RAM is needed, additional MC146818s may be included in the system. The time/calendar functions may be disabled by holding the DV0-DV2 dividers, in Register A, in the reset state by setting the SET bit in Register B or by removing the oscillator. Holding the dividers in reset prevents interrupts or SQW output from operating while setting the SET bit allows these functions to occur. With the dividers clear, the available user RAM is extended to 59 bytes. The high-order bit of the seconds byte, bit 7 of Register A, and all bits of Registers C and D cannot effectively be used as general purpose RAM.

INTERRUPTS

The RTC plus RAM includes three separate fully automatic sources of interrupts to the processor. The alarm interrupt may be programmed to occur at rates from once-per-second to one-a-day. The periodic interrupt may be selected for rates from half-a-second to 30.517 μs. The update-ended interrupt may be used to indicate to the program that an update cycle is completed. Each of these independent interrupt conditions are described in greater detail in other sections.

The processor program selects which interrupts, if any, it wishes to receive. Three bits in Register B enable the three interrupts. Writing a "1" to a interrupt-enable bit permits that interrupt to be initiated when the event occurs. A "0" in the interrupt-enable bit prohibits the IRQ pin from being asserted due to the interrupt cause.

If an interrupt flag is already set when the interrupt becomes enabled, the $\overline{IRQ}$ pin is immediately activated, though the interrupt initiating the event may have occurred much earlier. Thus, there are cases where the program should clear such earlier initiated interrupts before first enabling new interrupts.

When an interrupt event occurs a flag bit is set to a "1" in Register C. Each of the three interrupt sources have separate flag bits in Register C, which are set independent of the state of the corresponding enable bits in Register B. The flag bit may be used with or without enabling the corresponding enable bits.

In the software scanned case, the program does not enable the interrupt. The "interrupt" flag bit becomes a status bit, which the software interrogates, when it wishes. When the software detects that the flag is set, it is an indication to software that the "interrupt" event occurred since the bit was last read.

However, there is one precaution. The flag bits in Register C are cleared (record of the interrupt event is erased) when Register C is read. Double latching is included with Register C so the bits which are set are stable throughout the read cycle. All bits which are high when read by the program are cleared, and new interrupts (on any bits) are held until after the read cycle. One, two, or three flag bits may be found to be set when Register C is read. The program should inspect all utilized flag bits every time Register C is read to insure that no interrupts are lost.

The second flag bit usage method is with fully enabled interrupts. When an interrupt-flag bit is set and the corresponding interrupt-enable bit is also set, the $\overline{IRQ}$ pin is asserted low. $\overline{IRQ}$ is asserted as long as at least one of the three interrupt sources has its flag and enable bits both set. The IRQF bit in Register C is a "1" whenever the $\overline{IRQ}$ pin is being driven low.

The processor program can determine that the RTC initiated the interrupt by reading Register C. A "1" in bit 7 (IRQF bit) indicates that one or more interrupts have been initiated by the part. The act of reading Register C clears all the then-active flag bits, plus the IRQF bit. When the program finds IRQF set, it should look at each of the individual flag bits in the same byte which have the corresponding interrupt-mask bits set and service each interrupt which is set. Again, more than one interrupt-flag bit may be set.

DIVIDER STAGES

The MC146818 has 22 binary-divider stages following the time base as shown in Figure 1. The output of the dividers is a 1 Hz signal to the update-cycle logic. The dividers are controller by three divider bus (DV2, DV1, and DV0) in Register A.

DIVIDER CONTROL

The divider-control bits have three uses, as shown in Table 4. Three usable operating time bases may be selected (4.194304 MHz, 1.048576 MHz, or 32.768 kHz). The divider chain may be held reset, which allows precision setting of the time. When the divider is changed from reset to an operating time base, the first update cycle is one-half second later. The divider-control bits are also used to facilitate testing the MC146818.

TABLE 4 — DIVIDER CONFIGURATIONS

Time-Base Frequency	Divider Bits Register A DV2	DV1	DV0	Operation Mode	Divider Reset	Bypass First N-Divider Bits
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.

3

SQUARE-WAVE OUTPUT SELECTION

Fifteen of the 22 divider taps are made available to a 1-of-15 selector as shown in Figure 1. The first purpose of selecting a divider tap is to generate a square-wave output signal at the SQW pin. The RS0-RS3 bits in Register A establish the square-wave frequency as listed in Table 5. The SQW frequency selection shares the 1-of-15 selector with periodic interrupts.

Once the frequency is selected, the output of the SQW pin may be turned on and off under program control with the square-wave enable (SQWE) bit in Register B. Altering the divider, square-wave output selection bits, or the SQWE output-enable bit may generate an asymmetrical waveform at the time of execution. The square-wave output pin has a number of potential uses. For example, it can serve as a frequency standard for external use, a frequency synthesizer, or could be used to generate one or more audio tones under program control.

PERIODIC INTERRUPT SELECTION

The periodic interrupt allows the $\overline{IRQ}$ pin to be triggered from once every 500 ms to once every 30.517 µs. The periodic interrupt is separate from the alarm interrupt which may be output from once-per-second to once-per-day.

Table 5 shows that the periodic interrupt rate is selected with the same Register A bits which select the square-wave frequency. Changing one also changes the other. But each function may be separately enabled so that a program could switch between the two features or use both. The SQW pin is enabled by the SQWE bit in Register B. Similarly the periodic interrupt is enabled by the PIE bit in Register B.

Periodic interrupt is usable by practically all real-time systems. It can be used to scan for all forms of inputs from contact closures to serial receive bits or bytes. It can be used in multiplexing displays or with software counters to measure inputs, create output intervals, or await the next needed software function.

TABLE 5 — PERIODIC INTERRUPT RATE AND SQUARE WAVE OUTPUT FREQUENCY

Select Bits Register A				4.194304 or 1.048576 MHz Time Base		32.768 kHz Time Base	
RS3	RS2	RS1	RS0	Periodic Interrupt Rate 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.

To properly setup the internal counters for daylight savings time operation, the user must set the time at least two seconds before the rollover will occur. Likewise, the time must be set at least two seconds before the end of the 29th or 30th day of the month.

REGISTERS

The MC146818 has four registers which are accessible to the processor program. The four registers are also fully accessible during the update cycle.

REGISTER A ($0A)

MSB — LSB — Read/Write Register except UIP

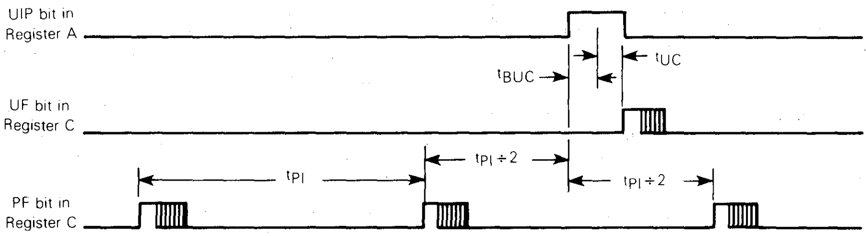
b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

UIP — The update in progress (UIP) bit is a status flag that may be monitored by the program. When UIP is a "1" the update cycle is in progress or will soon begin. When UIP is a "0" the update cycle is not in progress and will not be for at least 244 μs (for all time bases). This is detailed in Table 6. The time, calendar, and alarm information in RAM is fully available to the program when the UIP bit is zero — it is not in transition. The UIP bit is a read-only bit, and is not affected by Reset. Writing the SET bit in Register B to a "1" inhibit any update cycle and then clear the UIP status bit.

TABLE 6 — UPDATE CYCLE TIMES

UIP Bit	Time Base (OSC1)	Update Cycle Time (t_{UC})	Minimum Time Before Update Cycle (t_{BUC})
1	4.194304 MHz	248 μs	–
1	1.048576 MHz	248 μs	–
1	32.768 kHz	1984 μs	–
0	4.194304 MHz	–	244 μs
0	1.048576 MHz	–	244 μs
0	32.768 kHz	–	244 μs

FIGURE 16 — UPDATE-ENDED AND PERIODIC INTERRUPT RELATIONSHIPS



t_{PI} = Periodic Interrupt Time Interval (500 ms, 250 ms, 125 ms, 62.5 ms, etc. per Table 5)
t_{UC} = Update Cycle Time (248 μs or 1984 μs)
t_{BUC} = Delay Time Before Update Cycle (244 μs)

DV2, DV1, DV0 – Three bits are used to permit the program to select various conditions of the 22-stage divider chain. The divider selection bits identify which of the three time-base frequencies is in use. Table 4 shows that time bases of 4.194304 MHz, 1.048576 MHz, and 32.768 kHz may be used. The divider selection bits are also used to reset the divider chain. When the time/calendar is first initialized, the program may start the divider at the precise time stored in the RAM. When the divider reset is removed the first update cycle begins one-half second later. These three read/write bits are not affected by $\overline{\text{RESET}}$.

RS3, RS2, RS1, RS0 – The four rate selection bits select one of 15 taps on the 22-stage divider, or disable the divider output. The tap selected may be used to generate an output square wave (SQW pin) and/or a periodic interrupt. The program may do one of the following: 1) enable the interrupt with the PIE bit, 2) enable the SQW output pin with the SQWE bit, 3) enable both at the same time at the same rate, or 4) enable neither. Table 5 lists the periodic interrupt rates and the square-wave frequencies that may be chosen with the RS bits. These four bits are read/write bits which are not affected by $\overline{\text{RESET}}$.

REGISTER B ($0B)

MSB							LSB	
b7	b6	b5	b4	b3	b2	b1	b0	Read/Write Register
SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE	

SET – When the SET bit is a "0", the update cycle functions normally by advancing the counts once-per-second. When the SET bit is written to a "1", any update cycle in progress is aborted and the program may initialize the time and calendar bytes without an update occurring in the midst of initializing. SET is a read/write bit which is not modified by $\overline{\text{RESET}}$ or internal functions of the MC146818.

PIE – The periodic interrupt enable (PIE) bit is a read/write bit which allows the periodic-interrupt flag (PF) bit in Register C to cause the $\overline{\text{IRQ}}$ pin to be driven low. A program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3, RS2, RS1, and RS0 bits in Register A. A zero in PIE blocks $\overline{\text{IRQ}}$ from being initiated by a periodic interrupt, but the periodic flag (PF) bit is still set at the periodic rate. PIE is not modified by any internal MC146818 functions, but is cleared to "0" by a $\overline{\text{RESET}}$.

AIE – The alarm interrupt enable (AIE) bit is a read/write bit which when set to a "1" permits the alarm flag (AF) bit in Register C to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes (including a "don't care" alarm code of binary 11XXXXXX). When the AIE bit is a "0", the AF bit does not initiate an $\overline{\text{IRQ}}$ signal. The $\overline{\text{RESET}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

UIE – The UIE (update-ended interrupt enable) bit is a read/write bit which enables the update-end flag (UF) bit in Register C to assert $\overline{\text{IRQ}}$. The $\overline{\text{RESET}}$ pin going low or the SET bit going high clears the UIE bit.

SQWE – When the square-wave enable (SQWE) bit is set to a "1" by the program, a square-wave signal at the frequency specified in the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to a zero the SQW pin is held low. The state of SQWE is cleared by the $\overline{\text{RESET}}$ pin. SQWE is a read/write bit.

DM – The data mode (DM) bit indicates whether time and calendar updates are to use binary or BCD formats. The DM bit is written by the processor program and may be read by the program, but is not modified by any internal functions or $\overline{\text{RESET}}$. A "1" in DM signifies binary data, while a "0" in DM specifies binary-coded-decimal (BCD) data.

24/12 – The 24/12 control bit establishes the format of the hours bytes as either the 24-hour mode (a "1") or the 12-hour mode (a "0"). This is a read/write bit, which is affected only by software.

DSE – The daylight savings enable (DSE) bit is a read/write bit which allows the program to enable two special updates (when DSE is a "1"). On the last Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a "0". DSE is not changed by any internal operations or reset.

REGISTER C ($0C)

MSB							LSB	
b7	b6	b5	b4	b3	b	b1	b0	Read-Only Register
IRQF	PF	AF	UF	0	0	0	0	

IRQF – The interrupt request flag (IRQF) is set to a "1" when one or more of the following are true:

PF = PIE = "1"

AF = AIE = "1"

UF = UIE = "1"

i.e., IRQF = PF•PIE + AF•AIE + UF•UIE

Any time the IRQF bit is a "1", the $\overline{\text{IRQ}}$ pin is driven low. All flag bits are cleared after Register C is read by the program or when the $\overline{\text{RESET}}$ pin is low.

PF – The periodic interrupt flag (PF) is a read-only bit which is set to a "1" when a particular edge is detected on the selected tap of the divider chain. The RS3 to RS0 bits establish the periodic rate. PF is set to a "1" independent of the state of the PIE bit. PF being a "1" initiates an $\overline{\text{IRQ}}$ signal and sets the IRQF bit when PIE is also a "1". The PF bit is cleared by a $\overline{\text{RESET}}$ or a software read of Register C.

AF – A "1" in the AF (alarm interrupt flag) bit indicates that the current time has matched the alarm time. A "1" in the AF causes the $\overline{\text{IRQ}}$ pin to go low, and a "1" to appear in the IRQF bit, when the AIE bit also is a "1." A $\overline{\text{RESET}}$ or a read of Register C clears AF.

UF – The update-ended interrupt flag (UF) bit is set after each update cycle. When the UIE bit is a "1", the "1" in UF causes the IRQF bit to be a "1", asserting $\overline{\text{IRQ}}$. UF is cleared by a Register C read or a $\overline{\text{RESET}}$.

b3 TO b0 – The unused bits of Status Register 1 are read as "0's". They can not be written.

REGISTER D ($0D)

MSB — LSB

b7	b6	b5	b4	b3	b2	b1	b0
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.

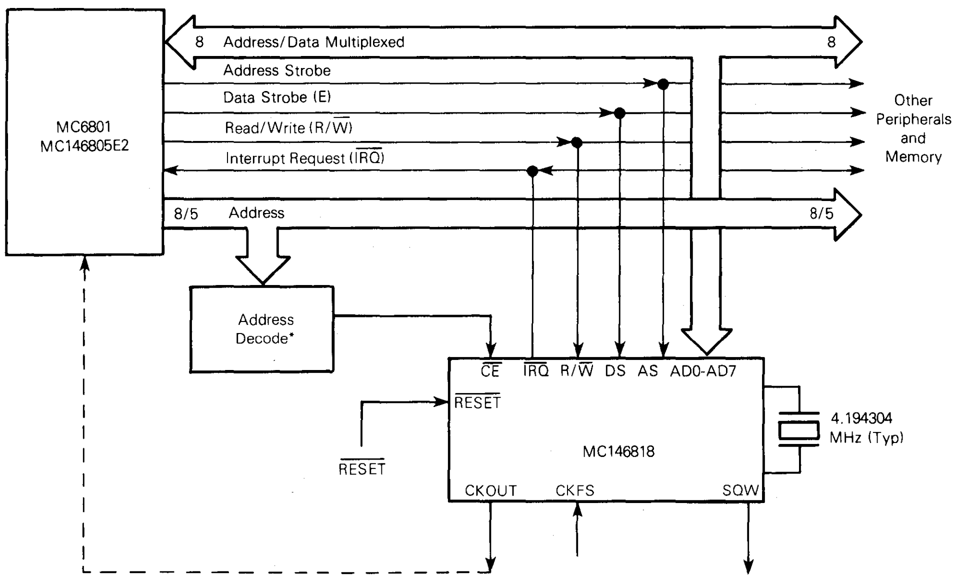
An example using either the Motorola MC6800, MC6802, MC6808, or MC6809 microprocessor is shown in Figure 21.

Figure 22 illustrates the subroutines which may be used for data transfers in a non-multiplexed system. The subroutines should be entered with the registers containing the following data:

Accumulator A: The address of the RTC to be accessed.
Accumulator B: Write: The data to be written.
Read: The data read from the RTC.

The RTC is mapped to two consecutive memory locations – RTC and RTC + 1 as shown in Figure 21.

FIGURE 17 – MC146818 INTERFACED WITH MOTOROLA COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS



*High-Speed Silicon-Gate CMOS or TTL Address Decoding

FIGURE 18 — MC146818 INTERFACED WITH
COMPETITOR COMPATIBLE MULTIPLEXED BUS MICROPROCESSORS

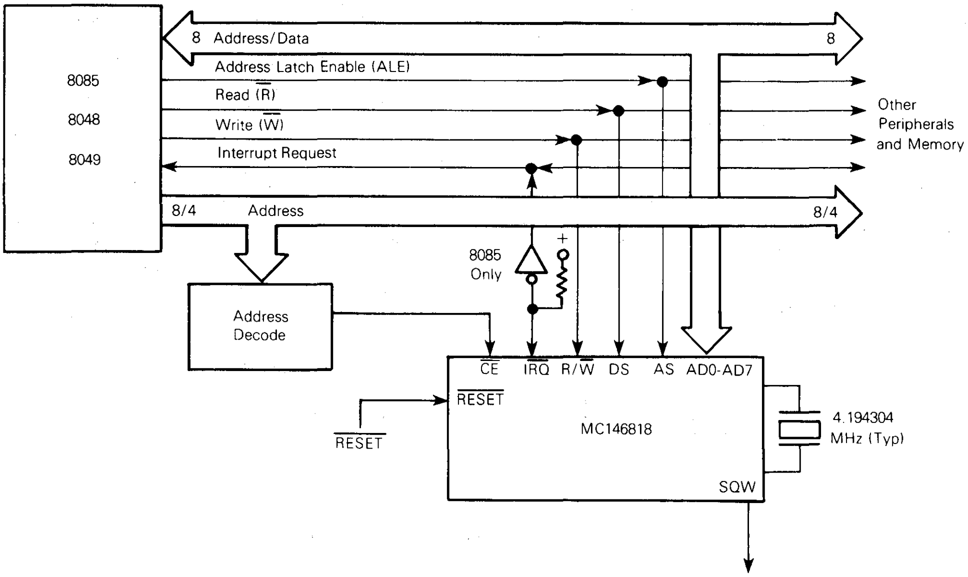
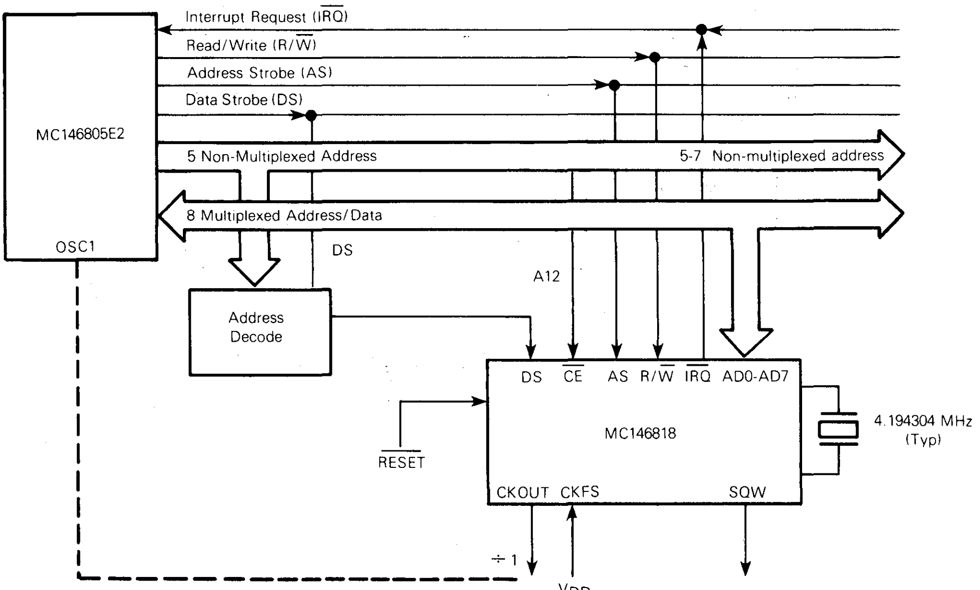


FIGURE 19 — MC146818 INTERFACE WITH MC146805E2
CMOS MULTIPLEXED MICROPROCESSOR WITH SLOW ADDRESSING DECODING



This illustrates the use of CMOS gating for address decoding.

FIGURE 20 — MC146818 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE CHIP MICROCOMPUTER

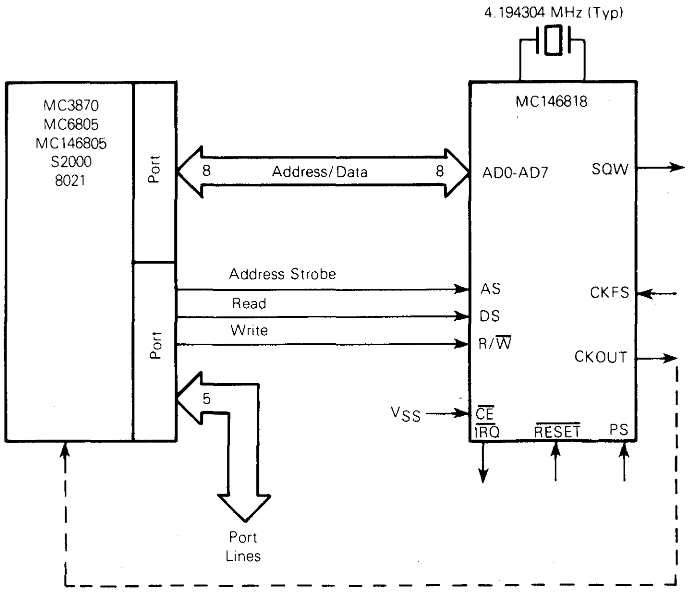


FIGURE 21 — MC146818 INTERFACED WITH MOTOROLA PROCESSORS

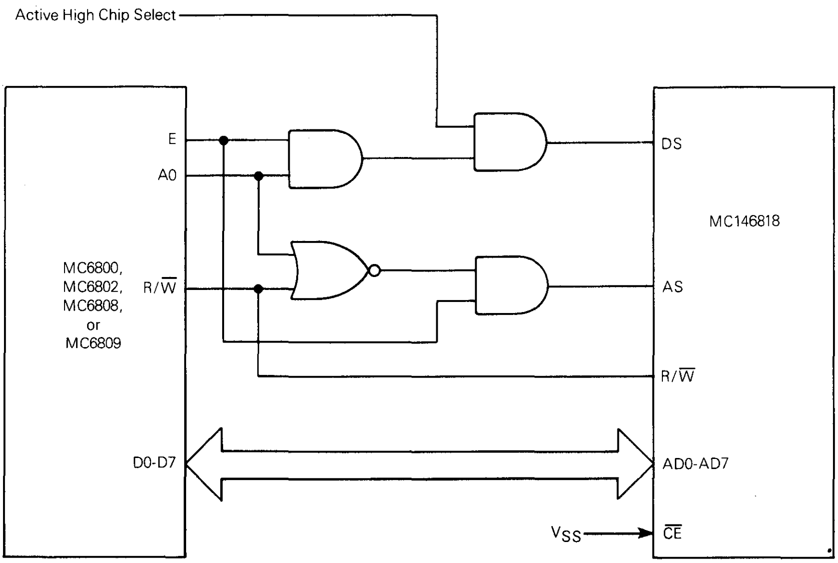


FIGURE 22 — SUBROUTINE FOR READING AND WRITING THE MC146818 WITH A NON-MULTIPLEXED BUS

```
READ    STA   RTC     Generate AS and Latch Data from ACCA
        LDAB  RTC+1   Generate DS and Get Data
        RTS

WRITE   STA   RTC     Generate AS and Latch Data from ACCA
        STAB  RTC+1   Generate DS and Store Data
        RTS
```


IMPORTANT NOTICES

Those devices made with date code 3N4GXXXX have the following exceptions when used in the Motorola mode of MOTEL.

1. $V_{DD} = 3$ to 5.25 V for operation
2. DS $V_{IL} = 0.6$ V Max.

The falling edge of chip select should occur during the active high pulse of address strobe, only on those units with date code GC6XXXX.

MOTOROLA

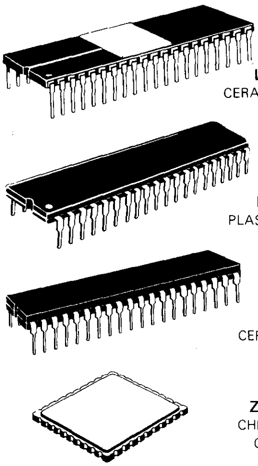
MC146823

Advance Information

CMOS

(HIGH-DENSITY HIGH-PERFORMANCE SILICON-GATE)

PARALLEL INTERFACE



L SUFFIX
CERAMIC PACKAGE
CASE 715

P SUFFIX
PLASTIC PACKAGE
CASE 711

S SUFFIX
CERDIP PACKAGE
CASE 734

Z SUFFIX
CHIP CARRIER
CASE 761

CMOS PARALLEL INTERFACE

The MC146823 CMOS parallel interface (CPI) provides a universal means of interfacing external signals with the MC146805E2 CMOS microprocessor and other multiplexed bus microprocessors. The unique MOTEL circuit on-chip allows direct interfacing to most industry CMOS microprocessors, as well as many NMOS MPUs.

The MC146823 CPI includes three bidirectional 8-bit ports or 24 I/O pins. Each I/O line may be separately established as an input or an output under program control via data direction registers associated with each port. Using the bit change and test instructions of the MC146805E2, each individual I/O pin can be separately accessed. All port registers are read/write bytes to accommodate read-modify-write instructions. Features include:

- 24 Individually Programmed I/O Pins
- MOTEL Circuit for Bus Compatibility with Many Microprocessors
- Multiplexed Bus Compatibility with: MC146805E2, MC6801, MC6803, and Competitive Microprocessors
- Data Direction Registers for Ports A, B, and C
- Four Port C I/O Pins May Be Used as Control Lines for:
  - Four Interrupt Inputs
  - Input Byte Latch
  - Output Pulse
  - Handshake Activity
- 15 Registers Addressed as Memory Locations
- Handshake Control Logic for Input and Output Peripheral Operation
- Interrupt Output Pin
- Reset Input to Clear Interrupts and Initialize Internal Registers
- 3.0 Volt to 5.5 Volt Operating V_{DD}

PIN ASSIGNMENT

PC2	1	40	V_{DD}
PC1	2	39	PC3
PC0	3	38	PC4/CA1
PA0	4	37	PC5/CA2
PA1	5	36	PC6/CB1
PA2	6	35	PC7/CB2
PA3	7	34	PB0
PA4	8	33	PB1
PA5	9	32	PB2
PA6	10	31	PB3
PA7	11	30	PB4
AD0	12	29	PB5
AD1	13	28	PB6
AD2	14	27	PB7
AD3	15	26	$\overline{IRQ}$
AD4	16	25	$\overline{RESET}$
AD5	17	24	DS
AD6	18	23	R/$\overline{W}$
AD7	19	22	AS
V_{SS}	20	21	$\overline{CE}$

Pin assignments are the same for both the dual-in-line and chip carrier package.

ORDERING INFORMATION

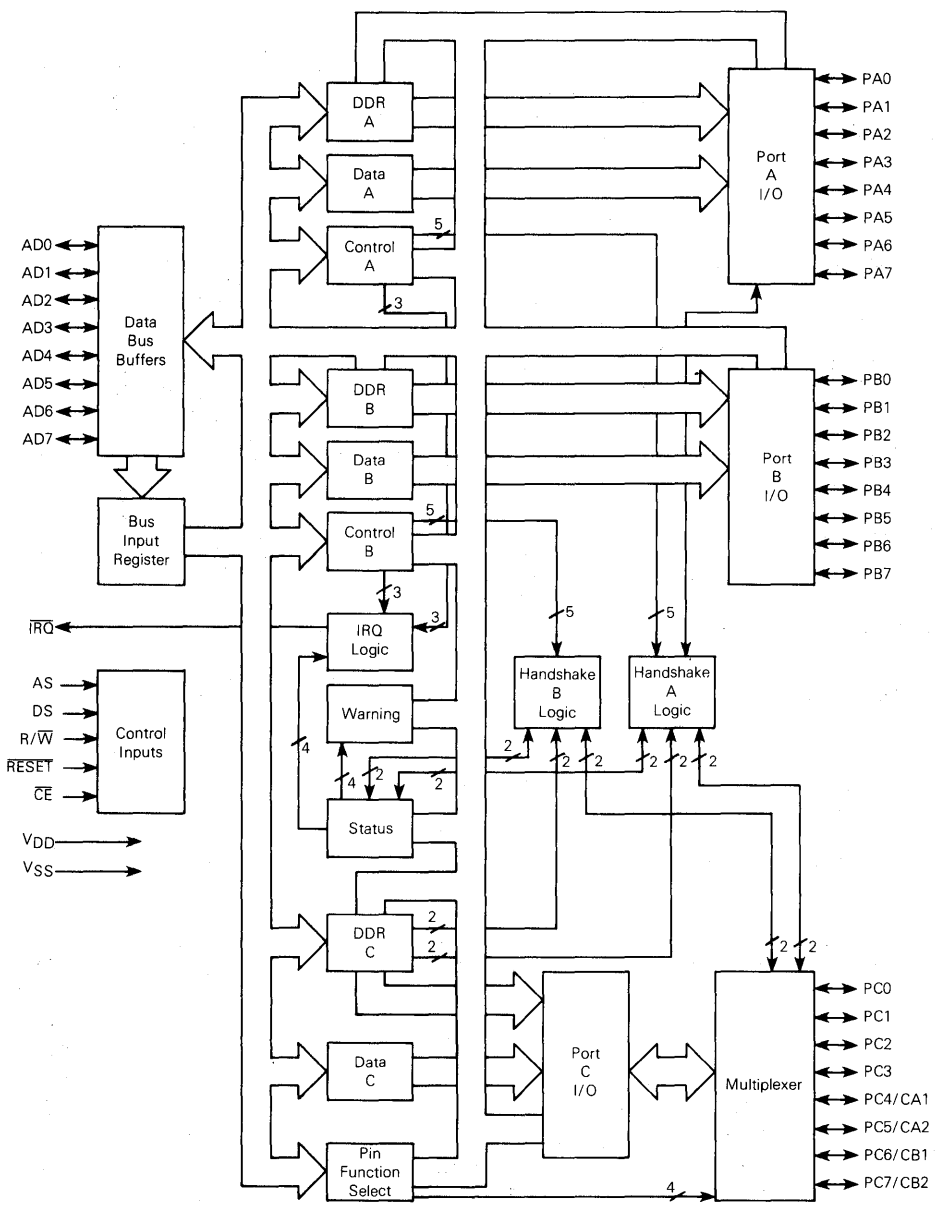
($T_A = 0°C$ to $+70°C$)

Package Type	Order Number
Plastic – P Suffix	MC146823P
Ceramic (Side Brazed) – L Suffix	MC146823L
Cerdip – S Suffix	MC146823S
Chip Carrier – Z Suffix	MC146823Z

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

3

BLOCK DIAGRAM



MAXIMUM RATINGS (Voltages reference to V_{SS})

Ratings	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +8.0	V
All Input Voltages	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	10	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

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

THERMAL CHARACTERISTICS

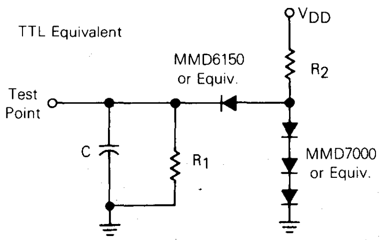
Characteristics	Symbol	Value	Unit
Thermal Resistance			
Ceramic	θ_{JA}	50	°C/W
Plastic		100	
Cerdip		60	
Chip Carrier		TBD	

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, unless otherwise noted)

Parameter	Symbol	Min	Max	Unit
Output Voltage ($I_{Load} \leq 10\ \mu A$)	V_{OL}	–	0.1	V
	V_{OH}	$V_{DD}-0.1$	–	V
Output High Voltage				
(I_{Load} = −1.6 mA) AD0-AD7	V_{OH}	4.1	V_{DD}	
(I_{Load} = −0.2 mA) PA0-PA7, PC0-PC7	V_{OH}	4.1	V_{DD}	V
(I_{Load} = −0.36 mA) PB0-PB7	V_{OH}	4.1	V_{DD}	
Output Low Voltage				
(I_{Load} = 1.6 mA) AD0-AD7, PB0-PB7	V_{OL}	V_{SS}	0.4	
(I_{Load} = 0.8 mA) PA0-PA7, PC0-PC7	V_{OL}	V_{SS}	0.4	V
(I_{Load} = 1.0 mA) $\overline{IRQ}$	V_{OL}	V_{SS}	0.4	
Input High Voltage, AD0-AD7, AS, DS, R/$\overline{W}$, $\overline{CE}$, PA0-PA7, PB0-PB7, PC0-PC7	V_{IH}	$V_{DD}-2.0$	V_{DD}	V
$\overline{RESET}$	V_{IH}	$V_{DD}-0.8$	V_{DD}	
Input Low Voltage (All Inputs)	V_{IL}	V_{SS}	0.8	V
Quiescent Current — No dc Loads (All Ports Programmed as Inputs, All Inputs = V_{DD} − 0.2 V)	I_{DD}	–	160	μA
Total Supply Current (All Ports Programmed as Inputs, CE = V_{IL}, t_{cyc} = 1 μs)	I_{DD}	–	3.0	mA
Input Current, $\overline{CE}$, AS, R/$\overline{W}$, DS, $\overline{RESET}$	I_{in}	–	±1.0	μA
Hi-Z State Leakage, AD0-AD7, PA0-PA7, PB0-PB7, PC0-PC7	I_{TSL}	–	±10.0	μA

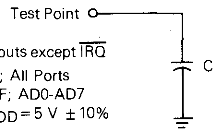
EQUIVALENT TEST LOADS

TTL Equivalent

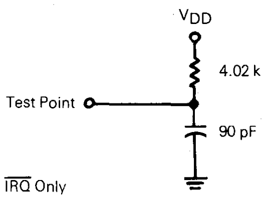


Pin	R1	R2	C
AD0-AD7	2.55k	2.0k	130 pF
PA0-PA7, PC0-PC7	20.0k	4.32k	50 pF
PB0-PB7	11.5k	2.1k	50 pF

CMOS Equivalent



For all outputs except $\overline{IRQ}$
C = 50 pF; All Ports
= 130 pF; AD0-AD7
for V_{DD} = 5 V ±10%



$\overline{IRQ}$ Only

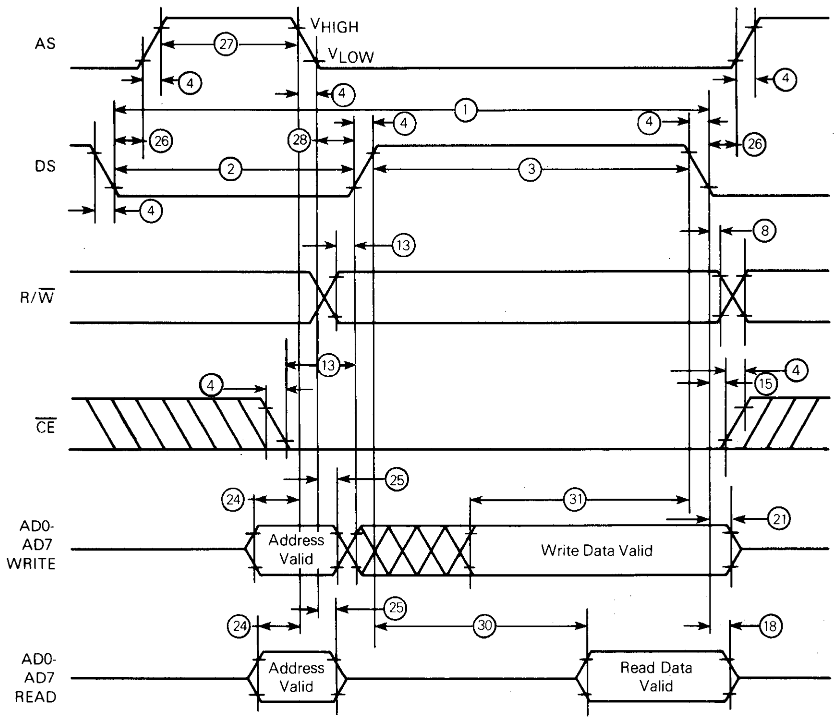
BUS TIMING (V_{DD} = 5 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0° to 70°C, unless otherwise noted)

Ident. Number	Characteristics	Symbol	Min	Max	Unit
1	Cycle Time	t_{cyc}	1000	dc	ns
2	Pulse Width, 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/$\overline{W}$ and $\overline{CE}$ Setup Time Before DS/E	t_{RWS}	25	–	ns
15	Chip Enable Hold Time	t_{CH}	0	–	ns
18	Read Data Hold Time	t_{DHR}	10	100	ns
21	Write Data Hold Time	t_{DHW}	0	–	ns
24	Muxed Address Valid Time to AS/ALE Fall	t_{ASL}	25	–	ns
25	Muxed Address Hold Time	t_{AHL}	20	–	ns
26	Delay Time DS/E to AS/ALE Rise	t_{ASD}	60	–	ns
27	Pulse Width, AS/ALE High	PW_{ASH}	170	–	ns
28	Delay Time, AS/ALE to DS/E Rise	t_{ASED}	60	–	ns
30	Peripheral Output Data Delay Time from DS/E or $\overline{RD}$	t_{DDR}	20	240	ns
31	Peripheral Data Setup Time	t_{DSW}	220	–	ns

NOTE: Designations E, ALE, $\overline{RD}$, and $\overline{WR}$ refer to signals from alternative microprocessor signals.

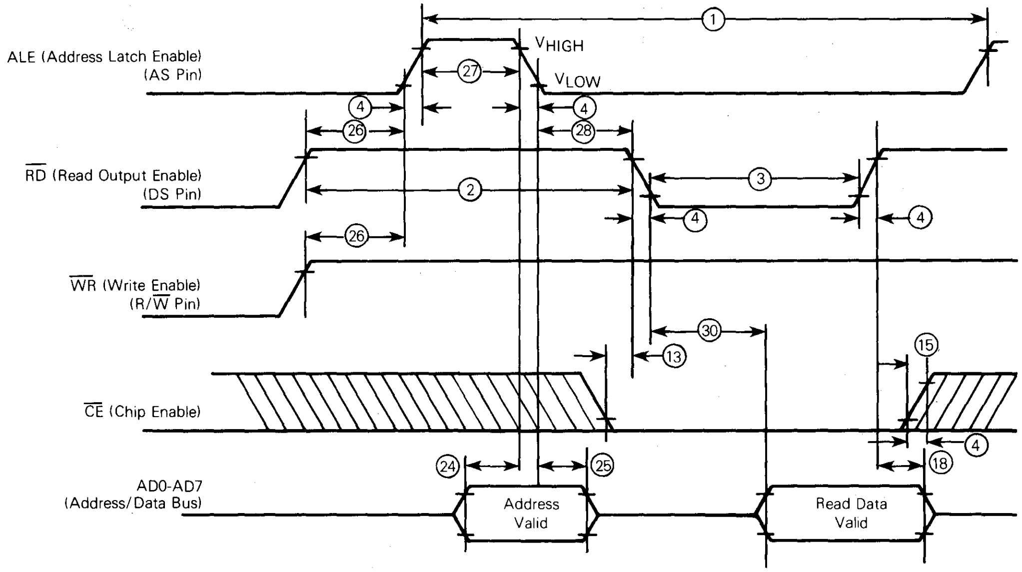
3

BUS TIMING DIAGRAM

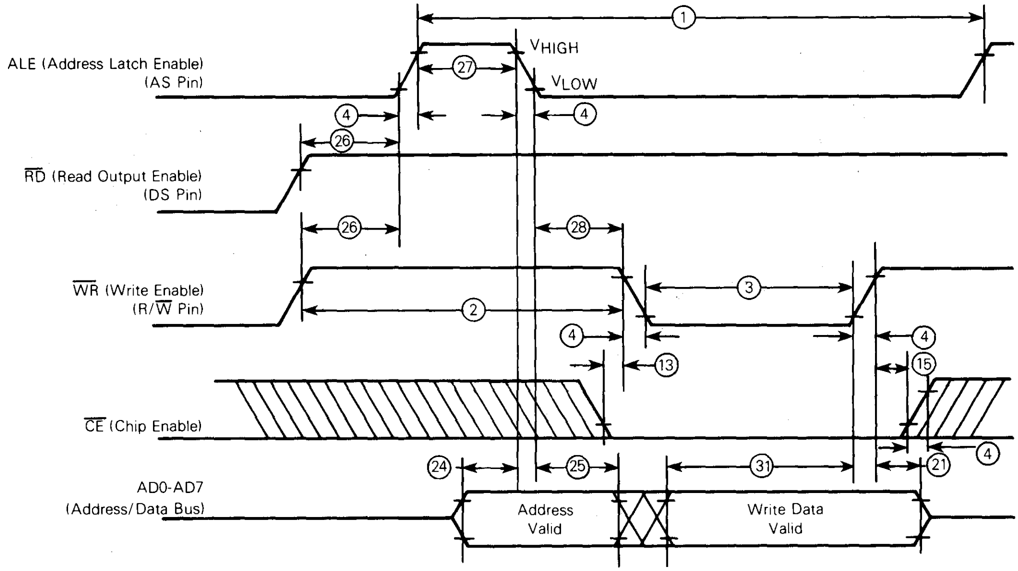


NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V ± 10%

BUS READ TIMING COMPETITOR MULTIPLEXED BUS



BUS WRITE TIMING COMPETITOR MULTIPLEXED BUS



NOTE: $V_{HIGH} = V_{DD} - 2.0$ V, $V_{LOW} = 0.8$ V, for $V_{DD} = 5.0$ V $\pm 10\%$

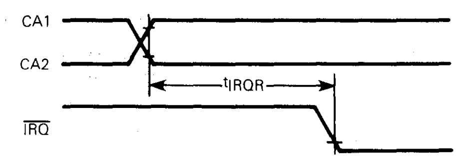
CONTROL TIMING (V_{DD}=5.0 Vdc ±10%, V_{SS}=0 Vdc, T_A=0°C to 70°C)

Parameter	Symbol	Min	Max	Unit
Interrupt Response (Input Modes 1 and 3)	t_{IRQR}	TBD	–	μs
Delay, CA1 (CB1) Active Transition to CA2 (CB2) High (Output Mode 0)	t_{C2}	TBD	–	μs
Delay, CA2 Transition from Positive Edge of AS (Output Modes 0 and 1)	t_{A2}	TBD	–	μs
Delay, CB2 Transition from Negative Edge of AS (Output Modes 0 and 1)	t_{B2}	TBD	–	μs
CA2/CB2 Pulse Width (Output Mode 1)	t_{PW}	TBD	TBD	ns
Delay, V_{DD} Rise to $\overline{RESET}$ High	t_{RLH}	TBD	–	μs
Pulse Width, $\overline{RESET}$	t_{RW}	TBD	–	ns

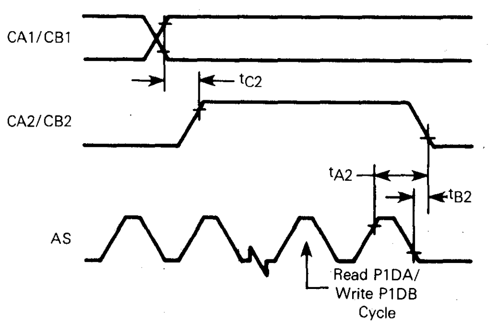
TBD = To be determined.

CONTROL TIMING DIAGRAMS

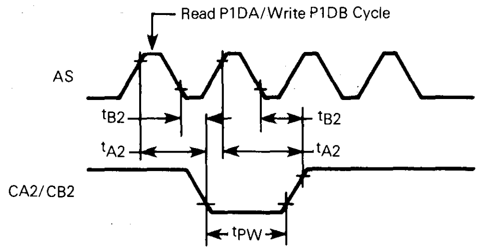
$\overline{IRQ}$ RESPONSE (INPUT MODES 1 AND 3)



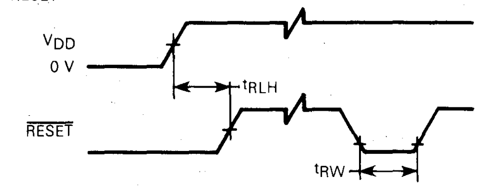
CA2/CB2 DELAY (OUTPUT MODE 0)



CA2/CB2 DELAY (OUTPUT MODE 1)



$\overline{RESET}$



GENERAL DESCRIPTION

The MC146823, CMOS parallel interface (CPI), contains 24 individual bidirectional I/O lines configured in three 8-bit ports. The 15 internal registers, which control the mode of operation and contain the status of the port pins, are accessed via an 8-bit multiplexed address/data bus. The lower four address bits (AD0-AD3) of the multiplexed address bus determine which register is to be accessed (see Figure 1). The four address bits (AD4, AD5, AD6, and AD7) must be separately decoded to position this memory map within each 256 byte address space available via the 8-bit multiplexed address bus. For more detailed information refer to **REGISTER DESCRIPTION.**

FIGURE 1 — REGISTER ADDRESS MAP

0	Port A Data, Clear CA1 Interrupt	P1DA
1	Port A Data, Clear CA2 Interrupt	P2DA
2	Port A Data	PDA
3	Port B Data	PDB
4	Port C Data	PDC
5	Not Used	—
6	Data Direction Register for Port A	DDRA
7	Data Direction Register for Port B	DDRB
8	Data Direction Register for Port C	DDRC
9	Control Register for Port A	CRA
A	Control Register for Port B	CRB
B	Pin Function Select Register for Port C	FSR
C	Port B Data, Clear CB1 Interrupt	P1DB
D	Port B Data, Clear CB2 Interrupt	P2DB
E	Handshake/Interrupt Status Register	HSR
F	Handshake Over-Run Warning Register	HWR

The CPI is implemented with the MOTEL circuit which allows direct interface with either of the two major multiplexed microprocessor bus types. A detailed description of the MOTEL circuit is provided in the **MOTEL** section.

Three data direction registers (DDRs), one for each port, determine which pins are outputs and which are inputs. A logic zero on a DDR bit configures its associated pin as an input; and a logic one configures the pin as an output. Upon reset, the DDRs are cleared to logic zero to configure all port pins as inputs.

Actual port data may be read or written via the port data registers (PDA, PDB, and PDC). Ports A and B each have two additional data registers (P1DA and P2DA — P1DB and P2DB) which are used to clear the associated handshake/interrupt status register bits (HSA1 and HSA2 — HSB1 and HSB2), respectively. Port A may also be configured as an 8-bit latch when used with CA1. Reset has no effect on the contents of the port data registers. Users are advised to initialize the port data registers before changing any port pin to an output.

Four pins on port C (PC4/CA1, PC5/CA2, PC6/CB1, and PC7/CB2) may additionally be programmed as handshake lines for ports A and B via the port C function select register (FSR). Both ports A and B have one input-only line and one bidirectional handshake line each associated with them. The handshake lines may be programmed to perform a variety of tasks such as interrupt requests, setting flags, latching data, and data transfer requests and/or acknowledgements. The handshake functions are programmed via control registers A and B (CRA and CRB). Additional information may be found in **PIN DESCRIPTIONS, REGISTER DESCRIPTION,** or **HANDSHAKE OPERATION.**

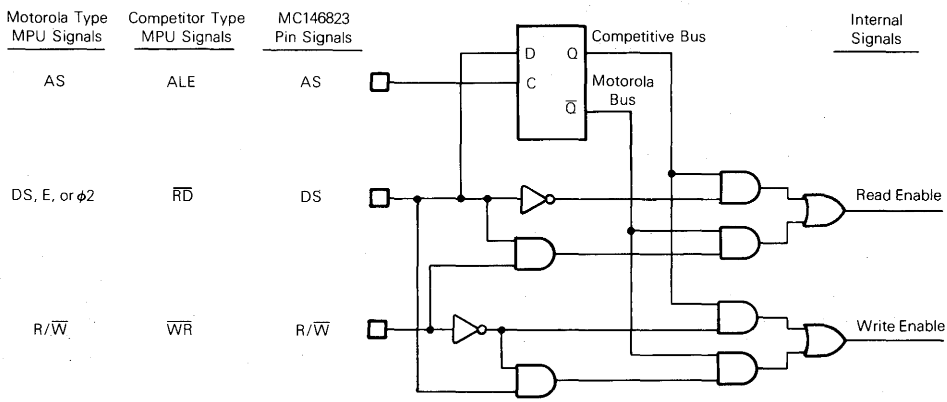
MOTEL

The MOTEL circuit is a concept that permits the MC146823 to be directly interfaced with different types of multiplexed bus microprocessors without any additional external logic. For a more detailed description of the multiplexed bus, see **MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7).** Most multiplexed microprocessors use one of two synchronous buses to interface peripherals. One bus was originated by Motorola in the MC6803 and the other by Intel in the 8085.

The MOTEL circuit (for MOTorola and intEL bus) is built into peripheral and memory ICs to permit direct connection to either type of bus. A functional diagram of the MOTEL circuit is shown in Figure 2.

3

FIGURE 2 — FUNCTIONAL DIAGRAM OF MOTEL CIRCUIT



The microprocessor type is automatically selected by the MOTEL circuit through latching the state of the DS/$\overline{RD}$ pin with AS/ALE. Since DS is always low during AS and $\overline{RD}$ is always high during ALE, the latch automatically indicates with which type microprocessor bus it is interfaced.

PIN DESCRIPTIONS

The following paragraphs contain a brief description of the input and output pins. References (if applicable) are given to other paragraphs that contain more detail about the function being performed.

MULTIPLEXED BIDIRECTIONAL ADDRESS/DATA BUS (AD0-AD7)

Multiplexed bus processors save pins by presenting the address during the first portion of the bus cycle and using the same pins during the second portion of the bus cycle for data. Address-then-data multiplexing does not slow the access time of the MC146823 since the bus reversal from address to data is occurring during the internal register access time.

The address must be valid t_{ASL} prior to the fall of AS/ALE at which time the MC146823 latches the address present on the AD0-AD3 pins. Valid write data must be presented and held stable during the latter portion of the DS or $\overline{WR}$ pulses. In a read cycle, the MC146823 outputs eight bits of data during the latter portion of the DS or $\overline{RD}$ pulses, then ceases driving the bus (returns the output drivers to high impedance) t_{DHR} hold time after DS falls in the Motorola case of MOTEL or $\overline{RD}$ rises in the other case.

ADDRESS STROBE (AS)

The address strobe input pulse serves to demultiplex the bus. The falling edge of AS or ALE causes the addresses AD0-AD3 to be latched within the MC146823. The automatic MOTEL circuit in the MC146823 also latches the state of the DS pin with the falling edge of AS or ALE.

DATA STROBE OR READ (DS)

The DS input pin has two interpretations via the MOTEL circuit. When generated by a Motorola microprocessor, DS is a positive pulse during the latter portion of the bus cycle, and is variously called DS (data strobe), E (enable), or phase 2 (phase 2 clock). During read cycles, DS or $\overline{RD}$ signifies the time that the CPI is to drive the bidirectional bus. In write cycles, the trailing edge of DS or rising edge of $\overline{WR}$ causes the parallel interface to latch the written data present on the bidirectional bus.

The second MOTEL interpretation of DS is that of $\overline{RD}$, MEMR, or I/OR originating from the competitor's microprocessor. In this case, DS identifies the time period when the parallel interface drives the bus with read data. This interpretation of DS is also the same as an output-enable signal on a typical memory.

The MOTEL circuit, within the MC146823, latches the state of the DS pin on the falling edge of AS/ALE. When the Motorola mode of MOTEL is desired DS must be low during AS/ALE, which is the case with the Motorola multiplexed bus microprocessors. To insure the competitor mode of MOTEL, the DS pin must remain high during the time AS/ALE is high.

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

The MOTEL circuit treats the R/$\overline{W}$ input pin in one of two ways. First, when a Motorola microprocessor is connected, R/$\overline{W}$ is a level which indicates whether the current cycle is a read or write. A read cycle is indicated with a high level on R/$\overline{W}$ while DS is high, whereas a write cycle is a low on R/$\overline{W}$ while DS is high.

The second interpretation of R/$\overline{W}$ is as a negative write pulse, $\overline{WR}$, $\overline{MEMW}$, and $\overline{I/OW}$ from competitor's microprocessors. The MOTEL circuit in this mode gives the R/$\overline{W}$ pin the same meaning as the write ($\overline{W}$) pulse on many generic RAMs.

CHIP ENABLE ($\overline{CE}$)

The $\overline{CE}$ input signal must be asserted (low) for the bus cycle in which the MC146823 is to be accessed. $\overline{CE}$ is not latched and must be stable prior to and during DS (in the Motorola case of MOTEL) and prior to and during $\overline{RD}$ and $\overline{WR}$ (in the other MOTEL case). Bus cycles which take place without asserting $\overline{CE}$ cause no actions to take place within the MC146823. When $\overline{CE}$ is high, the multiplexed bus output is in a high-impedance state.

When $\overline{CE}$ is high, all data, DS, and R/$\overline{W}$ inputs from the microprocessor are disconnected within the MC146823. This permits the MC146823 to be isolated from a powered-down microprocessor.

RESET ($\overline{RESET}$)

The $\overline{RESET}$ input pin is an active-low line that is used to restore all register bits, except the port data register bits, to logical zeros. After reset, all port lines are configured as inputs and no interrupt or handshake lines are enabled.

INTERRUPT REQUEST ($\overline{IRQ}$)

The $\overline{IRQ}$ output line is an open-drain active-low signal that may be used to interrupt the microprocessor with a service request. The "open-drain" output allows this and other interrupt request lines to be wire ORed with a pullup resistor. The $\overline{IRQ}$ line is low when bit 7 of the status register is high. Bit 7 (IRQF) of the handshake/interrupt status register (HSR) is set if any enabled handshake transition occurs; and its associated control register bit is set to allow interrupts. Refer to **INTERRUPT DESCRIPTION** or **HANDSHAKE OPERATION** for additional information.

PORT A, BIDIRECTIONAL I/O LINES (PA0-PA7)

Each line of port A, PA0-PA7, is individually programmable as either an input or output via its data direction register (DDRA). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. See Figure 3 for typical I/O circuitry and Table 1 for I/O operation.

There are three data registers associated with port A: PDA, P1DA, and P2DA. P1DA and P2DA are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to the port A data register, PDA, is latched into the port A output latch regardless of the state of the DDRA. Data written to P1DA or P2DA is ignored and has no affect upon the output data latch or the I/O lines. An MPU read of port bits programmed as outputs reflect the last value written to the PDA register. Port A pins programmed as inputs may be latched via the handshake line PC4/CA1 (see

FIGURE 3 — TYPICAL PORT I/O CIRCUITRY

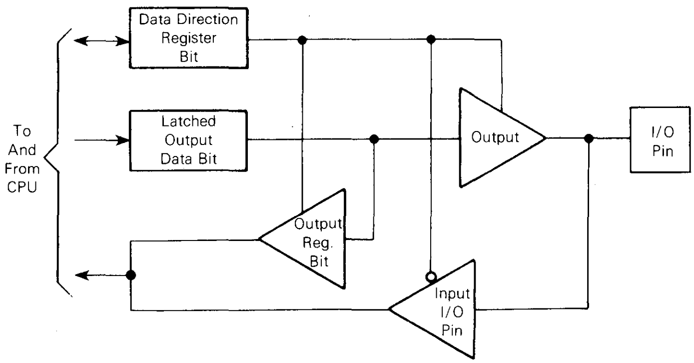


TABLE 1 — PORT DATA REGISTER ACCESSES (ALL PORTS)

R/$\overline{W}$	DDR Bit	Results
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

HANDSHAKE OPERATION) and latched input data may be read via any of the three port A data registers. If the port A input latch feature is not enabled, an MPU read of any port A data register reflects the current status of the port A input pins if the corresponding DDRA bits equal zero. Reset has no effect upon the contents of the port A data register; however, all pins will be placed in the input mode (all DDRA bits forced to equal zero) and all handshake lines will be disabled.

PORT B BIDIRECTIONAL I/O LINES (PB0-PB7)

Each line of port B, PB0-PB7, is individually programmable as either an input or an output via its data direction register (DDRB). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one.

There are three data registers associated with port B: PDB, P1DB, and P2DB. PDB is used for simple port B data reads and writes. P1DB and P2DB are accessed when certain handshake activity is desired. See **HANDSHAKE OPERATION** for more information.

Data written to PDB or P1DB data register is latched into the port B output latch regardless of the state of the DDRB. An MPU read of port bits programmed as outputs reflect the last value written to a port B data register. An MPU read of any port B register reflects the current status of the input pins whose DDRB bits equal zero. Reset has no effect upon the contents of the port B data register; however, all pins will be placed in the input mode (all DDRB bits forced to equal zero) and all handshake lines will be disabled.

PORT C, BIDIRECTIONAL I/O LINES (PC0-PC3)

Each line of port C, PC0-PC3, is individually programmable as either an input or an output via its data direction register (DDRC). An I/O pin is an input when its corresponding DDR bit is a logic zero and an output when the DDR bit is a logic one. Port C data register (PDC) is used for simple port C data reads and writes.

Data written into PDC is latched into the port C data latch regardless of the state of the DDRC. An MPU read of port C bits programmed as outputs reflect the last value written to the PDC register. An MPU read of the port C register reflects the current status of the corresponding input pins whose DDRC bits equal zero. Reset has no effect upon the contents of the port C data register; however, all pins will be placed in the input mode (all DDRC bits forced to equal zero) and all handshake lines will be disabled.

PORT C BIDIRECTIONAL I/O LINE OR PORT A INPUT HANDSHAKE LINE (PC4/CA1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC4/CA1 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC4/CA1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT A BIDIRECTIONAL HANDSHAKE LINE (PC5/CA2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port A via the port C function select register (FSR). If programmed as a port C I/O pin, PC5/CA2 performs as described in the PC0-PC3 pin description. If programmed as a port A handshake line, PC5/CA2 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B INPUT HANDSHAKE LINE (PC6/CB1)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O pin, PC6/CB1 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC6/CB1 performs as described in **HANDSHAKE OPERATION.**

PORT C BIDIRECTIONAL I/O LINE OR PORT B BIDIRECTIONAL HANDSHAKE LINE (PC7/CB2)

This line may be programmed as either a simple port C I/O line or as a handshake line for port B via the port C function select register (FSR). If programmed as a port C I/O line, PC7/CB2 performs as described in the PC0-PC3 pin description. If programmed as a port B handshake line, PC7/CB2 performs as described in **HANDSHAKE OPERATION.**

HANDSHAKE OPERATION

Up to four port C pins can be configured as handshake lines for ports A and B (one input-only and one bidirectional line for each port) via the port C function select register (FSR). The direction of data flow for the two bidirectional handshake lines (CA2 and CB2) is determined by bits 5 and 7, respectively, of the port C data direction register (DDRC). Actual handshake operation is defined by the appropriate port control register (CRA or CRB).

The control registers allow each handshake line to be programmed to operate in one of four modes. CA2 and CB2 each have four input and four output modes. For detailed information, see Tables 2 and 3.

A summary of the handshake modes is given in the input and output sections that follow. All handshake activity is disabled by reset.

INPUT

Handshake lines programmed as inputs operate in any of four different modes as defined by the control registers (see Table 2). A bit in the handshake/interrupt status register (HSR) is set to a logic one on an active transition of any handshake line programmed as an input. Modes 0 and 1 define a negative transition as active; modes 2 and 3 define a positive transition as active. If modes 1 or 3 are selected on any input handshake line then the active transition of that line results in the IRQF bit of the HSR being set to a logic one and causes the interrupt line ($\overline{IRQ}$) to go low. $\overline{IRQ}$ is released by clearing the HSR bits that are input handshake lines which have interrupts enabled.

If an active transition occurs while the associated HSR bit is set to a logic one, the corresponding bit in the handshake warning register (HWR) is set to a logic one indicating that service of at least one active transition was missed. An HWR bit is cleared to a logic zero by first accessing the appropriate port data register, to clear the appropriate HSR status bit, followed by a read of the HWR.

TABLE 2 — INPUT HANDSHAKE MODES

Mode	Control Register Bits*	Active Edge	Status Bit In HSR	$\overline{IRQ}$ Pin
0	00	– Edge	Set high on active edge.	Disabled
1	01	– Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.
2	10	+ Edge	Set high on active edge.	Disabled
3	11	+ Edge	Set high on active edge.	Goes low when corresponding status flag in HSR goes high.

* Cleared to logic zero on reset.

TABLE 3 — OUTPUT HANDSHAKE LINES (CA2 AND CB2 ONLY)

Mode	Control Register CRA(B) Bits 3 and 4*	Handshake Line Set High	Handshake Line Cleared Low	Default Level
0	00	Handshake set high on active transition of CA1 input. Handshake set high on active transition of CB1 input.	Read of P1DA or a read of P2DA while HSA1 is cleared. Write of port B P1DB or write of P2DB while HSB1 is cleared.	High
1	01	High on the first positive (negative) transition of AS while CA2 (CB2) is low.	Low on the first positive (negative) transition on AS following a read (write) of port A(B) data registers P1DA(B) or P2DA(B).	High
2	10	Never	Always	Low
3	11	Always	Never	High

* Cleared to logic zero on reset.

INPUT LATCH

Port A input-only handshake line (PC4/CA1) can be programmed to function as a latch enable for port A input data via CA1 LE (bit 2 of CRA). If CA1 LE is programmed to a logic one, an active transition of PC4/CA1 will latch the current status of the port A input pins into all three port A data registers (PDA, P1DA, and P2DA). When CA1 LE is enabled, port A and PC4/CA1 function as an 8-bit transparent latch; that is, if the HSA1 bit in the HSR is a logic zero then a read of any port A register reflects the current state of the port A input pins and corresponding bits of the output data latch for port A output pins. If HSA1 is a logic one, a read of any port A data register reflects the state of the port A input pins when HSA1 was set and the corresponding bits of the port A output data latch for port A output pins.

Further transitions of PC4/CA1 result only in setting the HWA1 bit in the HWR and do not relatch data into the port A registers. Latched data is released only by clearing HSA1 in the HSR to a logic zero (HSA1 is cleared by reading P1DA).

OUTPUT

Each bidirectional handshake line programmed as an output by the DDRC operates in one of four modes as described in Table 3. Modes 2 and 3 force the output handshake line to reflect the state of bit 4 in the appropriate control register.

In modes 0 and 1, PC5/CA2 is forced low during the cycle following a read of P1DA or a read of P2DA while HSA1 is cleared. PC7/CB2 is forced low during the cycle following a write to P1DB or a write to P2DB while HSB1 is cleared. Because of these differences, port A is the preferred input port and port B is the preferred output port.

In mode 0, PC5/CA2 (PC7/CB2) is set high by an active transition of PC4/CA1 (PC6/CB1). In mode 1, PC5/CA2 (PC7/CB2) is set high in the cycle following the cycle in which PC5/CA2 (PC7/CB2) goes low. Mode 1 forces a low-going pulse on PC5/CA2 (PC7/CB2) following a read (write) of P1DA (P1DB) or P2DA (P2DB) that is approximately one cycle time wide.

When entering an output handshake mode for the first time after a reset, the handshake line outputs the default level as listed in Table 3.

INTERRUPT DESCRIPTION

The MC146823 allows an MPU interrupt request ($\overline{IRQ}$ low) via the input handshake lines. The input handshake line, operating in modes 1 or 3 as defined by the control registers (CRA and CRB), causes $\overline{IRQ}$ to go low when IRQF (interrupt flag) in the HSR is set to a logic one. $\overline{IRQ}$ is released when IRQF is cleared. See **Handshake/Interrupt Status Register** under **REGISTER DESCRIPTION** for additional information.

REGISTER DESCRIPTION

The MC146823 has 15 registers (see Figure 1) which define the mode of operation and status of the port pins. The following paragraphs describe these registers.

Register Names:

Control Register A (CRA)
Control Register B (CRB)

Register Addresses:

$9 (CRA)
$A (CRB)

Register Bits:

	7	6	5	4 3	2	1 0
$9	X	X	X	CA2 Mode	CA1 LE	CA1 Mode
$A	X	X	X	CB2 Mode	X	CB1 Mode

Purpose:

These two registers control the handshake and interrupt activity for those pins defined as handshake lines by the port C function select register (FSR).

Description:

CA2 and CB2 are programmed as inputs or outputs via the associated DDRC bits. Each handshake line is controlled by two mode bits. Bit 2 of CRA enables the Port A latch for an active CA1 transition. Table 2 describes the input handshake modes (CA1, CB1, CA2, CB2) and Table 3 describes the output handshake modes for CA2 and CB2.

3

Register Names:
Port A Data Registers (PDA, P1DA, P2DA)

Register Addresses:
$2 (PDA), $0 (P1DA), $1 (P2DA)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
These three registers serve different purposes. PDA is used to read input data and latch data written to the port A output pins. P1DA and P2DA are used to read input data and to affect handshake and status activity for PC4/CA1 and PC5/CA2. If enabled, port A input data may be latched into the three port A data registers on an active PC4/CA1 transition as described in **HANDSHAKE OPERATION.**

Description:
Data written into PDA is latched into the port A output latch (see Figure 3) regardless of the state of DDRA. Output pins, as defined by DDRA, assume the logic levels of the corresponding bits in the PDA output latch. The PDA output latch allows the user to read the state of the port A output data. If the input latch is not enabled, a read of any port A data register reflects the current state of the port A input pins as defined by DDRA and the contents of the output latch for output pins. Writes into P1DA or P2DA have no effect upon the output pins or the output data latch. Users are recommended to initialize the port A output latch before changing any pin to an output via the DDRA.

MPU accesses of P1DA or P2DA are primarily used to affect handshake and status activity. A summary of the effects on the status and warning bits of port A data register accesses is given in Table 4. For more information, see **HANDSHAKE OPERATION** and **Control Register A (CRA)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port A data register.

Register Names:
Port B Data Registers (PDB, P1DB, P2DB)

Register Addresses:
$3 (PDB), $C(P1DB), $D (P2DB)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
These three registers serve different purposes. The Port B data registers are used to read input data and to latch data written to the port B output pins. Writes to PDB and P1DB affect the contents of the output data latch while writes to P2DB do not affect the output data latch. P1DB and P2DB accesses additionally affect handshake and status activity for PC6/CB1 and PC7/CB2.

Description:
Data written into PDB and P1DB port B registers is latched into the port B output latch (see Figure 3) regardless of the state of DDRB. Output pins, as defined by DDRB, assume the logic levels of the corresponding bits in the port B output latch. Reads of any port B data registers reflect the contents of the output data latch for output pins and the current state of the input pins (as determined by DDRB). Users are recommended to initialize the port B output latch before changing any pin to an output via the DDRB.

MPU accesses of P1DB or P2DB are primarily used to affect handshake and status activity. A summary of the effects on status and warning register bits of port B data register accesses is given in Table 5. For more information, see **HANDSHAKE OPERATION** or **Control Register B (CRB)** under **REGISTER DESCRIPTION.** Reset has no effect upon the contents of any port B data register.

TABLE 4 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT A DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch	
				Read	Write
PDA	None	None	None	Yes	Yes
P1DA	HSA1 cleared to a logic zero.	HWA1 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No
P2DA	HSA2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CA2 goes low if output modes 0 or 1 are selected in the CRA.	Yes	No

TABLE 5 — SUMMARY OF EFFECTS ON HANDSHAKE STATUS, WARNING BITS, AND OUTPUT LATCH BY PORT B DATA REGISTER ACCESSES

Register Accessed	HSR Bit	HWR Bit	Handshake Reaction	Output Latch Read	Output Latch Write
PDB	None	None	None	Yes	Yes
P1DB	HSB1 cleared to a logic zero.	HWB1 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in the CRB.	Yes	Yes
P2DB	HSB2 cleared to a logic zero.	HWA2 loaded into buffer latch.	CB2 goes low if output modes 0 or 1 are selected in CRB.	Yes	No

Register Name:
Port C Data Register (PDC)

Register Address:
$4

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
The port C data register (PDC) is used to read input data and to latch data written to the output pins.

Description:
Data is written into the port C output latch (see Figure 3) regardless of the state of DDRC. Any port C pin defined as a handshake line by the port C function select register (FSR) is not affected by PDC. Output pins, as defined by DDRC, assume logic levels of the corresponding bits in the port C output latch. A read of PDC reflects the contents of the output latch for output pins and the current state of the input pins (as reflected in the DDRC). Reset has no effect upon the contents of PDC. Users are recommended to initialize the port C output data latch before changing any pin to an output via the DDRC.

Register Name:
Data Direction Register for Port A (B) (C)

Register Address:
$6 ($7) ($8)

Register Bits:

7	6	5	4	3	2	1	0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Purpose:
Each of the three data direction registers (DDRA, DDRB, and DDRC) define the direction of data flow of the port pins for ports A, B, and C.

Description:
A logic zero in a DDR bit places the corresponding port pin in the input mode. A logic one in a DDR bit places the corresponding pin in the output mode. Any port C pins defined as bidirectional handshake lines also use the port C DDR (DDRC). Input-only handshake lines are not affected by DDRC. Reset clears all DDR bits to logic zero configuring all port pins as inputs. The DDRs have no write-inhibit control over the port data output latches. Data may be written to the port data registers even though the pins are configured as inputs.

Register Name:
Port C Pin Function Select Register (FSR)

Register Address:
$B

Register Bits:

7	6	5	4	3	2	1	0
CFB2	CFB1	CFA2	CFA1	XX	XX	XX	XX

Purpose:
The port C pin function select register defines whether the multifunction port C pins are to operate as "normal" port C lines or as handshake lines.

Description:
A logic zero in any FSR bit defines the corresponding port C pin as a "normal" I/O pin. A logic one in any valid FSR bit defines the corresponding port C pin as a handshake line. Pins defined as handshake lines function according to the contents of control register A (CRA) or control register B (CRB). The port C data direction register (DDRC) is valid regardless of FSR contents for all pins except PC4/CA1 and PC6/CB1. Transitions on port C pins not defined as handshake pins do not effect the handshake/interrupt status register. Reset clears all FSR bits to a logic zero. Users are recommended to initialize the data direction and control registers before modifying the FSR.

Register Name:
Handshake/Interrupt Status Register (HSR)

Register Address:
\$E

Register Bits:

7	6	5	4	3	2	1	0
IRQF	XX	XX	XX	HSB2	HSA2	HSB1	HSA1

Purpose:
The handshake interrupt status register is a read-only flag register that may be used during a polling routine to determine if any enabled input handshake transition, as defined by the control register (CRA and CRB), has occurred.

3

Description:
If an enabled input handshake transition occurs then the appropriate HSR bit (HSB2, HSA2, HSB1, or HSA1) is set. The IRQ flag bit (bit 7, IRQF) is set when one or more of the HSR bits 0-3 and their corresponding control register bits are set to a logic one as shown in the following equation:

$$\text{Bit } 7 = \text{IRQF} = [\text{HSB2}\bullet\text{CRB2(3)}] + [\text{HSA2}\bullet\text{CRA2(3)}] + [\text{HSB1}\bullet\text{CRB1(0)}] + [\text{HSA1}\bullet\text{CRA1(0)}]$$

The numbers in ( ) indicate which bit in the control register enables the interrupt.

Handshake/interrupt status register bits are cleared by accessing the appropriate port data register. The following table lists the HSR bit and the port data register that must be accessed to clear the bit.

To Clear HSR Bit	Access Register
HSB2	P2DB
HSA2	P2DA
HSB1	P1DB
HSA1	P1DA

Reset clears all handshake/interrupt status register bits to a logic zero.

Register Name:
Handshake Warning Register (HWR)

Register Address:
\$F

Register Bits:

7	6	5	4	3	2	1	0
XX	XX	XX	XX	HWB2	HWA2	HWB1	HWA1

Purpose:
The warning register is a read-only flag register that may be used to determine if a second attempt to set a handshake/interrupt status register bit has been made before the original had been serviced.

Description:
Each bit in the handshake/interrupt status register, except IRQF, has a corresponding bit in the handshake warning register. If an attempt is made to set a bit in the handshake/interrupt status register that is already set, then the corresponding bit in the handshake warning register is also set. An attempt is the occurrence of any enabled input handshake transition as defined by the control registers.

A handshake warning register bit is cleared by first reading the appropriate data register then reading the handshake warning register. Reading the data register (either P1DA, P2DA, P1DB, or P2DB) loads a buffer latch with the proper bit in the handshake warning register (HWA1, HWA2, HWB1, and HWB2, respectively). The next read of the handshake warning register clears the appropriate bit without affecting the other three handshake warning register bits. The upper four bits, HWR4-HWR7, always read as logic zeros. If a port data register is not read before reading the handshake warning register, then the handshake warning register bits will remain unaffected. Reset clears all HWR bits to a logic zero.

Recommended status register handling sequence:

1.	Read status register	(User determines which if any enabled handshake transition occurred)
2.	Read/write port data indicated by status register	(Clears associated status bit and latches appropriate warning register bit in the buffer latch)
3.	Read warning register	(Latched warning bit is cleared and the remaining bits are unaffected)

TYPICAL INTERFACING

The MC146823 is best suited for use with microprocessors which generate an address-then-data-multiplexed bus. Figure 4 shows the MC146823 in a typical CMOS system that uses the MC146805E2 CMOS MPU. Other multiplexed microprocessors can be used as easily.

A single-chip microcomputer (MCU) may be interfaced with 11 port lines as shown in Figure 5. This interface also requires some software overhead to gain up to 13 additional I/O lines and the MC146823 handshake lines.

FIGURE 4 — A TYPICAL CMOS MICROPROCESSOR SYSTEM

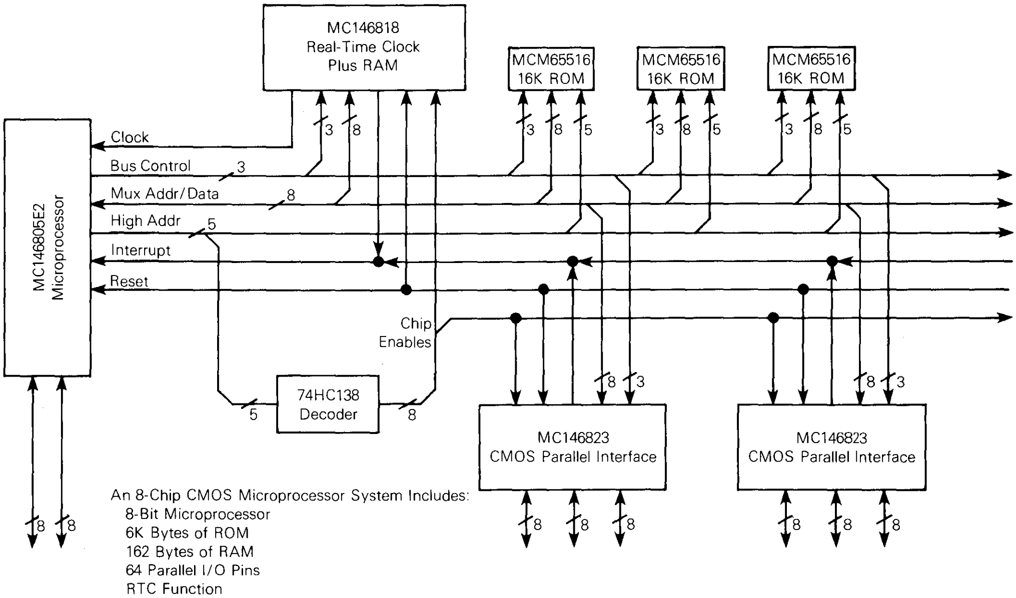
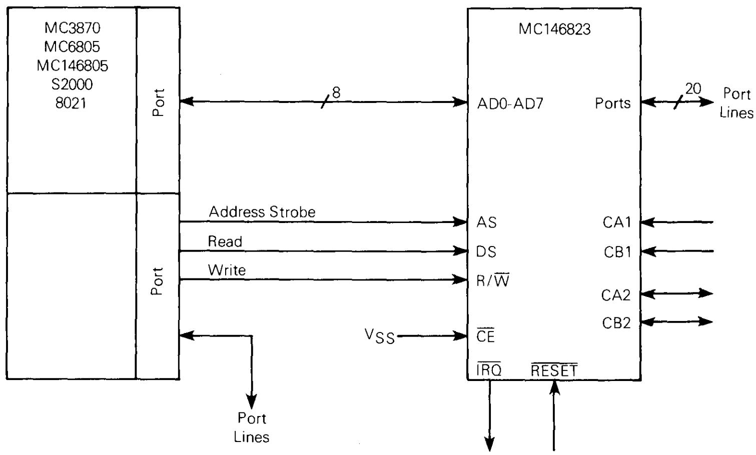


FIGURE 5 — MC146823 INTERFACED WITH THE PORTS OF A TYPICAL SINGLE-CHIP MICROCOMPUTER



3

Mechanical Data

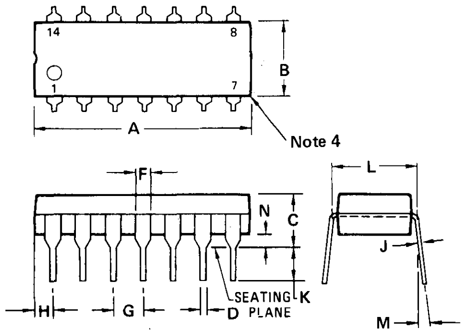
4

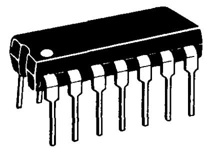
MECHANICAL DATA

The package availability for each device is indicated on the front page of the individual data sheets. Dimensions for the packages are given in this chapter.

14-PIN PACKAGE

PLASTIC PACKAGE
CASE 646-05





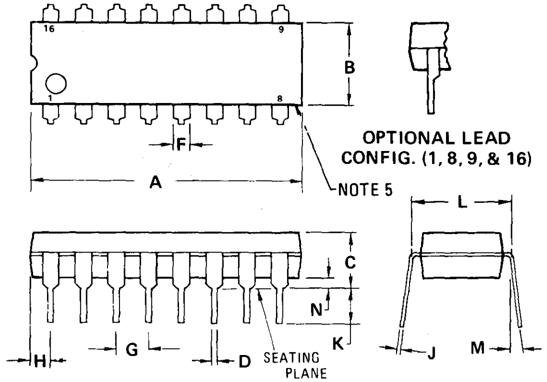
4

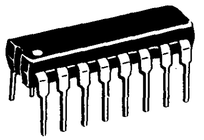
NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.
4. ROUNDED CORNERS OPTIONAL.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	18.16	19.56	0.715	0.770
B	6.10	6.60	0.240	0.260
C	4.06	5.08	0.160	0.200
D	0.38	0.53	0.015	0.021
F	1.02	1.78	0.040	0.070
G	2.54 BSC		0.100 BSC	
H	1.32	2.41	0.052	0.095
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	10°	0°	10°
N	0.51	1.02	0.020	0.040

16-PIN PACKAGE

PLASTIC PACKAGE
CASE 648-05

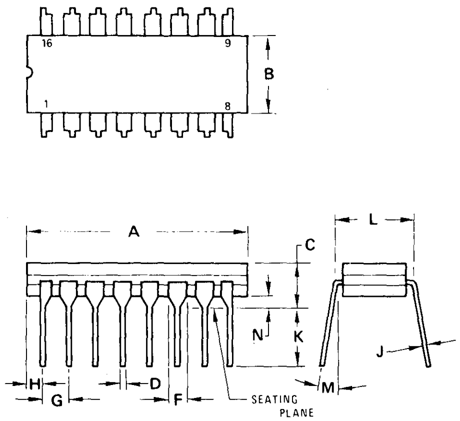


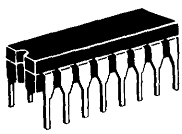


NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.
4. "F" DIMENSION IS FOR FULL LEADS. "HALF" LEADS ARE OPTIONAL AT LEAD POSITIONS 1, 8, 9, and 16).
5. ROUNDED CORNERS OPTIONAL.

	MILLIMETERS		INCHES	
DIM	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

CERAMIC PACKAGE
CASE 620-02





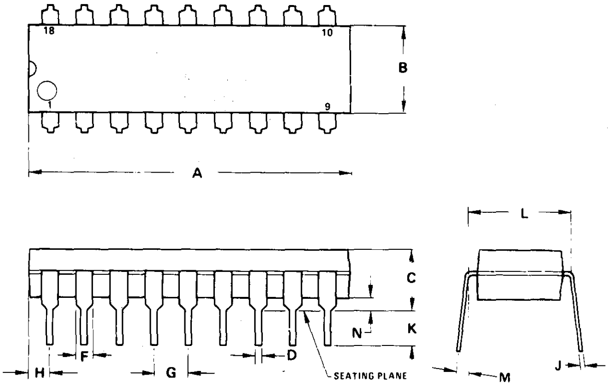
NOTES:
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
2 PKG. INDEX: NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
3 DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

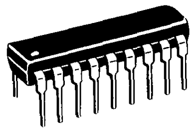
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	19.05	19.81	0.750	0.780
B	6.22	6.98	0.245	0.275
C	4.06	5.08	0.160	0.200
D	0.38	0.51	0.015	0.020
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.14	0.020	0.045
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.37	7.87	0.290	0.310
M	–	15°	–	15°
N	0.51	1.02	0.020	0.040

MECHANICAL DATA (Continued)

18-PIN PACKAGE

PLASTIC PACKAGE
CASE 707-02





DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	22.22	23.24	0.875	0.915
B	6.10	6.60	0.240	0.260
C	3.56	4.57	0.140	0.180
D	0.36	0.56	0.014	0.022
F	1.27	1.78	0.050	0.070
G	2.54 BSC		0.100 BSC	
H	1.02	1.52	0.040	0.060
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

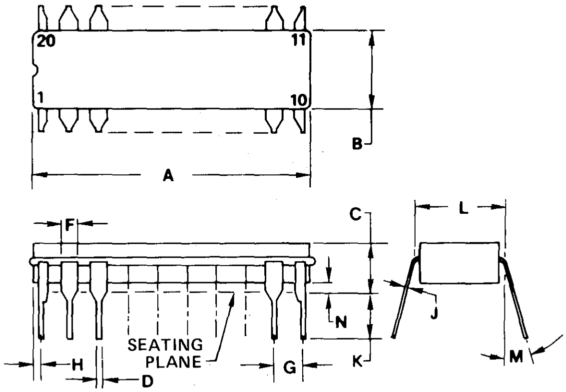
NOTES:

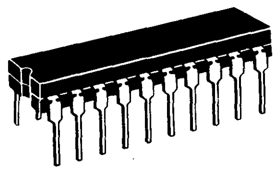
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

MECHANICAL DATA (Continued)

20-PIN PACKAGE

CERAMIC PACKAGE
CASE 732-03





NOTES:
1. LEADS WITHIN 0.25 mm (0.010) DIA, TRUE POSITION AT SEATING PLANE, AT MAXIMUM MATERIAL CONDITION.
2. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIM A AND B INCLUDES MENISCUS.

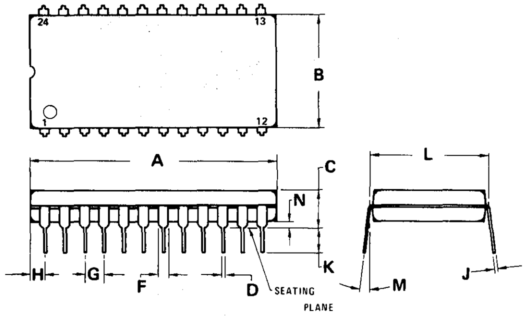
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	23.88	25.15	0.940	0.990
B	6.60	7.49	0.260	0.295
C	3.81	5.08	0.150	0.200
D	0.38	0.56	0.015	0.022
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.27	0.020	0.050
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.25	1.02	0.010	0.040

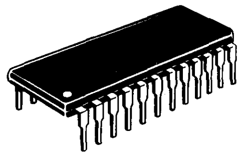
4

MECHANICAL DATA (Continued)

24-PIN PACKAGE

PLASTIC PACKAGE
CASE 709-02

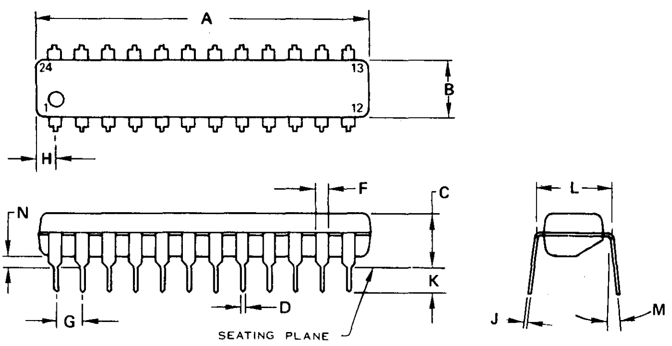


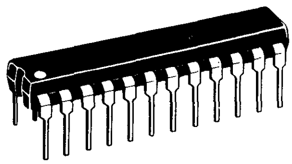


NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	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	1.65	2.03	0.065	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

PLASTIC PACKAGE
CASE 724-02





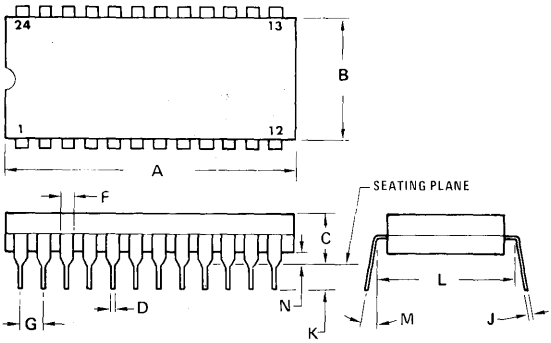
NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION (DIM D).

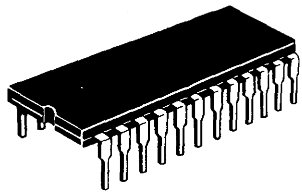
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.13	1.230	1.265
B	6.35	6.86	0.250	0.270
C	4.06	4.57	0.160	0.180
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.60	2.11	0.063	0.083
J	0.18	0.30	0.007	0.012
K	2.92	3.43	0.115	0.135
L	7.37	7.87	0.290	0.310
M	–	10°	–	10°
N	0.51	1.02	0.020	0.040

MECHANICAL DATA (Continued)

24-PIN PACKAGE (Continued)

CERAMIC PACKAGE
CASE 623-05

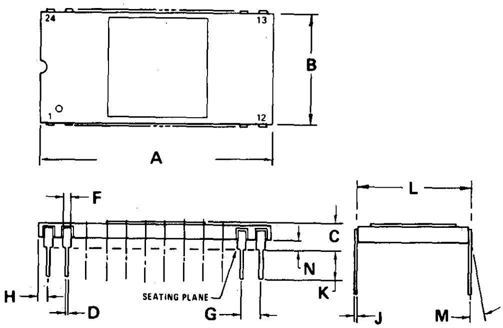


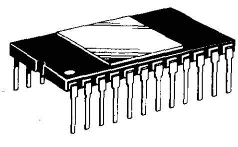


NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.77	1.230	1.290
B	12.70	15.49	0.500	0.610
C	4.06	5.59	0.160	0.220
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

FRIT-SEAL
CERAMIC PACKAGE
CASE 716-06





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.

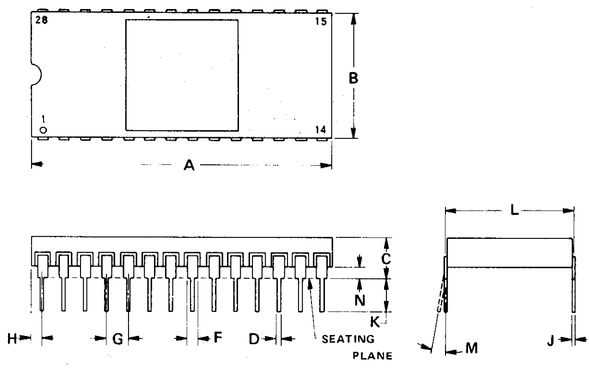
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	27.64	30.99	1.088	1.220
B	14.73	15.34	0.580	0.604
C	2.67	4.32	0.105	0.170
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.30	0.008	0.012
K	2.54	4.57	0.100	0.180
L	14.99	15.49	0.590	0.610
M	–	10°	–	10°
N	1.02	1.52	0.040	0.060

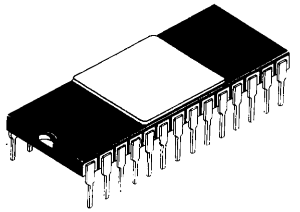
4

MECHANICAL DATA (Continued)

28-PIN PACKAGES

CERAMIC PACKAGE
CASE 719-03

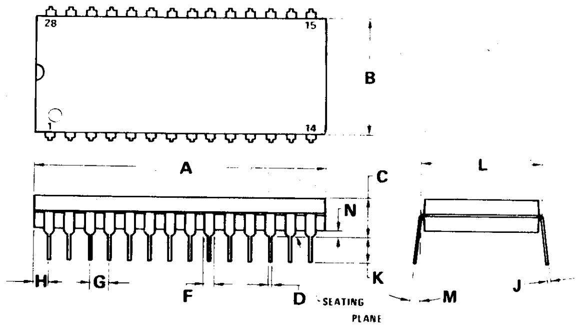


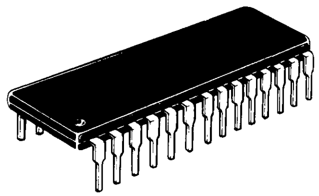


NOTES:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIAMETER (AT SEATING PLANE) AT MAXIMUM MATERIAL CONDITION.
2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

	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 710-02





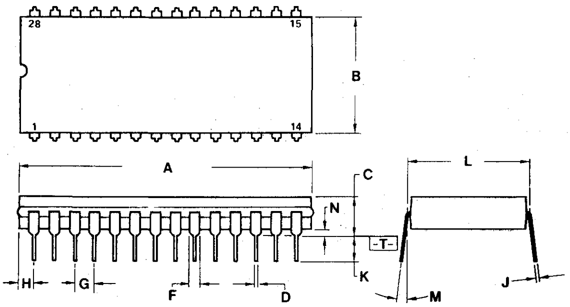
NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm(0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

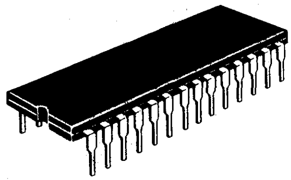
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N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

MECHANICAL DATA (Continued)

28-PIN PACKAGES (Continued)

CERPID PACKAGE
CASE 733-03





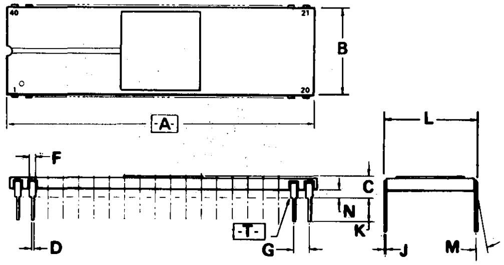
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.85	1.435	1.490
B	12.70	15.37	0.500	0.605
C	4.06	5.84	0.160	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

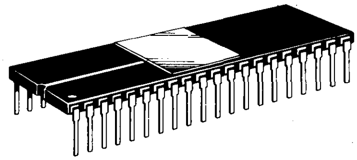
NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOL FOR LEADS:
   ⌖ | Ø 0.25 (0.010) Ⓜ | T | A Ⓜ
3. -T- IS SEATING PLANE.
4. DIM A AND B INCLUDES MENISCUS.
5. DIM -L- TO CENTER OF LEADS WHEN FORMED PARALLEL.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

MECHANICAL DATA (Continued)

40-PIN PACKAGES

CERAMIC PACKAGE
CASE 715-05





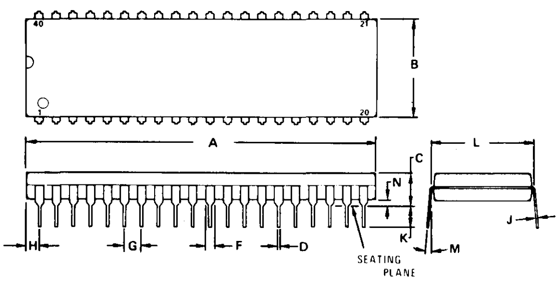
NOTES:
1. DIMENSION -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS:

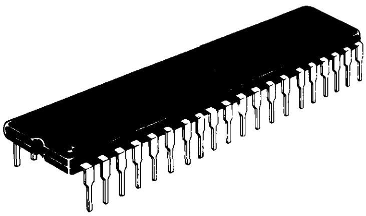
⊕ 0.25 (0.010) Ⓜ T A Ⓜ

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

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

PLASTIC PACKAGE
CASE 711-03





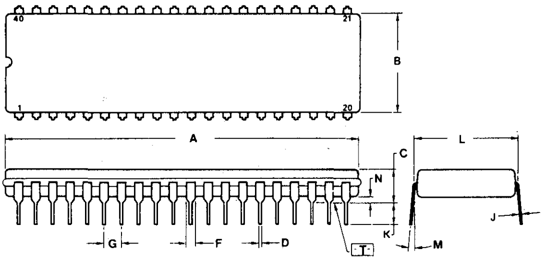
NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

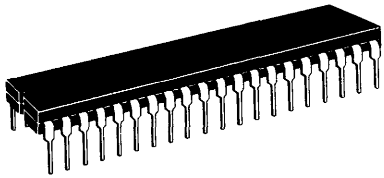
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N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

MECHANICAL DATA (Continued)

40-PIN PACKAGES (Continued)

CERDIP PACKAGE
CASE 734-04



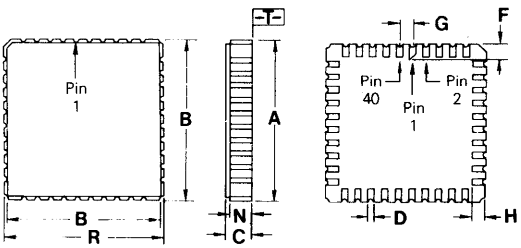


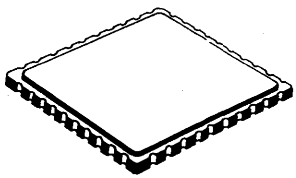
NOTES:
1. DIM -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS: | ⌖ | Ø 0.25(0.010) Ⓜ | T | A Ⓜ |
3. -T- IS SEATING PLANE.
4. DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONS A AND B INCLUDE MENISCUS.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	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^{0}	15^{0}	5^{0}	15^{0}
N	0.51	1.27	0.020	0.050

4

CHIP CARRIER
CASE 761-01





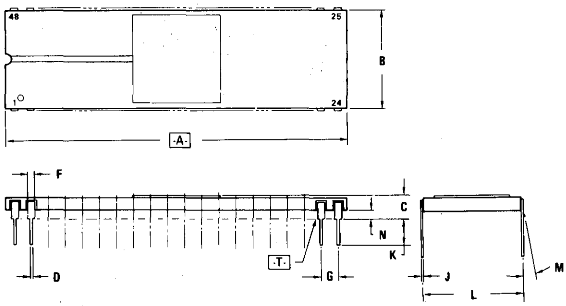
NOTES:
1. DIMENSIONS A & R ARE DATUMS.
2. -T- IS GAUGE PLANE.
3. POSITIONAL TOLERANCE FOR TERMINALS (D): 40 PLACES: | ⌖ | 0.25 (0.010) Ⓜ | T | A Ⓢ | R Ⓢ |
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

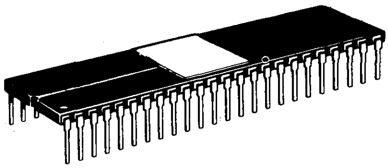
	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	11.94	12.57	0.470	0.495
B	11.05	11.30	0.435	0.445
C	1.60	2.08	0.063	0.082
D	0.33	0.69	0.013	0.027
F	1.07	1.47	0.042	0.058
G	1.02 BSC		0.040 BSC	
H	0.84	1.19	0.033	0.047
N	1.27	1.79	0.050	0.070
R	11.94	12.57	0.470	0.495

MECHANICAL DATA (Continued)

48-PIN PACKAGES

CERAMIC PACKAGE
CASE 740-02





NOTES:
1. DIMENSION -A- IS DATUM.
2. POSTIONAL TOLERANCE FOR LEADS:
   ⊕ Ø 0.25 (0.010) Ⓜ T AⓂ
3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	60.35	61.57	2.376	2.424
B	14.63	15.34	0.576	0.604
C	3.05	4.32	0.120	0.160
D	0.381	0.533	0.015	0.021
F	0.762	1.397	0.030	0.055
G	2.54 BSC		0.100 BSC	
J	0.203	0.330	0.008	0.013
K	2.54	4.19	0.100	0.165
L	14.99	15.65	0.590	0.616
M	0°	10°	0°	10°
N	1.016	1.524	0.040	0.060

Technical Training

5

TECHNICAL TRAINING SYSTEM DESIGN

Since 1974 when Motorola first introduced the M6800 Family course around the United States, Motorola technical training courses have been among the most popular and effective methods for system designers to catch up or keep up with the microprocessor/microcomputer state-of-the-art.

Motorola technical training courses are scheduled throughout the world with courses in the United States, Canada, Mexico, Europe, and Asia. The schedule is advertised periodically, and information is always available from the training headquarters in Phoenix.

A special session of any Motorola technical training course may be held at your facility. This can be a standard course or a course designed to fulfill your particular needs.

The following is a list of course offerings. For more detailed course descriptions, course schedule in your area, or enrollment procedures, write: Motorola Technical Training, P.O. Box 2953, Mail Drop HW-68, Phoenix, Az. 85062. Or Call 602-244-7126, 602-962-2345, or 602-244-4945.

COURSE OFFERINGS

Basic M6800 Family Course — 4 Days (MTT1)

MTT1 is the original course of the M6800 Family, kept up to date and improved during the several years of its existence. It's designed to bring you up to speed in just four days, covering the background you'll need to design, develop, and debug an MC6800-based microcomputer system.

Basic M6801 Course — 4 Days (MTT2)

MTT2 is a beginning course on microprocessors based on the powerful MC6801 hardware and software. It is very similar to Course MTT1, but focuses on the MC6801 rather than the MC6800.

MC6809 Update — 2 Days (MTT3)

Course MTT3 is designed for the student who is very knowledgeable about the MC6800 microprocessor and wants to be equally capable with the MC6809.

High-Level Software — 4 Days (MTT4)

This high-level software course generates a working knowledge of the resident software packages available to users of EXORciser-based MDOS systems.

MC6801 Update — 2 Days (MTT5)

Course MTT5 is designed for the student who is very knowledgeable about the MC6800 microprocessor and wants to be equally capable with the MC6801.

M6805 Introductory Course — 3 Days (MTT6)

MTT6 is an introductory course on Motorola's M6805/M146805 Family of one-chip microcomputers/controllers.

Understanding Microprocessor Basics — 1 Day (MTT7)

This course is a one-day non-technical course designed to acquaint managers, secretaries, buyers, salesmen, and other non-designers with microprocessors. We cover the whys, whats, and hows of microcomputer systems. We'll give you the buzz words and use simplified examples to explain basic concepts. It's a good non-technical course. If you understand terms such as data bus, interrupt, multiplexing, mnemonics, etc., then this course isn't for you.

MC68000 16-Bit Microprocessor — 4 Days (MTT8)

The general features of the MC68000 such as pin functions, registers, addressing modes, and instruction set are covered. In addition, the unique features such as primitive instructions for high-level software, exception handling, and position independent machine code generation are discussed. The development tools used in the course include the Assembler, Editor, and the MC68000 ECB module. Two labs help provide experience with the hardware.

Designing With Micromodules — 2 Days (MTT9)

This 2-day course is designed to develop an understanding of the board-level computer system design approach for potential Micromodule users. The theme of the course is "learning the use of Micromodules through examples."

8-Bit Development Systems — 2 Days (MTT10)

This course is designed to prepare the student to understand and use the basic functions of both MC6800 EXORciser and MC6809 EXORciser II systems.

Basic MC6809 Course — 4 Days (MTT11)

MTT11 is a beginning course on microprocessors based on the powerful MC6809 hardware and software. It is very similar to Course MTT1, but focuses on the MC6809 rather than the MC6800.

Pascal — 4 Days (MTT12)

This course is designed to enable even the novice programmer to write well-constructed programs in Pascal. The first three days are for illustration of standard Pascal and structured programming as taught in a college-level course. The fourth day includes Motorola extensions and implementation for the MC6809 and MC68000. Each student has the opportunity to complete and execute several programs.

EXORmacs — 2 Days (MTT13)

This course aids the student in becoming familiar with EXORmacs. Included are the use of Utilities, Assemblers, Editors/Debuggers, and how to use Pascal on EXORmacs.

MPL — 4 Days (MTT14)

This course is designed to teach the student how to use the MPL Compiler for programming his or her applications. Upon completion of the course, the student will understand the (MC6800 or MC6809) MPL Compiler, the Macro Assembler, the Linking Loader, and MDOS, and will have written and executed programs which use these products.

EXORmacs Operating Systems — 4 Days (MTT15)

This course familiarizes the student with the multi-layered structure and operation of the EXORmacs operating system software. Use of RMS68K and VERSAdos on a target system is also discussed.

Virtual System Course — 4 Days (MTT16)

This course familiarizes the student with the MC68010 and various MC68000 peripheral chips. A vertical system example and design techniques used to implement it are presented.

Basic Macro-Cell Array & CAD Course — 3 Days (MTT17)

MTT17 is an introduction to designing with macro-cell arrays. Basic concepts and trade-offs between current technologies are discussed.

MCA-I CAD Course — 3 Days (MTT17B)

To familiarize the student with the Motorola Computer-Aided Design System used in designing ECL Macrocell arrays. Basic concepts and customer interface are discussed.

MCA-II CAD Course — 3 Days (MTT17C)

To familiarize the student with the Motorola Computer-Aided Design System used in designing CMOS Macrocell arrays. Basic concepts and customer interface are discussed.

MC68000 Operating System (UNIX*-like) — 4 Days (MTT18)

This course teaches the student how to use the Motorola UNIX*-like operating system and the C compiler.

Designing with VERSAmodules/VMEmodules — 4 Days (MTT19)

This course teaches the student about designing with board level products based around the VERSAbus and the VMEbus.

Memory Products

6

MOTOROLA

Memory Selector Guide

Motorola has developed a very broad range of reliable MOS and bipolar memories for virtually any digital data processing system application. And for those whose requirements go beyond individual components, Motorola also supplies Memory Systems and Micromodules.

New Motorola memories are being introduced continually. **This selector guide lists all those available as of November 1983.** For later releases, additional technical information or pricing, contact your nearest authorized Motorola distributor or Motorola sales office.

RAMs

MOS DYNAMIC RAMs

Organization	Part Number	Access Time (ns Max)	Power Supplies	No. of Pins
16384 × 1	MCM4116BP15	150	+12, ±5 V	16
16384 × 1	MCM4116BP20	200	+12, ±5 V	16
16384 × 1	MCM4116BP25	250	+12, ±5 V	16
16384 × 1	MCM4517P10	100	+5 V	16
16384 × 1	MCM4517P12	120	+5 V	16
16384 × 1	MCM4517P15	150	+5 V	16
16384 × 1	MCM4517P20	200	+5 V	16
65536 × 1	MCM6664AP15[1]	150	+5 V	16
65536 × 1	MCM6664AP20[1]	200	+5 V	16
65536 × 1	MCM6665AP15	150	+5 V	16
65536 × 1	MCM6665AP20	200	+5 V	16
65536 × 1	MCM6664BP15[1]*	150	+5 V	16
65536 × 1	MCM6664BP20[1]*	200	+5 V	16
65536 × 1	MCM6665BP15*	150	+5 V	16
65536 × 1	MCM6665BP20*	200	+5 V	16

CMOS STATIC RAMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
2048 × 8	MCM6116P12	120	24
2048 × 8	MCM6116P15	150	24
2048 × 8	MCM6116P20	200	24
4096 × 1	MCM6147P55	55	18
4096 × 1	MCM6147P70	70	18

Operating temperature ranges: 0°C to 70°C

*To be introduced.
(Not all speed selections shown)

[1]Motorola's innovative pin #1 refresh

[2]300 mil package

MOS STATIC RAMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
128 × 8	MCM6810	450	24
128 × 8	MCM68A10	360	24
128 × 8	MCM68B10	250	24
1024 × 4	MCM2114P20	200	18
1024 × 4	MCM2114P25	250	18
1024 × 4	MCM2114P30	300	18
1024 × 4	MCM2114P45	450	18
1024 × 4	MCM21L14P20	200	18
1024 × 4	MCM21L14P25	250	18
1024 × 4	MCM21L14P30	300	18
1024 × 4	MCM21L14P45	450	18
2048 × 8	MCM2016HP45	45	24
2048 × 8	MCM2016HN45	45	24[2]
2048 × 8	MCM2016HY45	45	24[2]
2048 × 8	MCM2016HP55	55	24
2048 × 8	MCM2016HN55	55	24[2]
2048 × 8	MCM2016HY55	55	24[2]
2048 × 8	MCM2016HP70	70	24
2048 × 8	MCM2016HN70	70	24[2]
2048 × 8	MCM2016HY70	70	24[2]
16384 × 1	MCM2167HP35	35	20
16384 × 1	MCM2167HL35	35	20
16384 × 1	MCM2167HZ35	35	20
16384 × 1	MCM2167HP45	45	20
16384 × 1	MCM2167HL45	45	20
16384 × 1	MCM2167HZ45	45	20
16384 × 1	MCM2167HP70	70	20
16384 × 1	MCM2167HL70	70	20
16384 × 1	MCM2167HZ70	70	20

EPROMs

MOS EPROMs

Organization	Part Number	Access Time (ns max)	Power Supplies	No. of Pins
8192 × 8	MCM68764C	450	+5 V	24
8192 × 8	MCM68766C	450	+5 V	24
8192 × 8	MCM68766C35	350	+5 V	24

6

ROMs

MOS STATIC ROMs (+5 Volts)

Character Generators[3]

Organization	Part Number	Access Time (ns max)	No. of Pins
128 × (7 × 5)	MCM6670P	350	18
128 × (7 × 5)	MCM6674P	350	18
128 × (9 × 7)	MCM66700P	350	24
128 × (9 × 7)	MCM66710P	350	24
128 × (9 × 7)	MCM66714P	350	24
128 × (9 × 7)	MCM66720P	350	24
128 × (9 × 7)	MCM66730P	350	24
128 × (9 × 7)	MCM66734P	350	24
128 × (9 × 7)	MCM66740P	350	24
128 × (9 × 7)	MCM66750P	350	24
128 × (9 × 7)	MCM66760P	350	24
128 × (9 × 7)	MCM66770P	350	24
128 × (9 × 7)	MCM66780P	350	24
128 × (9 × 7)	MCM66790P	350	24

MOS Binary ROMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
2048 × 8	MCM68A316EP	350	24
2048 × 8	MCM68A316EP91[4]	350	24
4096 × 8	MCM68A332P	350	24
4096 × 8	MCM68A332P2[4]	350	24
8192 × 8	MCM68364P35	350	24
8192 × 8	MCM68364P35-3[4]	350	24
8192 × 8	MCM68364P25	250	24
8192 × 8	MCM68364P20	200	24
8192 × 8	MCM68365P25	250	24
8192 × 8	MCM68365P35	350	24
8192 × 8	MCM68366P25	250	24
8192 × 8	MCM68366P35	350	24
16384 × 8	MCM63128P15	150	28
16384 × 8	MCM63128P20	200	28
32768 × 8	MCM63256P15	150	28
32768 × 8	MCM63256P20	200	28

CMOS ROMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
256 × 4	MCM14524	1200	16
2048 × 8	MCM65516P43	430	18
2048 × 8	MCM65516P43M[8]	430	18
2048 × 8	MCM65516P55	550	18

Operating temperature ranges: 0°C to 70°C

*To be introduced.
(Not all speed selections shown)

[3]Character generators include shifted and unshifted characters, ASCII alphanumeric control, math, Japanese British, German, European and French symbols.

[4]Standard Patterns for MOS ROMs:
MCM68A316EP91 – Universal Code Converter and Character Generator
MCM68A332P2 – Sine/Cosine Look-Up Table
MCM68364P35-3 – Log/Antilog Look-Up Table
MCM65516P43M – MC146805 Monitor Program

6

Logic and Special Function Products

7

7

BUFFERS/INVERTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC04	Hex Inverter	LS04	*4069	LS/CMOS	14
HCT04	Hex Inverter with LSTTL-Compatible Inputs	LS04	*4069	LS/CMOS	14
HCU04	Hex Unbuffered Inverter	*LS04	4069	LS/CMOS	14
HC14	Hex Schmitt-Trigger Inverter	LS14	4584	LS/CMOS	14
HC125	Quad 3-State Noninverting Buffer	LS125		LS	14
HC126	Quad 3-State Noninverting Buffer	LS126		LS	14
HC240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver	LS240		LS	20
HCT240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS240		LS	20
HC241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS241		LS	20
HCT241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS241		LS	20
HC242	Quad 3-State Inverting Bus Transceiver	LS242		LS	14
HC243	Quad 3-State Noninverting Bus Transceiver	LS243		LS	14
HC244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS244		LS	20
HCT244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS244		LS	20
HC245	Octal 3-State Noninverting Bus Transceiver	LS245		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HC365	Hex 3-State Noninverting Buffer with Common Enables	LS365A		LS	16
HC366	Hex 3-State Inverting Buffer with Common Enables	LS366A		LS	16
HC367	Hex 3-State Noninverting Buffer with Separate 2-Bit and 4-Bit Sections	LS367A	*4503	LS/CMOS	16
HC368	Hex 3-State Inverting Buffer with Separate 2-Bit and 4-Bit Sections	LS368A		LS	16
HC540	Octal 3-State Inverting Buffer/Line Driver/Line Receiver	LS540		LS	20
HC541	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver	LS541		LS	20
HC640	Octal 3-State Inverting Bus Transceiver	LS640		LS	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HC643	Octal 3-State Inverting and Noninverting Bus Transceiver	LS643		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HC4049	Hex Inverting Buffer/Logic-Level Down Converter		4049	CMOS	16
HC4050	Hex Noninverting Buffer/Logic-Level Down Converter		4050	CMOS	16

* Suggested alternative

BUFFERS/INVERTERS

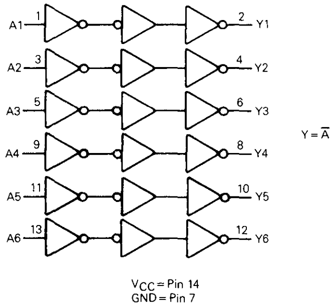
BUFFERS

Device	HC 04	HCT 04	HCU 04	HC 14	HC 125	HC 126	HC 240	HCT 240	HC 241	HCT 241	HC 242	HC 243	HC 244	HCT 244
# Pins	14	14	14	14	14	14	20	20	20	20	14	14	20	20
Quad Device					•	•					•	•		
Hex Device	•	•	•	•										
Octal Device							•	•	•	•			•	•
Noninverting Outputs					•	•			•	•		•	•	•
Inverting Outputs	•	•	•	•			•	•			•			
Single Stage (unbuffered)			•											
Schmitt Trigger				•										
3-State Outputs					•	•	•	•	•	•	•	•	•	•
Common Output Enables								•	•	•	•	•	•	•
Active-Low Output Enables					•		••	••	•	•	•	•	••	••
Active-High Output Enables						•			•	•	•	•		
Separate 4-Bit Sections							•	•	•	•			•	•
Separate 2-Bit and 4-Bit Sections														
Transceiver											•	•		
Direction Control														
Logic-Level Down Converter														
LSTTL-Compatible Inputs		•						•		•				•

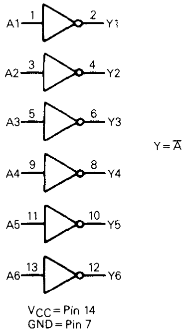
Device	HC 245	HCT 245	HC 365	HC 366	HC 367	HC 368	HC 540	HC 541	HC 640	HCT 640	HC 643	HCT 643	HC 4049	HC 4050
# Pins	20	20	16	16	16	16	20	20	20	20	20	20	16	16
Quad Device														
Hex Device			•	•	•	•							•	•
Octal Device	•	•					•	•	•	•	•	•		
Noninverting Outputs	•	•	•		•			•			•	•		•
Inverting Outputs				•		•	•		•	•	•	•	•	
Single Stage (unbuffered)														
Schmitt Trigger														
3-State Outputs	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Common Output Enables	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Active-Low Output Enables	•	•	••	••	••	••	••	••	•	•	•	•		
Active-High Output Enables														
Separate 4-Bit Sections														
Separate 2-Bit and 4-Bit Sections					•	•								
Transceiver	•	•							•	•	•	•		
Direction Control	•	•							•	•	•	•		
Logic-Level Down Converter													•	•
LSTTL-Compatible Inputs		•								•		•		

BUFFERS/INVERTERS

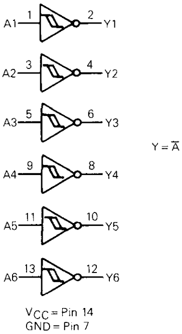
HC04
HCT04



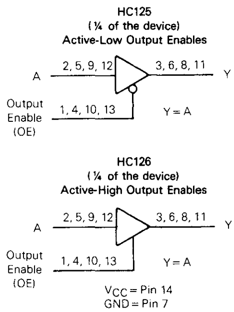
HCU04



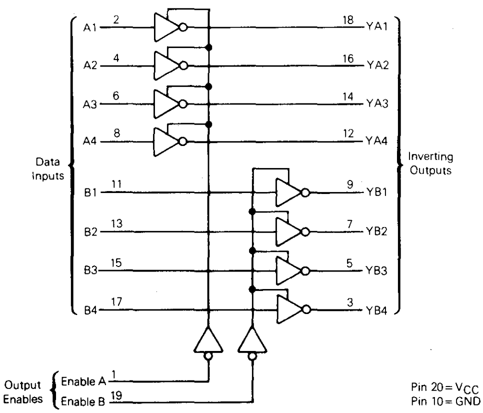
HC14



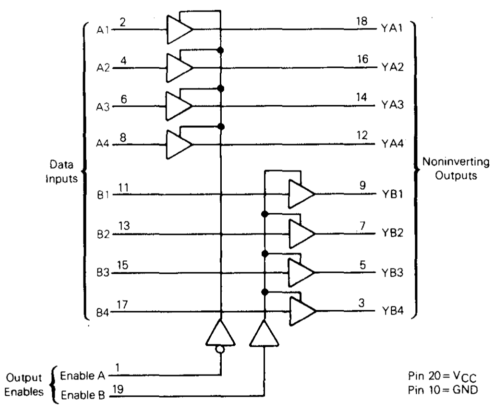
HC125
HC126



HC240
HCT240

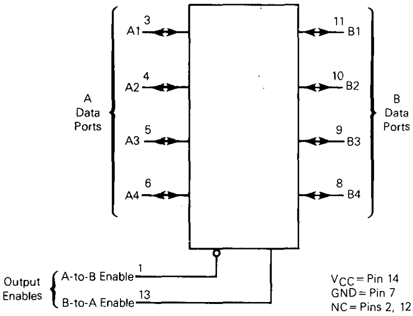


HC241
HCT241

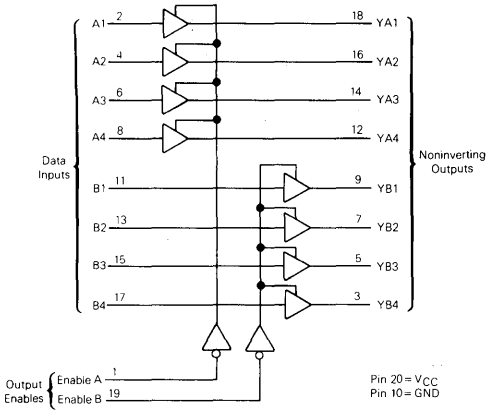


7

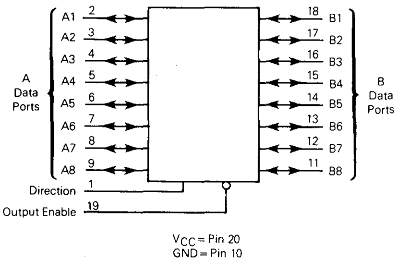
HC242
HC243



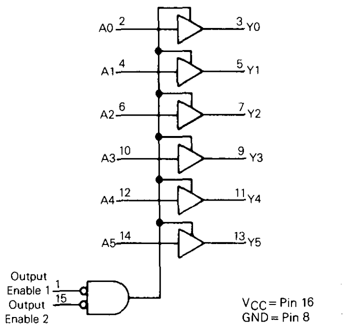
HC244
HCT244



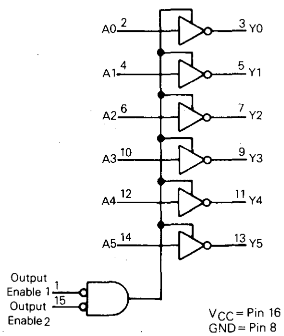
HC245
HCT245



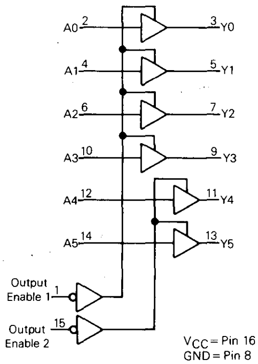
HC365



HC366

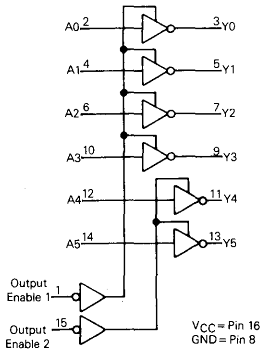


HC367

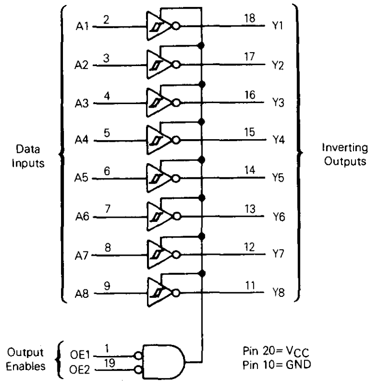


BUFFERS/INVERTERS

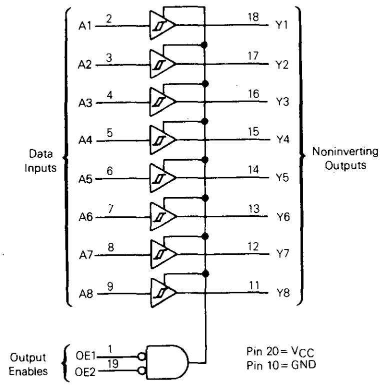
HC368



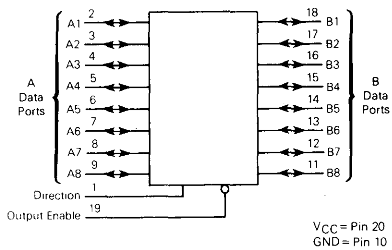
HC540



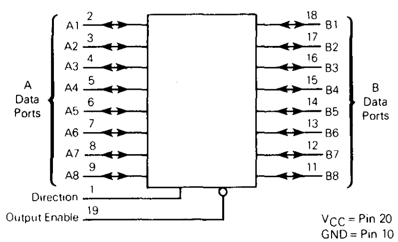
HC541



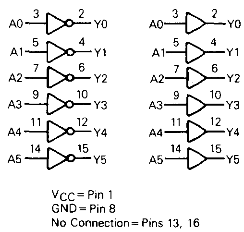
HC640
HCT640



HC643
HCT643



HC4049
HC4050



7

GATES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC00	Quad 2-Input NAND Gate	LS00	4011	LS	14
HCT00	Quad-2 Input NAND Gate with LSTTL-Compatible Inputs	LS00	4011	LS	14
HC02	Quad 2-Input NOR Gate	LS02	4001	LS	14
HC03	Quad 2-Input NAND Gate with Open-Drain Outputs	LS03	*4011	LS	14
HC08	Quad 2-Input AND Gate	LS08	4081	LS	14
HC10	Triple 3-Input NAND Gate	LS10	4023	LS	14
HC11	Triple 3-Input AND Gate	LS11	4073	LS	14
HC20	Dual 4-Input NAND Gate	LS20	4012	LS	14
HC27	Triple 3-Input NOR Gate	LS27	4025	LS	14
HC30	8-Input NAND Gate	LS30	4068	LS	14
HC32	Quad 2-Input OR Gate	LS32	4071	LS	14
HC51	2-Wide, 2-Input/2-Wide, 3-Input AND-OR-INVERT Gates	LS51	*4506	LS	14
☆HC58	2-Wide, 2-Input/2-Wide, 3-Input AND-OR Gates	*LS51	*4506		14
HC86	Quad 2-Input Exclusive OR Gate	LS86	4070	LS	14
HC132	Quad 2-Input Schmitt-Trigger NAND Gate	LS132	4093	LS	14
HC133	13-Input NAND Gate	LS133		LS	16
HC266	Quad 2-Input Exclusive NOR Gate	*LS266	4077	LS/CMOS	14
HC4002	Dual 4-Input NOR Gate	*LS25	4002	CMOS	14
HC4075	Triple 3-Input OR Gate		4075	CMOS	14
HC4078	8-Input NOR/OR Gate		4078	CMOS	14

* Suggested alternative

☆ High-Speed CMOS design only

7

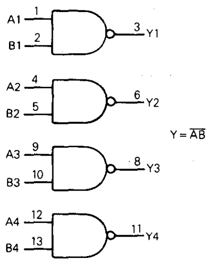
GATES

Device	HC 00	HCT 00	HC 02	HC 03	HC 08	HC 10	HC 11	HC 20	HC 27	HC 30
#Pins	14	14	14	14	14	14	14	14	14	14
Single Device										•
Dual Device								•		
Triple Device						•	•		•	
Quad Device	•	•	•	•	•					
NAND	•	•		•		•		•		•
NOR			•						•	
AND					•		•			
OR										
Exclusive OR										
Exclusive NOR										
AND-OR-INVERT										
AND-OR										
2-Input	•	•	•	•	•					
3-Input						•	•		•	
4-Input								•		
8-Input										•
13-Input										
Schmitt Trigger Inputs										
LSTTL-Compatible Inputs			•							
Open-Drain Outputs				•						

Device	HC 32	HC 51	HC 58	HC 86	HC 132	HC 133	HC 266	HC 4002	HC 4075	HC 4078
#Pins	14	14	14	14	14	16	14	14	14	14
Single Device		•	•			•				•
Dual Device								•		
Triple Device									•	
Quad Device	•			•	•		•			
NAND					•	•				
NOR								•		•
AND										
OR	•								•	
Exclusive OR				•						
Exclusive NOR							•			
AND-OR-INVERT		•								
AND-OR			•							
2-Input	•	•	•	•	•		•			
3-Input		•	•						•	
4-Input								•		
8-Input										•
13-Input						•				
Schmitt Trigger Inputs					•					
LSTTL-Compatible Inputs										
Open-Drain Outputs										

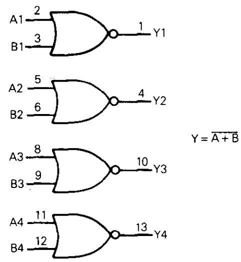
GATES

HC00
HCT00



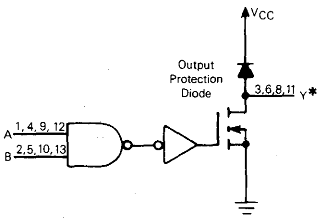
V_{CC} = Pin 14
GND = Pin 7

HC02



V_{CC} = Pin 14
GND = Pin 7

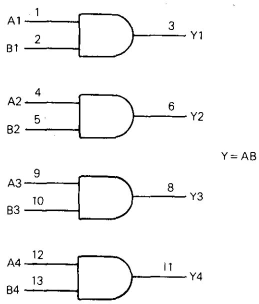
HC03



V_{CC} = Pin 14
GND = Pin 7

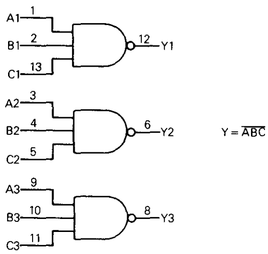
*Denotes open-drain outputs.

HC08

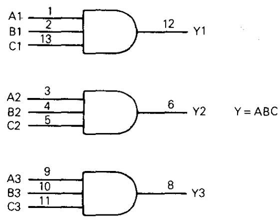


V_{CC} = Pin 14
GND = Pin 7

HC10



HC11

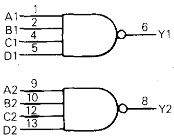


V_{CC} = Pin 14
GND = Pin 7

7

GATES

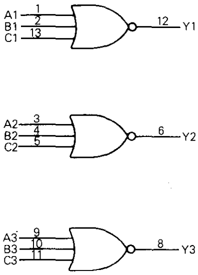
HC20



$Y = \overline{ABCD}$

V_{CC} = Pin 14
GND = Pin 7
NC = Pins 3, 11

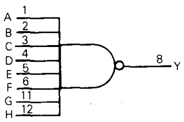
HC27



$Y = \overline{A + B + C}$

V_{CC} = Pin 14
GND = Pin 7

HC30

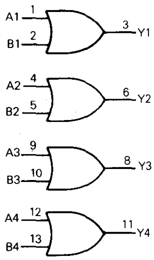


$Y = \overline{ABCDEFGH}$

Pins 9, 10, 13 = no connection

V_{CC} = Pin 14
GND = Pin 7

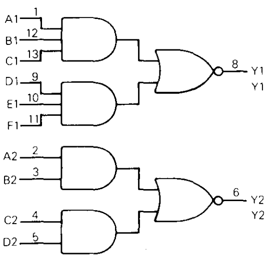
HC32



$Y = A + B$

V_{CC} = Pin 14
GND = Pin 7

HC51

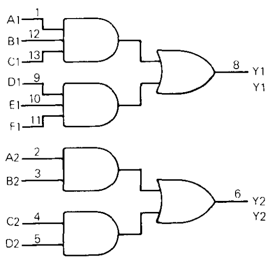


$Y1 = \overline{(A1 \bullet B1 \bullet C1) + (D1 \bullet E1 \bullet F1)}$

$Y2 = \overline{(A2 \bullet B2) + (C2 \bullet D2)}$

V_{CC} = Pin 14
GND = Pin 7

HC58



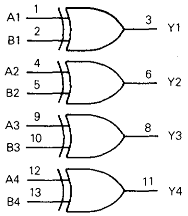
$Y1 = (A1 \bullet B1 \bullet C1) + (D1 \bullet E1 \bullet F1)$

$Y2 = (A2 \bullet B2) + (C2 \bullet D2)$

V_{CC} = Pin 14
GND = Pin 7

7

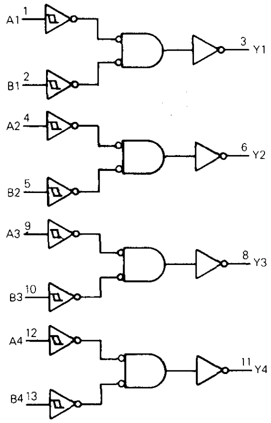
HC86



$Y = A \oplus B$
$= \overline{A}B + A\overline{B}$

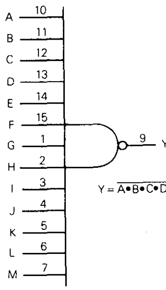
V_{CC} = Pin 14
GND = Pin 7

HC132



$Y = \overline{AB}$

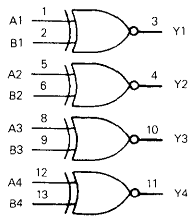
HC133



$Y = \overline{A \bullet B \bullet C \bullet D \bullet E \bullet F \bullet G \bullet H \bullet I \bullet J \bullet K \bullet L \bullet M}$

V_{CC} = Pin 16
GND = Pin 8

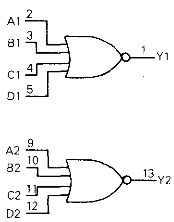
HC266



$Y = \overline{A \oplus B}$
$= AB + \overline{A}\overline{B}$

V_{CC} = Pin 14
GND = Pin 7

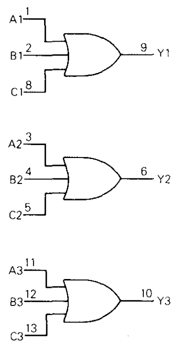
HC4002



$Y = \overline{A + B + C + D}$

V_{CC} = Pin 14
GND = Pin 7
No Connection = Pins 6, 8

HC4075

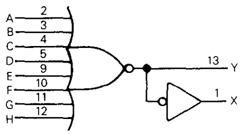


$Y = A + B + C$

V_{CC} = Pin 14
GND = Pin 7

7

HC4078



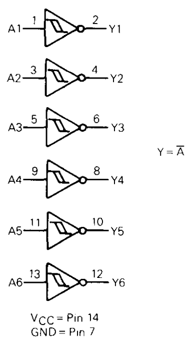
$Y = \overline{A+B+C+D+E+F+G+H}$
$X = A+B+C+D+E+F+G+H$

V_{CC} = Pin 14
GND = Pin 7
No Connection = Pins 6, 8

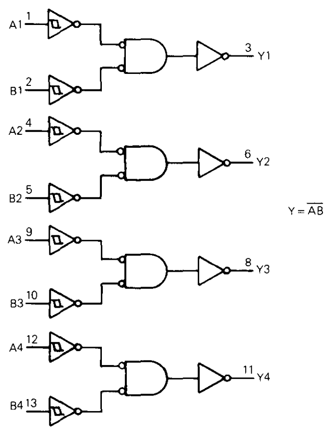
SCHMITT TRIGGERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC14	Hex Schmitt-Trigger Inverter	LS14	4584	LS/CMOS	14
HC132	Quad 2-Input Schmitt-Trigger NAND Gate	LS132	4093	LS	14

HC14



HC132



BUS TRANSCEIVERS

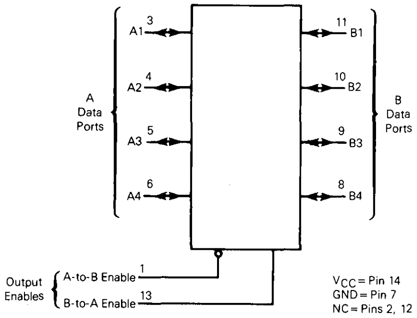
Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC242	Quad 3-State Inverting Bus Transceiver	LS242		LS	14
HC243	Quad 3-State Noninverting Bus Transceiver	LS243		LS	14
HC245	Octal 3-State Noninverting Bus Transceiver	LS245		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HC640	Octal 3-State Inverting Bus Transceiver	LS640		LS	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HC643	Octal 3-State Inverting and Noninverting Bus Transceiver	LS643		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HC646	Octal 3-State Noninverting Bus Transceiver and D-Type Flip-Flop	LS646		LS	24
HC648	Octal 3-State Inverting Bus Transceiver and D-Type Flip-Flop	LS648		LS	24

Device	HC 242	HC 243	HC 245	HCT 245	HC 640	HCT 640	HC 643	HCT 643	HC 646	HC 648
#Pins	14	14	20	20	20	20	20	20	24	24
Quad Device	•	•								
Octal Device			•	•	•	•	•	•	•	•
Buffer	•	•	•	•	•	•	•	•		
Storage Capability									•	•
Inverting Output	•				•	•	•	•	•	
Noninverting Output		•	•	•			•	•		•
Common Output Enables	•	•	•	•	•	•	•	•	•	•
Active-Low Output Enable	•	•	•	•	•	•	•	•	•	•
Active-High Output Enable	•	•								
Direction Control			•	•	•	•	•	•	•	•
LSTTL-Compatible Inputs				•		•		•		

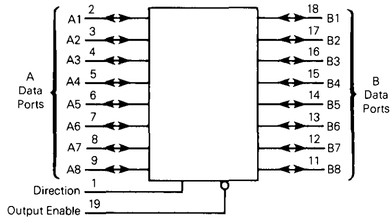
7

HC242
HC243

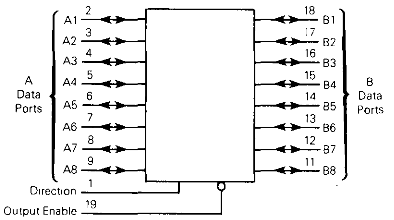
HC242—Inverting Outputs
HC243—Noninverting Outputs



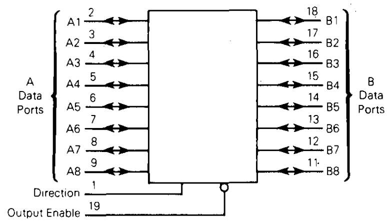
HC245
HCT245



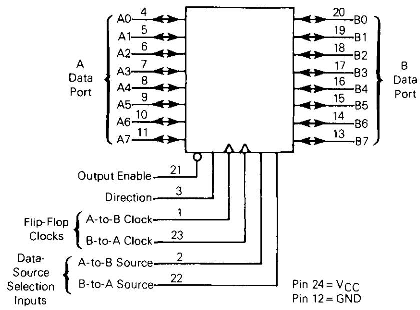
HC640
HCT640



HC643
HCT643



HC646
HC648



LATCHES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC75	Dual 2-Bit Transparent Latch	LS75	*4042	LS	16
HC259	8-Bit Addressable Latch/1-of-8 Decoder	LS259	*4099	LS	16
HC373	Octal 3-State Noninverting D-Type Transparent Latch	LS373, LS573		LS373	20
HCT373	Octal 3-State Noninverting D-Type Transparent Latch with LSTTL-Compatible Inputs	LS373, LS573		LS373	20
HC533	Octal 3-State Inverting D-Type Transparent Latch	LS533		LS	20
HC563	Octal 3-State Inverting D-Type Transparent Latch	LS533			20
HC573	Octal 3-State Noninverting D-Type Transparent Latch	LS373, LS573		LS573	20

*Suggested alternative

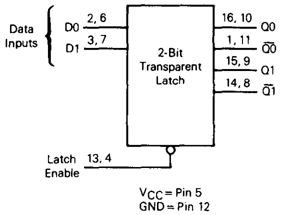
Device	HC 75	HC 259	HC 373	HCT 373	HC 533	HC 563	HC 573
#Pins	16	16	20	20	20	20	20
Single Device		•					
Dual Device	•						
Octal Device			•	•	•	•	•
1-Bit			•	•	•	•	•
2-Bit	•						
8-Bit		•					
Transparent	•	•	•	•	•	•	•
Addressable		•					
Noninverting Outputs	•	•	•	•			•
Inverting Outputs	•				•	•	
Common Latch Enable			•	•	•	•	•
Active-Low Latch Enable	•	•	•	•	•	•	•
Active-Low Reset		•					
3-State Outputs			•	•	•	•	•
Common Output Enable; Active-Low			•	•	•	•	•
LSTTL-Compatible Inputs				•			

These devices are identical in function and are different in pinout only: HC373 and HC573
HC533 and HC563

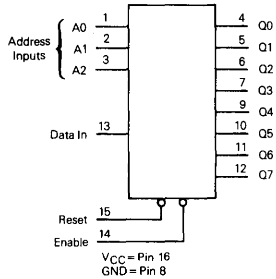
7

LATCHES

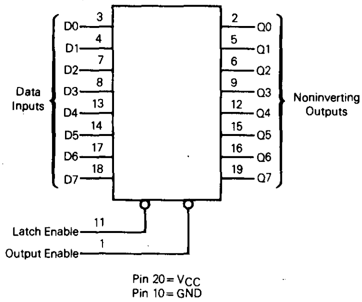
HC75



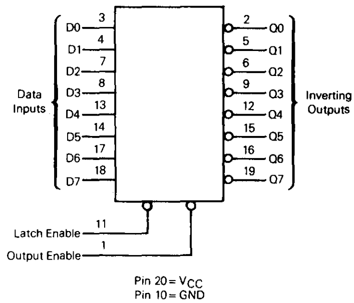
HC259



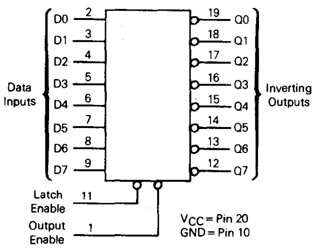
HC373
HCT373



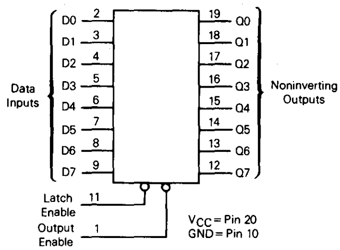
HC533



HC563



HC573



FLIP-FLOPS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC73	Dual J-K Flip-Flop with Reset	LS73A, LS107A	*4027	LS73A	14
HC74	Dual D-Type Flip-Flop with Set and Reset	LS74A	4013	LS	14
HC76	Dual J-K Flip-Flop with Set and Reset	LS76A, LS112A	*4027	LS76A	16
HC107	Dual J-K Flip-Flop with Reset	LS73A, LS107A	*4027	LS107A	14
HC109	Dual J-$\overline{K}$ Flip-Flop with Set and Reset	LS109A	*4027	LS	16
HC112	Dual J-K Flip-Flop with Set and Reset	LS76A, LS112A	*4027	LS112A	16
HC113	Dual J-K Flip-Flop with Set	LS113A	*4027	LS	14
HC173	Quad 3-State D-Type Flip-Flop with Common Clock and Reset	LS173A	4076	LS/CMOS	16
HC174	Hex D-Type Flip-Flop with Common Clock and Reset	LS174	4174	LS/CMOS	16
HC175	Quad D-Type Flip-Flop with Common Clock and Reset	LS175	4175	LS/CMOS	16
HC273	Octal D-Type Flip-Flop with Common Clock and Reset	LS273		LS	20
HC374	Octal 3-State Noninverting D-Type Flip-Flop	LS374, LS574		LS374	20
HCT374	Octal 3-State Noninverting D-Type Flip-Flop with LSTTL-Compatible Inputs	LS374, LS574		LS374	20
HC534	Octal 3-State Inverting D-Type Flip-Flop	LS534		LS	20
HC564	Octal 3-State Inverting D-Type Flip-Flop	LS534			20
HC574	Octal 3-State Noninverting D-Type Flip-Flop	LS374, LS574		LS574	20
HC646	Octal 3-State Noninverting Bus Transceiver and D-Type Flip-Flop	LS646		LS	24
HC648	Octal 3-State Inverting Bus Transceiver and D-Type Flip-Flop	LS648		LS	24

*Suggested alternative

7

FLIP-FLOPS

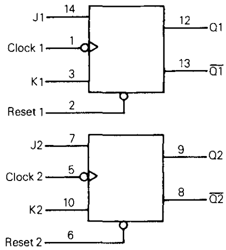
Device	HC 73	HC 74	HC 76	HC 107	HC 109	HC 112	HC 113	HC 173	HC 174
#Pins	14	14	16	14	16	16	14	16	16
Type	J-K	D	J-K	J-K	J-$\overline{\text{K}}$	J-K	J-K	D	D
Dual Device	•	•	•	•	•	•	•		
Quad Device								•	
Hex Device									•
Octal Device									
Common Clock								•	•
Negative-Transition Clocking	•		•	•		•	•		
Postive-Transition Clocking		•			•			•	•
Common, Active-Low Data Enables								••	
Noninverting Outputs	•	•	•	•	•	•	•	•	•
Inverting Outputs	•	•	•	•	•	•	•		
3-State Outputs								•	
Common, Active-Low Output Enables								••	
Common Reset								•	•
Active-Low Reset	•	•	•	•	•	•			•
Active-High Reset								•	
Active-Low Set		•	•		•	•	•		
Transceiver									
Direction Control									
LSTTL-Compatible Inputs									

Device	HC 175	HC 273	HC 374	HCT 374	HC 534	HC 564	HC 574	HC 646	HC 648
#Pins	16	20	20	20	20	20	20	24	24
Type	D	D	D	D	D	D	D	D	D
Dual Device									
Quad Device	•								
Hex Device									
Octal Device		•	•	•	•	•	•	•	•
Common Clock	•	•	•	•	•	•	•	•	•
Negative-Transition Clocking									
Positive-Transition Clocking	•	•	•	•	•	•	•	•	•
Common, Active-Low Data Enables									
Noninverting Outputs	•	•	•	•			•	•	
Inverting Outputs	•				•	•			•
3-State Outputs			•	•	•	•	•	•	•
Common, Active-Low Output Enables			•	•	•	•	•	•	•
Common Reset	•	•							
Active-Low Reset	•	•							
Active-High Reset									
Active-Low Set									
Transceiver								•	•
Direction Control								•	•
LSTTL-Compatible Inputs				•					

These devices are identical in function and are different in pinout only: HC73 and HC107
HC76 and HC112
HC374 and HC574
HC534 and HC564

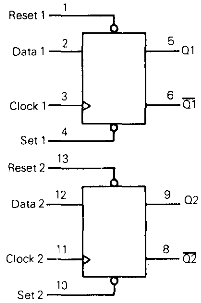
FLIP-FLOPS

HC73



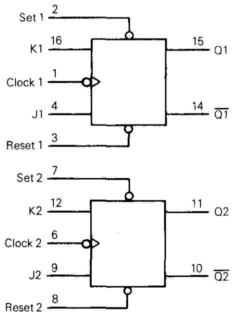
V_{CC} = Pin 4
GND = Pin 11

HC74



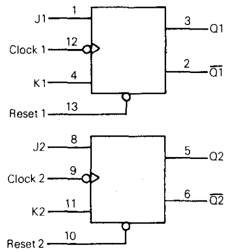
V_{CC} = Pin 14
GND = Pin 7

HC76



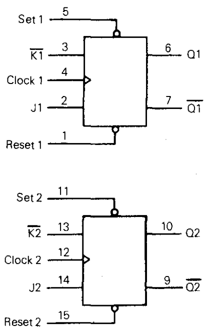
V_{CC} = Pin 5
GND = Pin 13

HC107



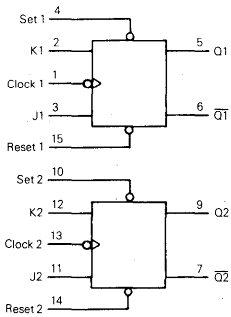
V_{CC} = Pin 14
GND = Pin 7

HC109



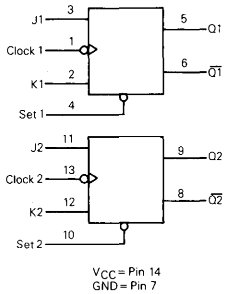
V_{CC} = Pin 16
GND = Pin 8

HC112

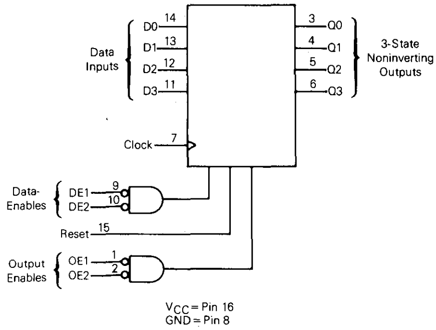


V_{CC} = Pin 16
GND = Pin 8

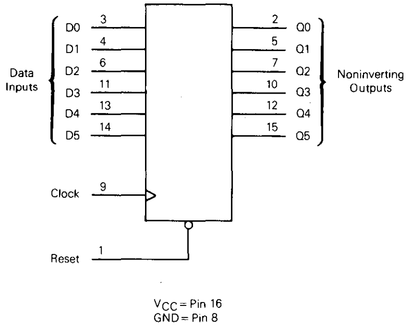
HC113



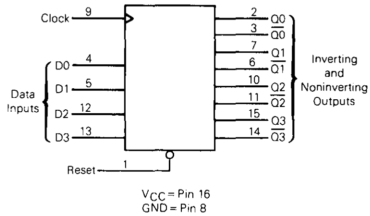
HC173



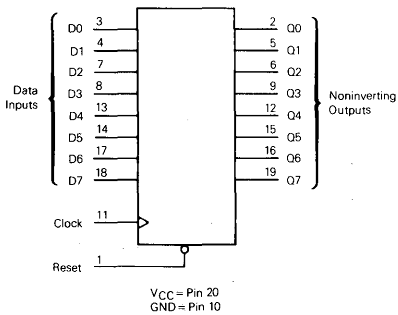
HC174



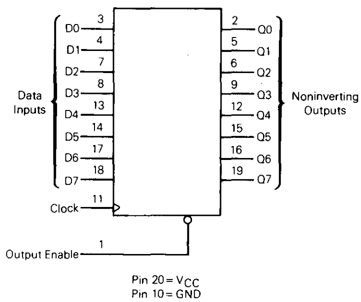
HC175



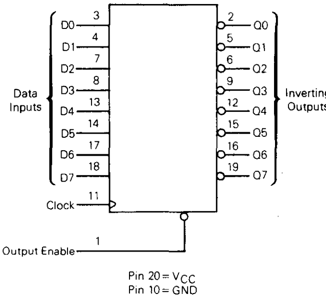
HC273



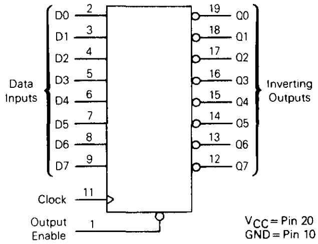
HC374
HCT374



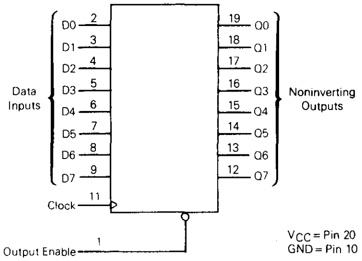
HC534



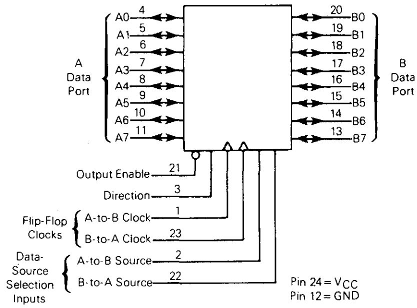
HC564



HC574



HC646
HC648



7

DIGITAL DATA SELECTORS/MULTIPLEXERS

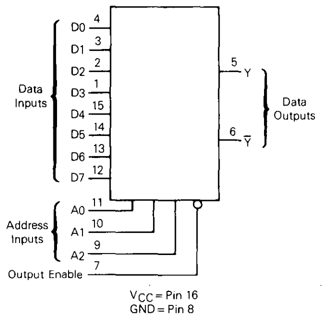
Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC151	8-Input Data Selector/Multiplexer	LS151	*4512	LS	16
HC153	Dual 4-Input Data Selector/Multiplexer	LS153	4539	LS/CMOS	16
HC157	Quad 2-Input Noninverting Data Selector/Multiplexer	LS157	*4519	LS	16
HC158	Quad 2-Input Inverting Data Selector/Multiplexer	LS158	*4519	LS	16
HC251	8-Input Data Selector/Multiplexer with 3-State Outputs	LS251	*4512	LS	16
HC253	Dual 4-Input Data Selector/Multiplexer with 3-State Outputs	LS253	*4539	LS/CMOS	16
HC257	Quad 2-Input Data Selector/Multiplexer with 3-State Outputs	LS257	*4519	LS	16
HC298	Quad 2-Input Data Selector/Multiplexer with Output Latch	LS298		LS	16
HC354	8-Input Data Selector/Multiplexer with Data and Address Latches and with 3-State Outputs	LS354, *LS356	*4512	LS354	20
HC356	8-Input Data Selector/Multiplexer with Data and Address Latches and with 3-State Outputs	*LS354, LS356	*4512	LS356	20

* Suggested alternative

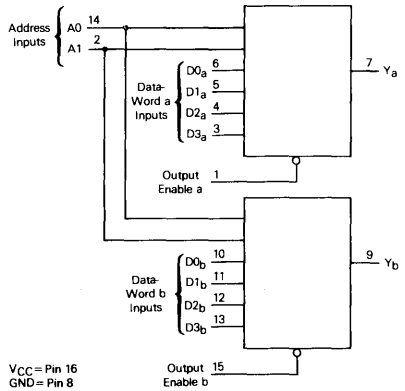
Device	HC 151	HC 153	HC 157	HC 158	HC 251	HC 253	HC 257	HC 298	HC 354	HC 356
#Pins	16	16	16	16	16	16	16	16	20	20
Description	One of 8 inputs is selected	One of 4 inputs is selected	One of two 4-bit words is selected	One of two 4-bit words is selected	One of 8 inputs is selected	One of 4 inputs is selected	One of two 4-bit words is selected	One of two 4-bit words is selected	One of 8 inputs is selected	One of 8 inputs is selected
Single Device	•				•				•	•
Dual Device		•				•				
Quad Device			•	•			•	•		
Data Latch with Active-Low Latch Enable									•	•
Common Address		•	•	•		•	•	•		
1-Bit Binary Address			•	•			•	•		
2-Bit Binary Address		•				•				
3-Bit Binary Address	•				•				•	•
Address Latch (Transparent)									•	
Address Latch (Non-transparent)										•
Active-Low Address Latch Enable									•	•
Output Latch with Active-Low Latch Clock								•		
Noninverting Output	•	•	•		•	•	•	•	•	•
Inverting Output	•			•	•				•	•
3-State Outputs					•	•	•		•	•
Common Output Enable			•	•			•			
Active-High Output Enable									•	•
Active-Low Output Enable	•	•	•	•	•	•	•		••	••

DIGITAL DATA SELECTORS/MULTIPLEXERS

HC151

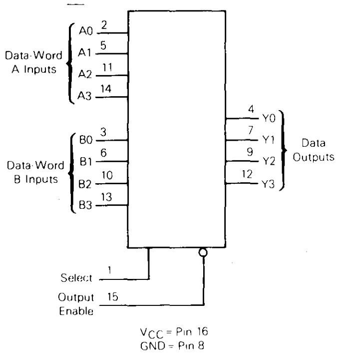


HC153

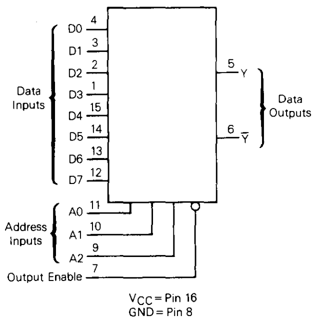


HC157
HC158

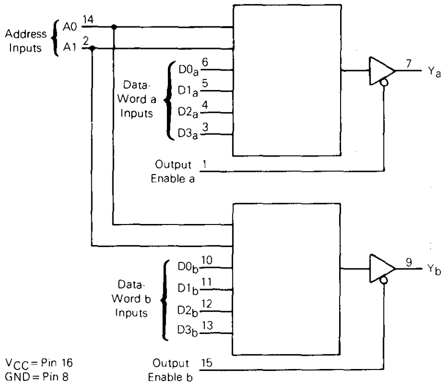
MC54/74HC157 – Noninverting Outputs
MC54/74HC158 – Inverting Outputs



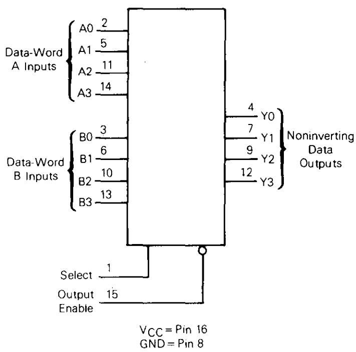
HC251



HC253

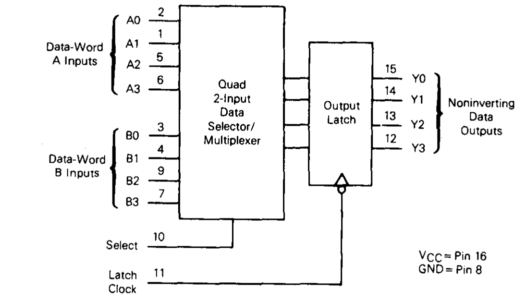


HC257

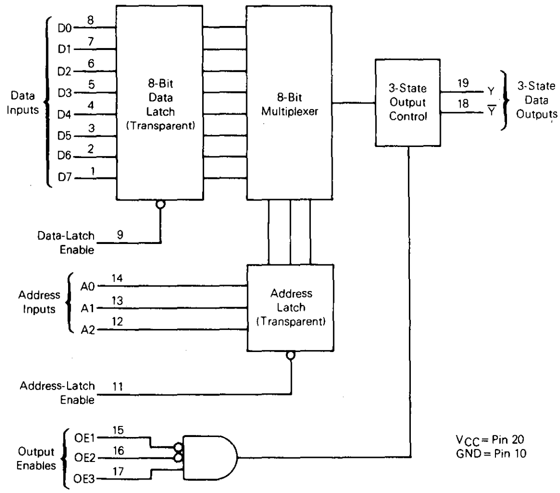


7

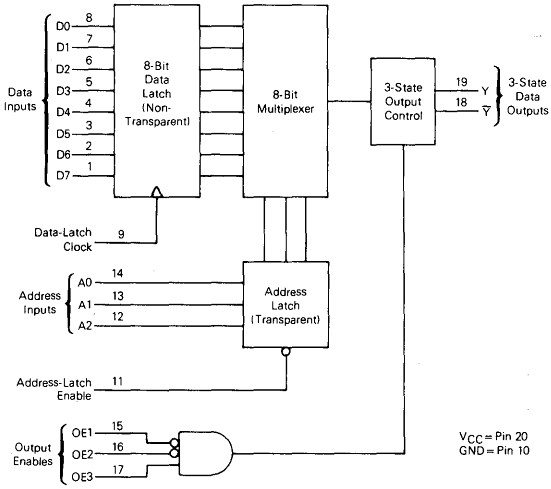
HC298



HC354



HC356



DECODERS/ DEMULTIPLEXERS/ DISPLAY DRIVERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC42	1-of-10 Decoder	LS42	*4028	LS	16
HC137	1-of-8 Decoder/Demultiplexer with Address Latch	LS137	*4028	LS	16
HC138	1-of-8 Decoder/Demultiplexer	LS138	*4028	LS	16
HCT138	1-of-8 Decoder/Demultiplexer with LSTTL-Compatible Inputs	LS138	*4028	LS	16
HC139	Dual 1-of-4 Decoder/Demultiplexer	LS139	4556	LS/CMOS	16
HC147	Decimal-to-BCD Priority Encoder	LS147		LS	16
HC154	1-of-16 Decoder/Demultiplexer	LS154, *LS159	*4515	LS154	24
HC237	1-of-8 Decoder/Demultiplexer with Address Latch	*LS137	*4028	LS	16
HC259	8-Bit Addressable Latch/1-of-8 Decoder	LS259	*4099	LS	16
HC4511	BCD-to-Seven-Segment Latch/Decoder/Display Driver	*LS47, *LS48, *LS49	4511	CMOS	16
HC4514	1-of-16 Decoder/Demultiplexer with Address Latch	*LS154, *LS159	4514, *4515	LS/CMOS	24
HC4543	BCD-to-Seven-Segment Latch/Decoder/Display Driver for Liquid-Crystal Displays	*LS47, *LS48, *LS49	4543	CMOS	16

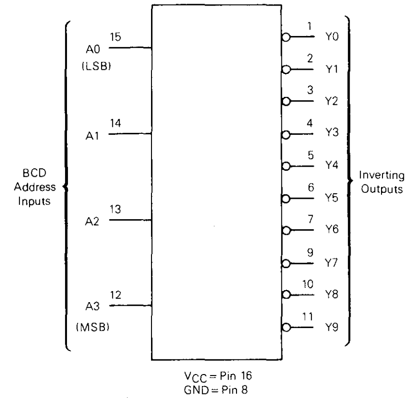
* Suggested alternative

DECODERS/DEMULTIPLEXERS/DISPLAY DRIVERS

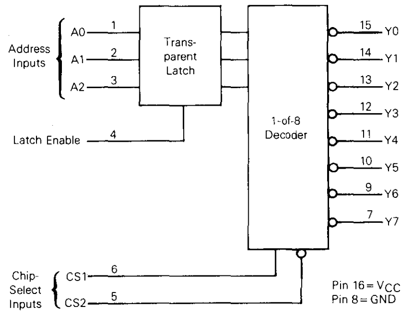
Device	HC42	HC137	HC138	HCT138	HC139	HC147
#Pins	16	16	16	16	16	16
Input Description	BCD Address	3-Bit Binary Address	3-Bit Binary Address	3-Bit Binary Address	2-Bit Binary Address	Any Combination of 9 Inputs
Output Description	One of 10	One of 8	One of 8	One of 8	One of 4	BCD Address of Highest Input
Single Device	•	•	•	•		•
Dual Device					•	
Address Input Latch		•				
Active-High Latch Enable		•				
Active-Low Latch Enable						
Active-Low Inputs						•
Active-Low Outputs	•	•	•	•	•	•
Active-High Outputs						
Active-Low Output Enable		•	••	••	•	
Active-High Output Enable		•	•	•		
Active-Low Reset						
Active-Low Blanking Input						
Active-Low Lamp-Test Input						
Phase Input (for LCD's)						
LSTTL-Compatible Inputs				•		

Device	HC154	HC237	HC259	HC4511	HC4514	HC4543
# Pins	24	16	16	16	24	16
Input Description	4-Bit Binary Address	3-Bit Binary Address	3-Bit Binary Address	BCD Data	4-Bit Binary Address	BCD Data
Output Description	One of 16	One of 8	One of 8	7-Segment Display	One of 16	7-Segment Display
Single Device	•	•	•	•	•	•
Dual Device						
Address Input Latch		•		•	•	•
Active-High Latch Enable		•		•		
Active-Low Latch Enable					•	•
Active-Low Inputs						
Active-Low Outputs	•					
Active-High Outputs		•	•	•	•	•
Active-Low Output Enable	••	•	•		•	
Active-High Output Enable		•				
Active-Low Reset			•			
Active-Low Blanking Input				•		•
Active-Low Lamp-Test Input				•		
Phase Input (for LCD's)						•
LSTTL-Compatible Inputs						

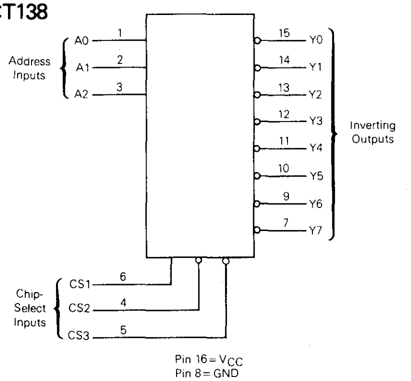
HC42



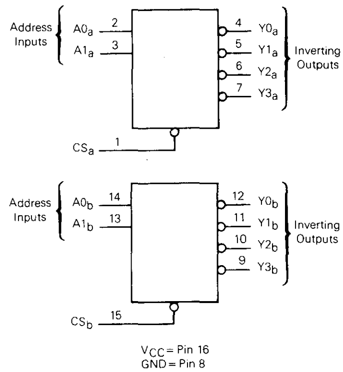
HC137



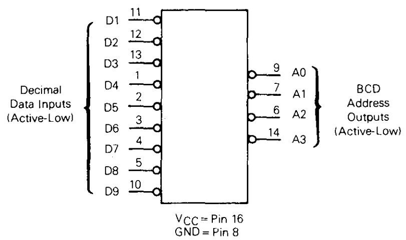
HC138
HCT138



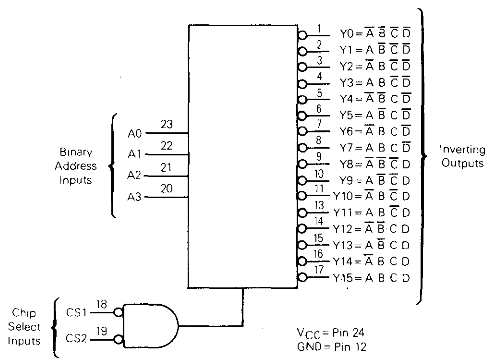
HC139



HC147

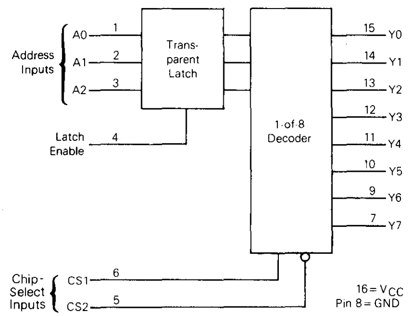


HC154

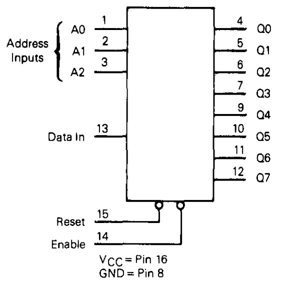


7

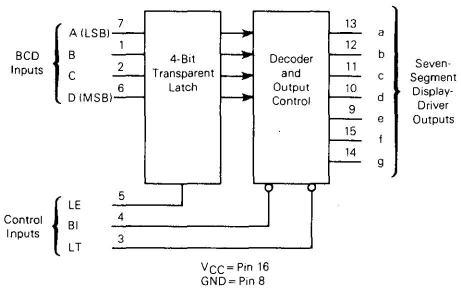
HC237



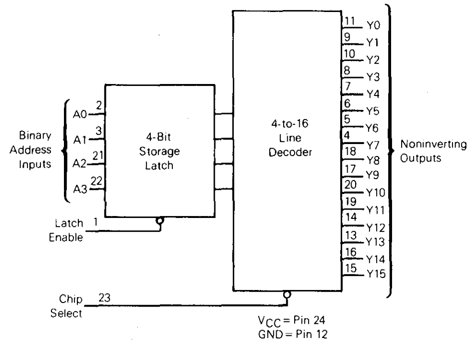
HC259



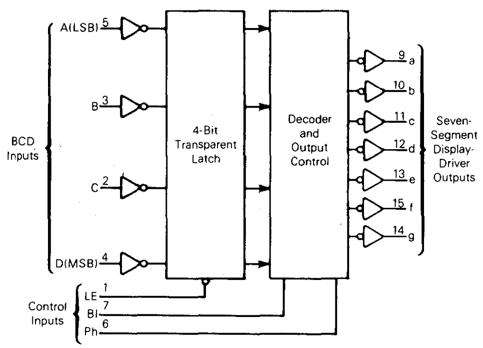
HC4511



HC4514



HC4543



ANALOG SWITCHES/ MULTIPLEXERS/ DEMULTIPLEXERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC4016	Quad Analog Switch/Multiplexer/Demultiplexer		4016,4066	CMOS	14
HC4051	8-Channel Analog Multiplexer/Demultiplexer		4051	CMOS	16
HC4052	Dual 4-Channel Analog Multiplexer/Demultiplexer		4052	CMOS	16
HC4053	Triple 2-Channel Analog Multiplexer/Demultiplexer		4053	CMOS	16
HC4066	Quad Analog Switch/Multiplexer/Demultiplexer with Enhanced On-Resistance Linearity		4066,4016	CMOS	14
☆HC4316	Quad Analog Switch/Multiplexer/Demultiplexer with Separate Analog and Digital Power Supplies		*4016		16
☆HC4351	8-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4051		18
☆HC4352	Dual 4-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4052		18
☆HC4353	Triple 2-Channel Analog Multiplexer/Demultiplexer with Address Latch		*4053		18

* Suggested alternative

☆ High-Speed CMOS design only

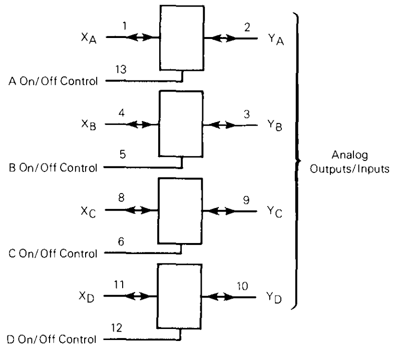
Device	HC4016	HC4051	HC4052	HC4053	HC4066
#Pins	14	16	16	16	14
Description	4 Independently Controlled Switches	A 3-Bit Address Selects One of 8 Switches	A 2-Bit Address Selects One of 4 Switches	A 3-Bit Address Selects Varying Combinations of the 6 Switches	4 Independently Controlled Switches
Single Device		•			
Dual Device			•		
Triple Device				•	
Quad Device	•				•
1-to-1 Multiplexing	•				•
2-to-1 Multiplexing				•	
4-to-1 Multiplexing			•		
8-to-1 Multiplexing		•			
Active-High ON/OFF Control	•				•
Common Address Inputs			•	•	
2-Bit Binary Address			•		
3-Bit Binary Address		•		•	
Address Latch with Active-Low Latch Enable					
Common Switch Enable		•	•	•	
Active-Low Enable		•	•	•	
Active-High Enable					
Separate Analog and Control Reference Power Supplies		•	•	•	
Switched tubs (for R_{ON} and Prop. Delay Improvement)					•

Device	HC4316	HC4351	HC4352	HC4353
#Pins	16	18	18	18
Description	4 Independently Controlled Switches. (Has a separate Analog Lower Power Supply)	A 3-Bit Address Selects One of 8 Switches. (Has an Address Latch)	A 2-Bit Address Selects One of 4 Switches. (Has an Address Latch)	A 3-Bit Address Selects Varying Combinations of the 6 Switches. (Has an Address Latch)
Single Device		•		
Dual Device			•	
Triple Device				•
Quad Device	•			
1-to-1 Multiplexing	•			
2-to-1 Multiplexing				•
4-to-1 Multiplexing			•	
8-to-1 Multiplexing		•		
Active-High ON/OFF Control	•			
Common Address Inputs			•	•
2-Bit Binary Address			•	
3-Bit Binary Address		•		•
Address Latch with Active-Low Latch Enable		•	•	•
Common Switch Enable	•	••	••	••
Active-Low Enable	•	•	•	•
Active-High Enable		•	•	•
Separate Analog and Control Reference Power Supplies	•	•	•	•
Switched tubs (for R_{ON} and Prop. Delay Improvement)				

7

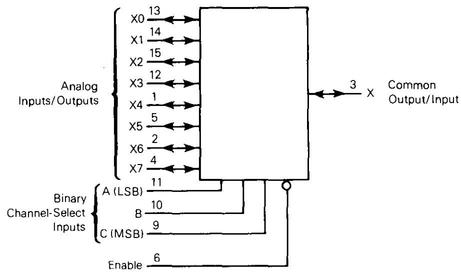
ANALOG SWITCHES/MULTIPLEXERS/DEMULTIPLEXERS

HC4016



X_A, X_B, X_C, X_D = Analog Inputs/Outputs
V_{CC} = Pin 14
GND = Pin 7

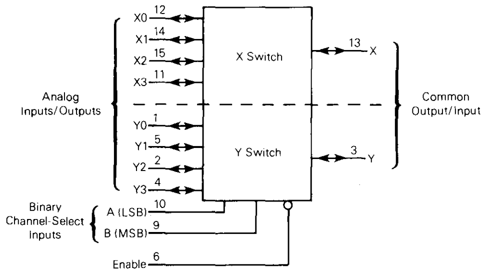
HC4051



V_{CC} = Pin 16 = Positive analog and digital power supply
V_{EE} = Pin 7 = Negative analog power supply
GND = Pin 8 = Negative digital power supply

NOTE: Control inputs are referenced to GND. Analog inputs/outputs are referenced to V_{EE}. V_{EE} must be ≤GND.

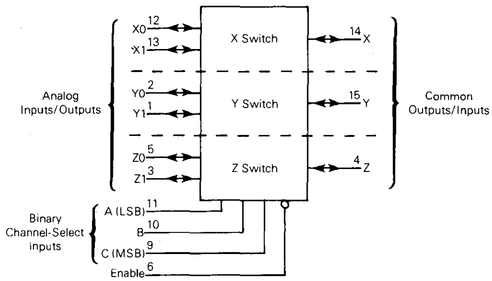
HC4052



V_{CC} = Pin 16 = Positive analog and digital power supply
V_{EE} = Pin 7 = Negative analog power supply
GND = Pin 8 = Negative digital power supply

NOTE: Control inputs are referenced to GND. Analog inputs/outputs are referenced to V_{EE}. V_{EE} must be ≤GND.

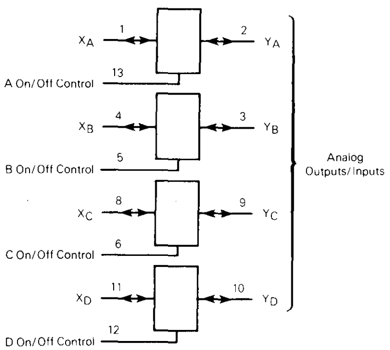
HC4053



V_{CC} = Pin 16 = Positive analog and digital power supply
V_{EE} = Pin 7 = Negative analog power supply
GND = Pin 8 = Negative digital power supply

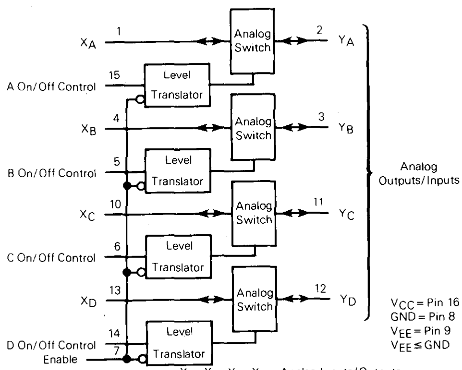
NOTE: Control inputs are referenced to GND. Analog inputs/outputs are referenced to V_{EE}. V_{EE} must be ≤GND.

HC4066



X_A, X_B, X_C, X_D = Analog Inputs/Outputs
V_{CC} = Pin 14
GND = Pin 7

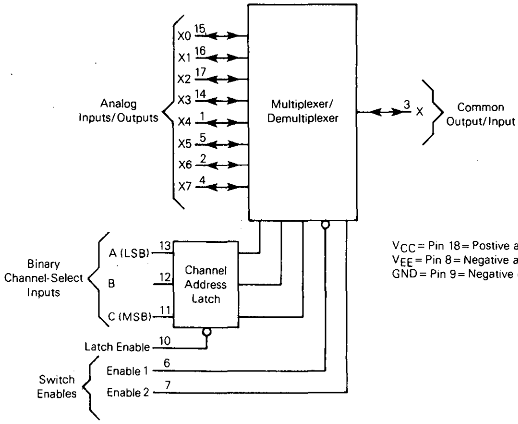
HC4316



X_A, X_B, X_C, X_D = Analog Inputs/Outputs

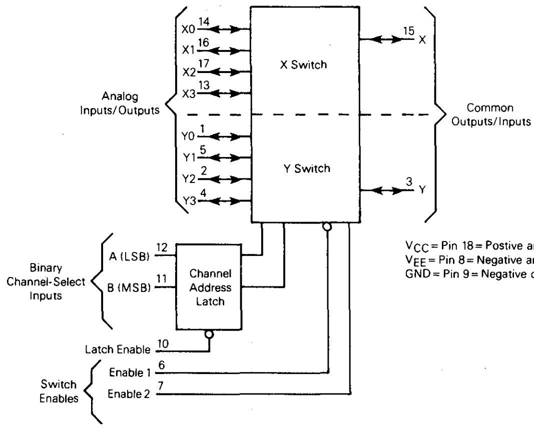
7

HC4351



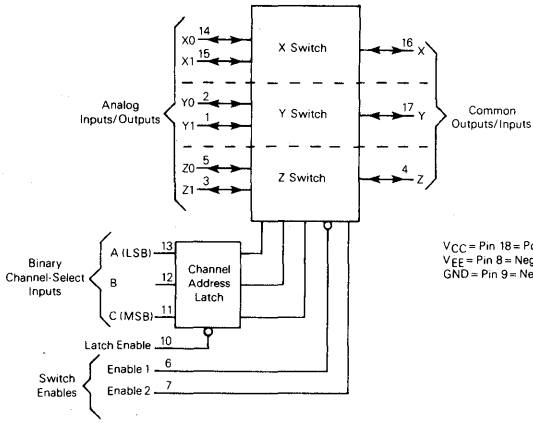
V_{CC} = Pin 18 = Postive analog and digital power supply
V_{EE} = Pin 8 = Negative analog power supply
GND = Pin 9 = Negative digital power supply

HC4352



V_{CC} = Pin 18 = Postive analog and digital power supply
V_{EE} = Pin 8 = Negative analog power supply
GND = Pin 9 = Negative digital power supply

HC4353



V_{CC} = Pin 18 = Postive analog and digital power supply
V_{EE} = Pin 8 = Negative analog power supply
GND = Pin 9 = Negative digital power supply

NOTE: Control inputs are referenced to GND. Analog inputs/outputs are referenced to V_{EE}. V_{EE} must be $\leq$ GND.

7

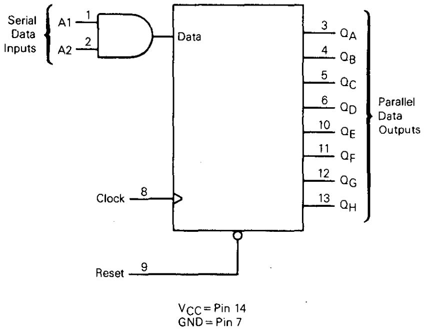
SHIFT REGISTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC164	8-Bit Serial-Input/Parallel-Output Shift Register	LS164	*4034	LS	14
HC165	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register	LS165	*4021	LS	16
HC166	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with Reset	LS166	*4021	LS	16
HC194	4-Bit Bidirectional Universal Shift Register	LS194A	4194	LS/CMOS	16
HC195	4-Bit Universal Shift Register	LS195A	*4035	LS	16
HC299	8-Bit Bidirectional Universal Shift Register with 3-State Parallel Outputs	LS299		LS	20
☆HC589	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with 3-State Output	*LS597			16
HC595	8-Bit Serial-Input/Serial- or Parallel-Output Shift Register with Latched 3-State Outputs	LS595	*4034	LS	16
HC597	8-Bit Serial- or Parallel-Input/Serial-Output Shift Register with Input Latch	LS597		LS	16
HC4015	Dual 4-Bit Serial-Input/Parallel-Output Shift Register		4015	CMOS	16

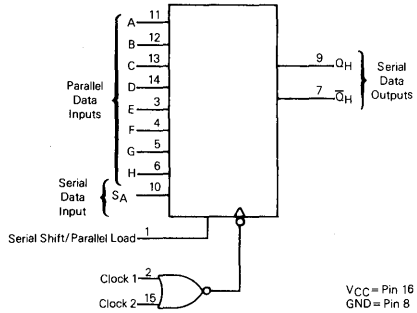
*Suggested alternative
☆High-Speed CMOS design only

Device	HC164	HC165	HC166	HC194	HC195	HC299	HC589	HC595	HC597	HC4015
#Pins	14	16	16	16	16	20	16	16	16	16
4-Bit Register				•	•					•
8-Bit Register	•	•	•			•	•	•	•	
Serial Data Input	•	•	•	•	•	••	•	•	•	•
Parallel Data Inputs		•	•	•	•	•	•		•	
Serial Output Only		•	•				•		•	
Parallel Outputs	•			•	•	•		•		•
Inverting Output		•			•					
Noninverting Output	•	•	•	•	•	•	•	•	•	•
Serial Shift/Parallel Load Control		•	•	•	•	•	•		•	
Shifts One Direction Only	•	•	•		•		•	•	•	•
Shifts Both Directions				•		•				
Positive-Transition Clocking	•	•	•	•	•	•	•	•	•	•
Active-High Clock Enable		•	•							
Input Data Enable	•									
Data Latch with Active-High Latch Clock							•		•	
Output Latch with Active-High Latch Clock								•		
3-State Outputs						•	•	•		
Active-Low Output Enable						••	•	•		
Active-High Reset										•
Active-Low Reset	•		•	•	•	•		•	•	

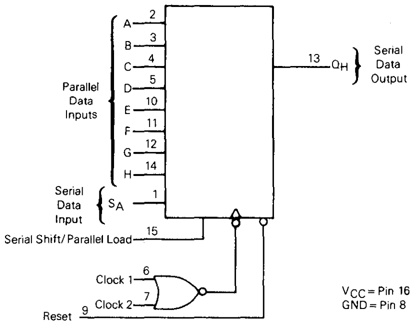
HC164



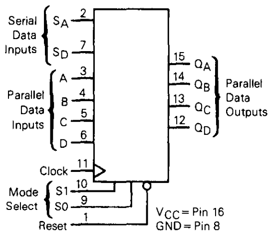
HC165



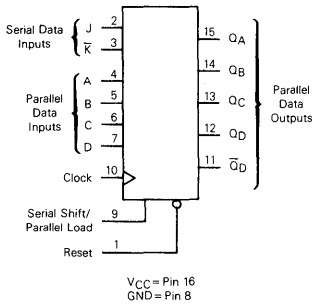
HC166



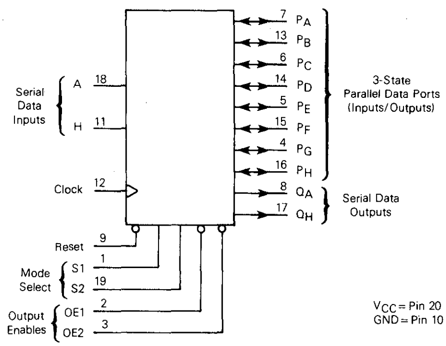
HC194



HC195

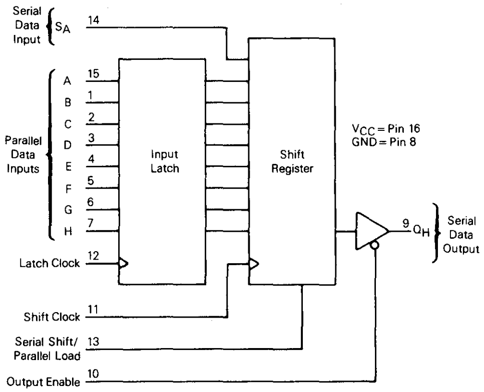


HC299

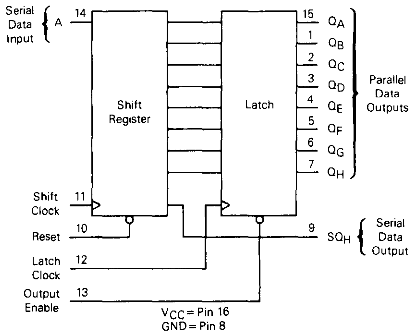


7

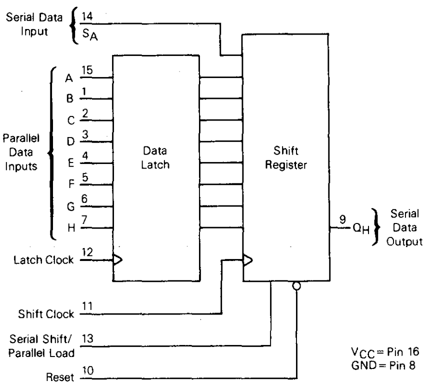
HC589



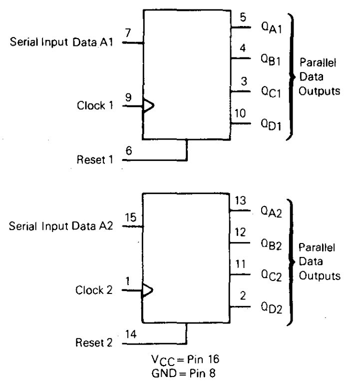
HC595



HC597



HC4015



7

COUNTERS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC90	4-Stage Binary Ripple Counter with ÷ 2 and ÷ 5 Sections	LS90		LS	14
HC92	4-Stage Binary Ripple Counter with ÷ 2 and ÷ 6 Sections	LS92		LS	14
HC93	4-Stage Binary Ripple Counter with ÷ 2 and ÷ 8 Sections	LS93		LS	14
HC160	Presettable BCD Counter with Asynchronous Reset	LS160A	4160	LS/CMOS	16
HC161	Presettable 4-Bit Binary Counter with Asynchronous Reset	LS161A	4161	LS/CMOS	16
HC162	Presettable BCD Counter with Synchronous Reset	LS162A	4162	LS/CMOS	16
HC163	Presettable 4-Bit Binary Counter with Synchronous Reset	LS163A	4163	LS/CMOS	16
HC190	Presettable BCD Up/Down Counter	LS190	*4510	LS	16
HC191	Presettable 4-Bit Binary Up/Down Counter	LS191	*4516	LS	16
HC192	Presettable BCD Up/Down Counter with Reset	LS192	*4510	LS	16
HC193	Presettable 4-Bit Binary Up/Down Counter with Reset	LS193	*4516	LS	16
HC390	Dual 4-Stage Binary Ripple Counter with ÷ 2 and ÷ 5 Sections	LS390		LS	16
HC393	Dual 4-Stage Binary Ripple Counter	LS393	*4520	LS	14
HC4017	Decade Counter/Divider		4017	CMOS	16
HC4020	14-Stage Binary Ripple Counter		4020	CMOS	16
HC4024	7-Stage Binary Ripple Counter		4024	CMOS	14
HC4040	12-Stage Binary Ripple Counter		4040	CMOS	16
HC4060	14-Stage Binary Ripple Counter with Oscillator		4060	CMOS	16
HC4518	Dual BCD Counter		4518	CMOS	16
HC4520	Dual 4-Bit Binary Counter		4520	CMOS	16

*Suggested alternative

7

COUNTERS

Device	HC 90	HC 92	HC 93	HC 160	HC 161	HC 162	HC 163	HC 190	HC 191	HC 192
#Pins	14	14	14	16	16	16	16	16	16	16
Single Device	•	•	•	•	•	•	•	•	•	•
Dual Device										
Ripple Counter	•	•	•							
Number of Ripple Counter Internal Stages	4	4	4							
Number of Stages with Available Outputs	4	4	4							
Count Up	•	•	•	•	•	•	•	•	•	•
Count Down								•	•	•
4-Bit Binary Counter			•		•		•		•	
BCD Counter	•			•		•		•		•
Decimal Counter										
Separate ÷ 2 Section	•	•	•							
Separate ÷ 5 Section	•									
Separate ÷ 6 Section		•								
Separate ÷ 8 Section			•							
On-Chip Oscillator Capability										
Separate Count-Up and Count-Down Clocks										•
Count Up/Count Down Control Input								•	•	
Positive-Transition Clocking				•	•	•	•	•	•	•
Negative-Transition Clocking	•	•	•							
Active-High Clock Enable										
Active-Low Clock Enable										
Active-High Count Enable				••	••	••	••			
Active-Low Count Enable								•	•	
Active-High Set	•									
Active-High Reset	•	•	•	•	•	•	•			•
4-Bit Binary Preset Data Inputs					•		•		•	
BCD Preset Data Inputs				•		•		•		•
Active-Low Load Preset				•	•	•	•	•	•	•
Carry Output				•	•	•	•	•	•	•
Borrow Output										•
Ripple Clock Output								•	•	

7

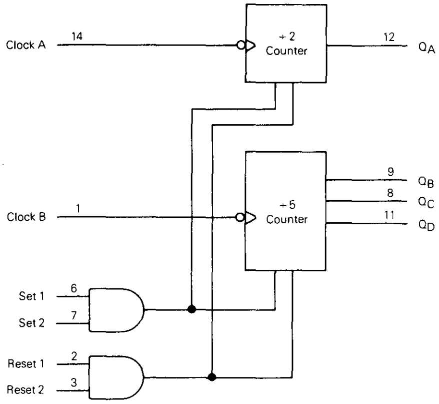
COUNTERS

Device	HC 193	HC 390	HC 393	HC 4017	HC 4020	HC 4024	HC 4040	HC 4060	HC 4518	HC 4520
#Pins	16	16	14	16	16	14	16	16	16	16
Single Device	•			•	•	•	•	•		
Dual Device		•	•						•	•
Ripple Counter		•	•		•	•	•	•		
Number of Ripple Counter Internal Stages		4	4		14	7	12	14		
Number of Stages with Available Outputs		4	4		12	7	12	10		
Count Up	•	•	•	•	•	•	•	•	•	•
Count Down	•									
4-Bit Binary Counter	•		•							•
BCD Counter		•							•	
Decimal Counter				•						
Separate ÷ 2 Section		•								
Separate ÷ 5 Section		•								
Separate ÷ 6 Section										
Separate ÷ 8 Section										
On-Chip Oscillator Capability								•		
Separate Count-Up and Count-Down Clocks	•									
Count Up/Count Down Control Input										
Positive-Transition Clocking	•			•					•	•
Negative-Transition Clocking		•	•	•	•	•	•	•	•	•
Active-High Clock Enable				•					•	•
Active-Low Clock Enable				•					•	•
Active-High Count Enable										
Active-Low Count Enable										
Active-High Set										
Active-High Reset	•	•	•	•	•	•	•	•	•	•
4-Bit Binary Preset Data Inputs	•									
BCD Preset Data Inputs										
Active-Low Load Preset	•									
Carry Output	•									
Borrow Output	•									
Ripple Clock Output										

7

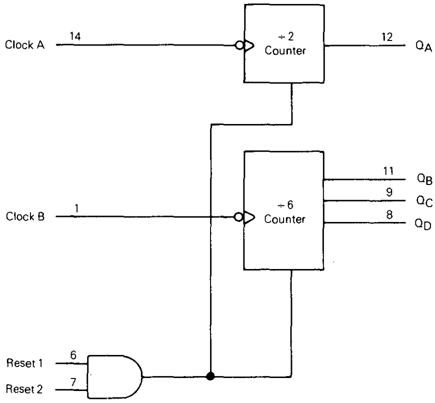
COUNTERS

HC90



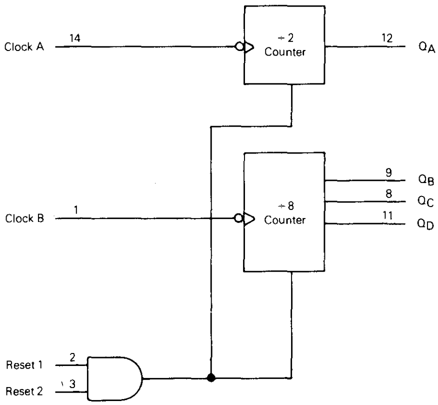
V_{CC} = Pin 5
GND = Pin 10
No Connection = Pins 4, 13

HC92



V_{CC} = Pin 5
GND = Pin 10
No Connection = Pins 2, 3, 4, 13

HC93

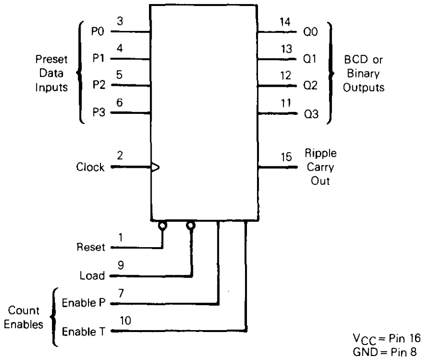


V_{CC} = Pin 5
GND = Pin 10
No Connection = Pins 4, 6, 7, 13

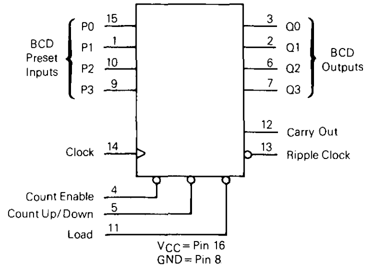
7

COUNTERS

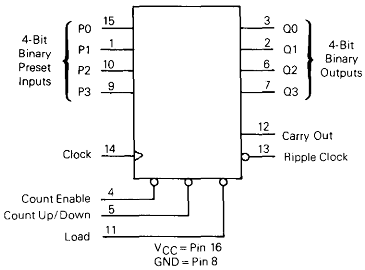
HC160 HC162
HC161 HC163



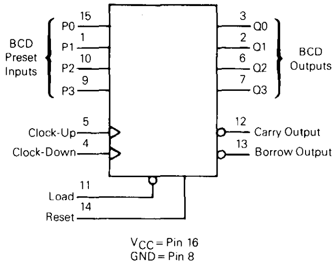
HC190



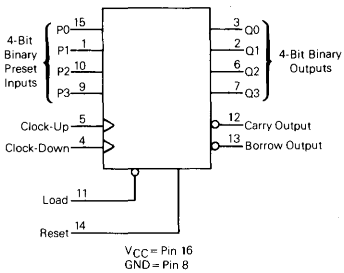
HC191



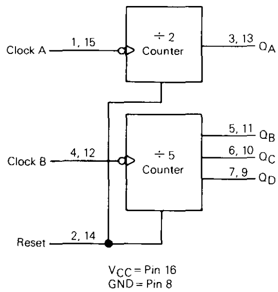
HC192



HC193



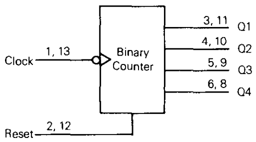
HC390



7

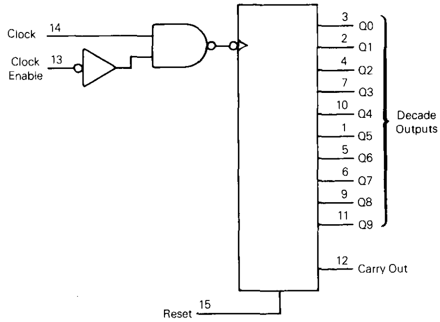
COUNTERS

HC393



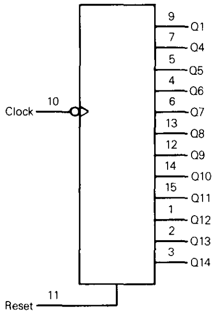
V_{CC} = Pin 14
GND = Pin 7

HC4017



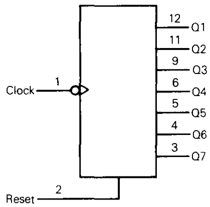
V_{CC} = Pin 16
GND = Pin 8

HC4020



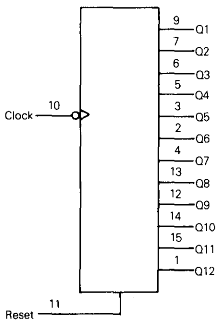
V_{CC} = Pin 16
GND = Pin 8

HC4024



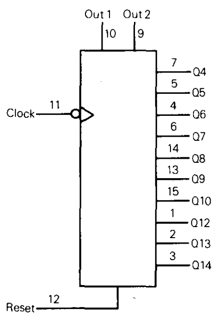
V_{CC} = Pin 14
GND = Pin 7
No Connection = Pins 8, 10 and 13

HC4040



V_{CC} = Pin 16
GND = Pin 8

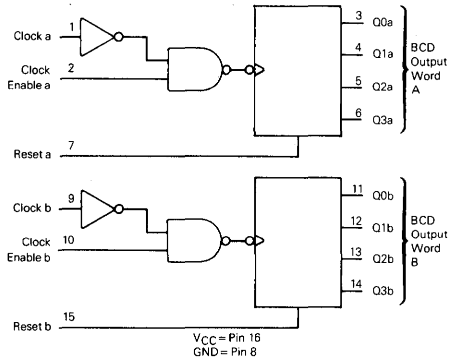
HC4060



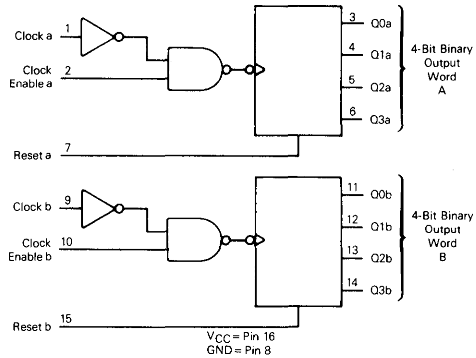
V_{CC} = Pin 16
GND = Pin 8

7

HC4518



HC4520



7

MONOSTABLE MULTIVIBRATORS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC123	Dual Retriggerable Monostable Multivibrator	LS123	*4538, *4528	LS	16
HC221	Dual Monostable Multivibrator	LS221	*4538, *4528	LS	16
HC423	Dual Retriggerable Monostable Multivibrator	LS423	*4538 *4528	LS	16
HC4538	Dual Precision Monostable Multivibrator (Retriggerable, Resettable)	*LS423	4538, 4528	CMOS	16

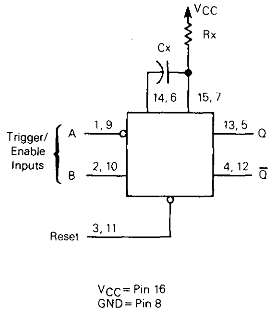
* Suggested alternative

Device	HC123	HC221	HC423	HC4538
#Pins	16	16	16	16
Dual Device	•	•	•	•
Precision Pulse Width				•
Retriggerable	•		•	•
Positive-Transition Trigger	•	•	•	•
Negative-Transition Trigger	•	•	•	•
Active-Low Trigger Enable	•	•	•	•
Active-High Trigger Enable	•	•	•	•
Active-Low Reset	•	•	•	•
Triggerable by Reset Pin	•	•		
Inverting Output	•	•	•	•
Noninverting Output	•	•	•	•

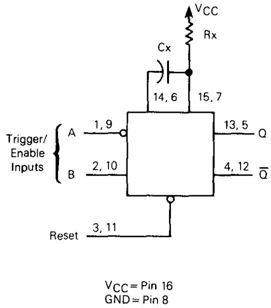
7

MONOSTABLE MULTIVIBRATORS

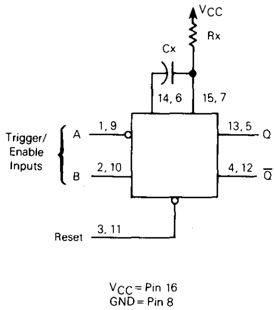
HC123



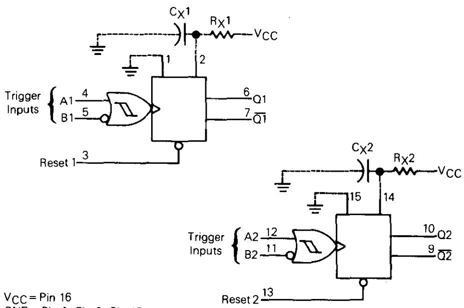
HC221



HC423



HC4538



VCC = Pin 16
GND = Pin 1, Pin 8, Pin 15
RX and CX are external components

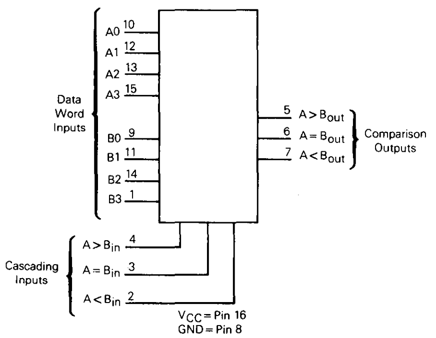
ARITHMETIC CIRCUITS

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC85	4-Bit Magnitude Comparator	LS85	*4585	LS	16
HC181	4-Bit Arithmetic Logic Unit	LS181	4581	LS/CMOS	24
HC182	Carry Lookahead Generator	LS182	4582	LS/CMOS	16
HC280	9-Bit Odd/Even Parity Generator/Checker	LS280	*4531	LS	14
HC283	4-Bit Binary Full Adder with Fast Carry	LS283, LS83	4008	LS283	16
HC688	8-Bit Equality Comparator	LS688		LS	20
HCT688	8-Bit Equality Comparator with LSTTL-Compatible Inputs	LS688		LS	20

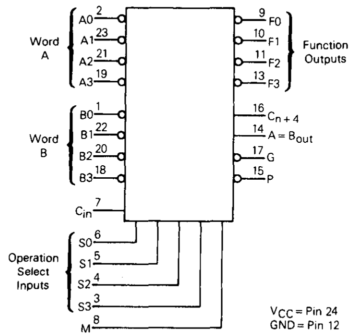
* Suggested alternative

ARITHMETIC CIRCUITS

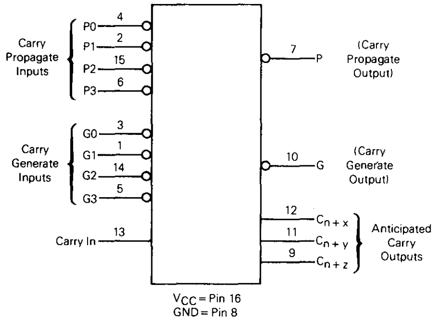
HC85



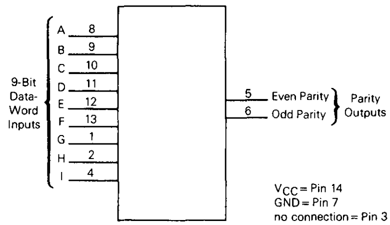
HC181



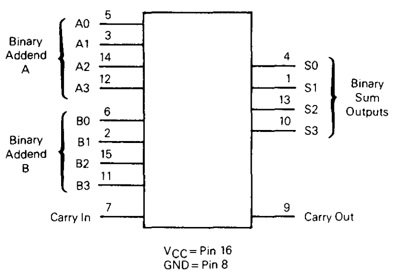
HC182



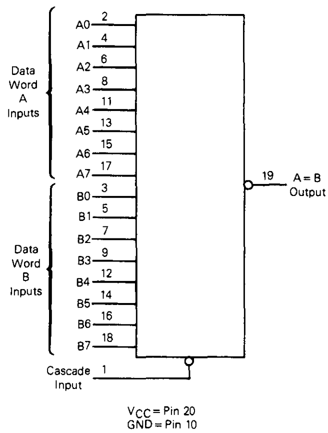
HC280



HC283



HC688
HCT688

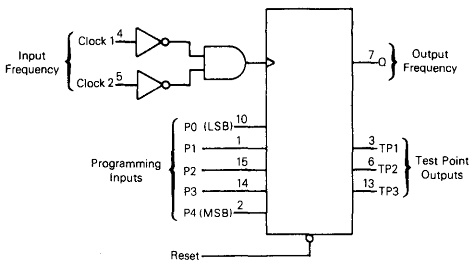


MISCELLANEOUS DEVICES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HC292	Programmable Frequency Divider/Digital Timer	LS292		LS	16
HC294	Programmable Frequency Divider/Digital Timer	LS294		LS	16
HC4046	Phase-Locked Loop	*LS297	4046	CMOS	16

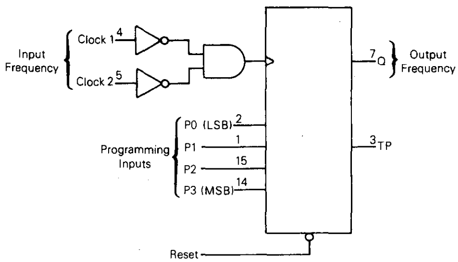
* Suggested alternative

HC292



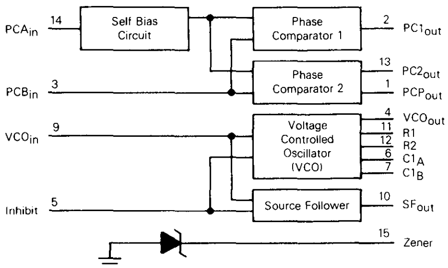
V_{CC} = Pin 16
GND = Pin 8
No Connection = Pin 9, 12

HC294



V_{CC} = Pin 16
GND = Pin 8
Pins 6, 9, 10, 12, 13 = No Connection

HC4046



V_{CC} = Pin 16
GND = Pin 8

7

LSTTL INPUT-COMPATIBLE DEVICES

Device Number MC54/MC74	Function	Functional Equivalent LSTTL Device 54/74	Functional Equivalent CMOS Device MC1XXXX or CDXXXX	Direct Pin Compatibility	Number of Pins
HCT00	Quad 2-Input NAND Gate with LSTTL-Compatible Inputs	LS00	4001	LS	14
HCT04	Hex Inverter with LSTTL-Compatible Inputs	LS04	*4069	LS/CMOS	14
HCT34	Hex Buffer with LSTTL-Compatible Inputs	LS07	*4050	LS	14
HCT138	1-of-8 Decoder/Demultiplexer with LSTTL-Compatible Inputs	LS138	*4028	LS	16
HCT240	Octal 3-State Inverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS240		LS	20
HCT241	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS241		LS	20
HCT244	Octal 3-State Noninverting Buffer/Line Driver/Line Receiver with LSTTL-Compatible Inputs	LS244		LS	20
HCT245	Octal 3-State Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS245		LS	20
HCT373	Octal 3-State Noninverting D-Type Transparent Latch with LSTTL-Compatible Inputs	LS373, LS573		LS373	20
HCT374	Octal 3-State Noninverting D-Type Flip-Flop with LSTTL-Compatible Inputs	LS374, LS574		LS374	20
HCT640	Octal 3-State Inverting Bus Transceiver with LSTTL-Compatible Inputs	LS640		LS	20
HCT643	Octal 3-State Inverting and Noninverting Bus Transceiver with LSTTL-Compatible Inputs	LS643		LS	20
HCT688	8-Bit Equality Comparator with LSTTL-Compatible Inputs	LS688		LS	20

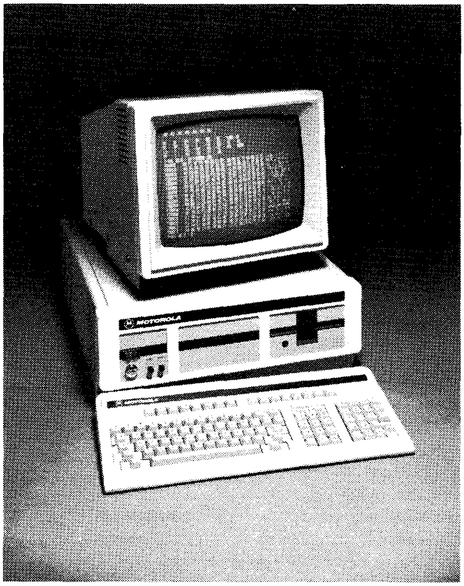
*Suggested alternative

Development Systems and Board-Level Products

8

MOTOROLA
MICROCOMPUTER DEVELOPMENT SYSTEMS

VME/10 MICROCOMPUTER SYSTEM



The VME/10 Microcomputer System is a compact yet powerful desktop designer's workstation that can be used for developing advanced microprocessor-based systems using Motorola's 8-bit and 16-bit families of microprocessors, microcomputers, and peripheral components.

MAJOR BENEFITS

- Provides Efficient Design Support for M6800 and M68000 MPU Families
- Excellent Development Software Complement
- Customizable Through VMEbus and I/O Channel for End Applications
- Multi-mode Graphics Hardware with Both Monochrome and Color Options.

The VME/10 Microcomputer System combines the flexibility of a customizable workstation with the attributes of a powerful development support system that let the system integrator or OEM design an end product with the same hardware and software that can eventually constitute the end system itself. With appropriate interfaces and peripherals, these systems may be specialized designers' workstations, or perhaps front-end processors associated with larger external equipments such as factory automation systems or large complex medical diagnostic instruments. In addition to raw processing power, these small but capable systems have the flexibility for just the right I/O and performance improvement features for dedicated, user-defined systems.

BASIC DESIGN FEATURES

- MC68010 16/32-bit Microprocessor Unit
- MC68451 Memory Management Unit
- Industry-standard VMEbus interface with full bus arbitration logic and software controllable interrupter.
- I/O Channel Interface for adding off-board resources such as A/D converters, serial and parallel I/O ports, etc.
- 384K Byte Dynamic RAM (multiported between graphics controller and local bus, and VMEbus).
- 8K Byte Static RAM for storage of user-definable character sets and display attributes.
- Two 28-pin sockets for ROM/PROM/EPROM storage of up to 64K bytes for custom applications.
- Battery backed-up time-of-day clock with 50 bytes of CMOS RAM storage.
- 15″ video display having the following software controllable display formats:
  1. 25 lines by 80 characters — 8 x 10 characters with descenders (10 x 12 character field)
  2. 800 x 300 pixel for low resolution graphics
  3. 800 x 600 pixel for medium resolution graphics
  4. Pixel graphics with overlaid character displays
- Monochrome video display standard, with 7-level gray scaling (color optional).
- Detachable full ASCII keyboard with cursor control keys, numeric pad and 16 function keys.
- Mass Storage Subsystem providing both 5¼″ Floppy Disk and 5¼″ Winchester Disk Storage Units.

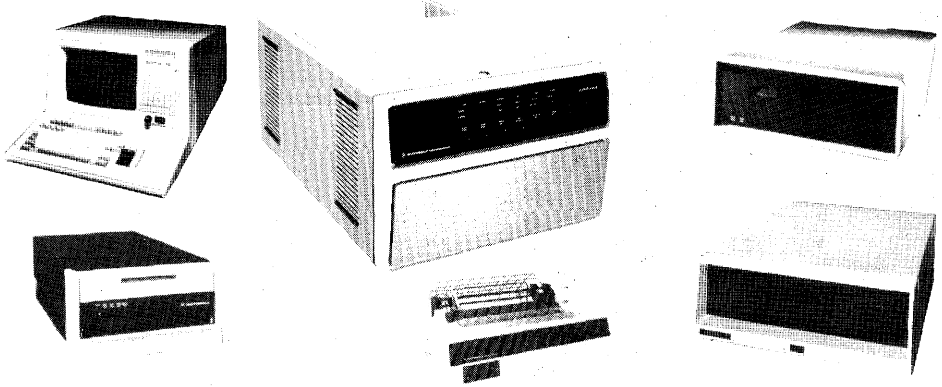
  Floppy Disk
  1 Mbyte Unformatted Capacity (655K Byte Formatted)

  Winchester Disk
  Choice of: (a) 6.38 Mbyte Unformatted Capacity (5 Mbyte Formatted)
  (b) 19.1 Mbyte Unformatted Capacity (15 Mbyte Formatted)
- Card cage options for feature expansion capability.
  Choice of: (a) Five I/O Channel Card Cage Slots (with 6.38 Mbyte Winchester option)
  (b) Five VMEbus Card Cage Slots with VMEbus backplane, plus four I/O Channel Slots (with 19.1 Mbyte Winchester option)
- Conformance to ergonomic standards applicable to video display and keyboard.
- VERSAdos Real-Time, Multitasking Operating System with M68000 Family Macro Assembler, plus tools and utilities.
- Capability of hosting hardware development tools
  - HDS-400 for M68000 Family 16/32-bit Emulation
  - HDS-200 for M6800 Family 8-bit Emulation
  - Bus State Analyzer for Logic Analysis Functions

8

EXORmacs

M68000 DEVELOPMENT SYSTEM



- Complete Development System for MC68000 MPU
- Up to Eight User Stations
- Multi-Processor Bus Arbitration
- Multi-Tasking Real-Time Operating System
- Resident Pascal High-Level Language
- Diagnostic Firmware
- Up to 192 Megabyte Fixed/Removable Hard Disk
- And Up To 2 Megabyte Dual Drive Floppy Disk
- Provisions For Future 32-Bit Microprocessors
- Optional Cross-Development Software for 8-Bit MPUs

The EXORmacs Development System is a state-of-the-art instrument for designing and developing advanced 16-bit microprocessor based systems using Motorola families of microprocessors, microcomputers, and peripheral components.

Coupled with the Motorola HDS-400 Microprocessor Hardware/Software Development Station it is also ideally suited for developing applications using the VERSAmodule and VMEmodule families of 16-bit board level application products and accessories.

Designed for flexibility and ease of use, the EXORmacs Development System takes advantage of the power and features of the MC68000 microprocessor unit (MPU). It reduces cost and development time by incorporating features which support 16-bit and future 32-bit microprocessor designs, as well as providing high-level language support through Pascal and FORTRAN. With an appropriate number of accessories, such as terminals, multiple-channel communications modules and hardware development stations, up to eight users may simultaneously develop and debug M68000 programs.

System Expansion Modules

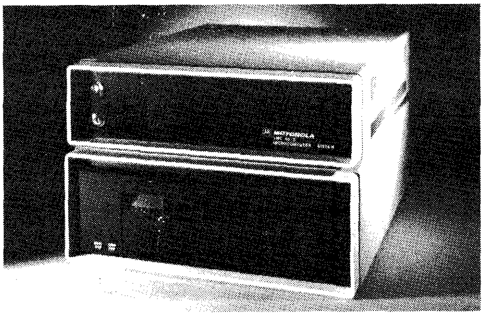
Multichannel Communications
Module (MCCM) — M68KMCCM
VERSAbus Adapter Module — M68KVAM
VERSAbus RAM 128K Byte — M68KVM10-3
VERSAbus RAM 256K Byte — M68KVM11-1
VERSAbus RAM 512K Byte — M68KVM11-2
VERSAbus Extender — M68KEXTM
VERSAbus Wirewrap — M68KWW

EXORmacs Basic System Configurations

- Hardware Chassis — with Power Supply and 15-slot Backplane
  Resident Module Complement
    MC68000 MPU/MMU Module
    DEbug Module
    256K Dynamic RAM
    Disk Controller Module
- Software
  System V/68 Operating System Software
    M68000 System V/68 OS
    M68000 C Language Compiler Assembler and Linker
    Instrumentation Support Utilities
  VERSAdos Software Development Tools
    VERSAdos Operating System
    CRT Editor
    Macro Assembler
    Linkage Editor
    Symbolic Debug
- Peripherals
    EXORterm 155 Display Console
    Choice of Mass Storage:
    1 Megabyte Floppy Disk
    8/8 Megabyte Hard Disk
    25/25 Megabyte Hard Disk
    16/16 Megabyte Hard Disk
    16/80 Megabyte Hard Disk

EXORbus, EXORciser®, EXORmacs®, EXORset, EXORterm, UNIDOS and VERSAbus are trademarks of Motorola Inc.

MOTOROLA
MICROCOMPUTER DEVELOPMENT SYSTEMS (continued)



VMC 68/2 Microcomputer

The VMC 68 Series is a high performance microcomputer system family intended for application by OEM's and system integrators starting from a product integration level formerly available only to the minicomputer user. The VMC 68 System Family will find wide application in industrial process control, automated testing, data acquisition, supervisory control, and many other factory and lab automation uses. The VMC 68 Series is based on the 16-bit M68000-based VERSAmodule Family of modular microcomputer products utilizing the industry and IEEE proposed VERSAbus standard system interconnect providing multiprocessing and intelligent peripheral controller architecture.

Also featured is the I/O channel which provides for the use of a broad selection of I/O modules for I/O flexibility.

VMC 68/2 System Features and Capabilities

Hardware-Only Package	Complete System Package
• MC68000-based VM02 Monoboard Microcomputer — Direct Addressing to 16 Megabytes — 128K Bytes Dual-Port RAM — Multiprocessor Architecture with System Controller Features — I/O Channel Interface for Functional Tailoring — Dual Multiprotocol RS-232C Serial Ports for System Flexibility • Dual 16-Bit Parallel Port I/Omodule — Centronics compatible Printer Interface — General Parallel I/O Applications • VERSAbug Firmware — Debug — Disk Bootstrap Load — Self-Test — Up/Downline Load • MC68120-based VM21 Universal Disk Controller — High-speed DMA data transfer to/from 1 or 2 SMD interface compatible disk drives AND Up to 4 EXORdisk II or III Floppy Disk Drives • VM11 Dynamic RAM Module with 256K Bytes of "global" RAM for program development and efficient multitasking system operation • 4 or 8-slot VERSAbus compatible VMC 68/2 Chassis — Power Fail/Restart Circuitry — 5 or 10 I/Omodule card slots for I/O Channel functional tailoring (Dual Parallel Port module occupies one of these slots) • 0°C to 50°C (32°F to 122°F) Operating Temperature Range • For 115 Vac 60 Hz Operation	In addition to all features of the Hardware-Only package: • MLD-16 Mass Storage Unit incorporating Disk Drive, SMD interface electronic module, and Disk Power Supply • 16 Megabyte (unformatted) 8-inch SMD interface compatible Disk Drive — 8M Bytes Fixed, plus 8M Bytes Removable Cartridge for storage and one-to-one System Backup — Embedded Servo Information to eliminate cartridge interchange problems and the need for head alignment — Simple Installation — Quiet Operation — High Performance — Disk Compartment sealed during operation — Exceptional Reliability (7500 Hour MTBF) — Long Service Life requiring no preventive maintenance in a benign environment • VERSAdos Real-Time Multitasking Operating System with Assembler and Utilities, including: — MC68000 Structured Macro Assembler — Text Editor, Linkage Editor, and Multitasking Debugger — System Diagnostics — System Generation (SYSGEN) capability for feature tailoring of the VERSAdos System • +10°C to +40°C (50°F to 104°F) Operating Temperature Range

Ordering Information

MVMC682-114	Four-slot VMC 68/2 Microcomputer System
MVMC682-118	Eight-slot VMC 68/2 Microcomputer System
MVMC682-114H	Four-slot VMC 68/2 Microcomputer System hardware-only package.
MVMC682-118H	Eight-slot VMC 68/2 Microcomputer System hardware-only package.

8

M68000 System Development Software

SYSTEM V/68 AND VERSAdos OPERATING SYSTEMS

System V/68

The System V/68 Operating System is the standard UNIX-derived Operating System for the M68000 family of microprocessors. It offers a small compact kernel, which provides process scheduling and I/O facilities to all programs. In addition, a powerful command shell for interactive system controls and an extensive set of utility programs for many tasks, such as program development, text processing, electronic mail, and networking support are included.

Host Systems

The System V/68 Operating System is available as the host environment on Motorola development systems. The EXORmacs is a multiuser system capable of supporting up to eight users simultaneously. The VME/10 System is a single-user system. Hard disk is required for System V/68. Future Motorola Microsystems development systems will also be supported by the System V/68 Operating System.

Instrumentation Support

Communications support for the Motorola HDS-400 Hardware Development Station is included in System V/68. This provides customers with the full system development capability (both hardware and software) that they have come to expect from Motorola.

Languages

As an integral part of System V/68, C Language is offered. C Language has developed into one of the most popular commercial programming languages, and is used frequently in developing portable application software. System V/68 offers significant enhancements to C Language, along with several new language utilities. CXREF, a new cross reference program, and CFLOW, a new flow analysis program, are just two of the new utilities offered. System V/68 also includes a FORTRAN 77 compiler as well as an M68000 assembler and linker/loader.

Programmer's Workbench

The Programmer's Workbench utilities support the development of large software systems in a professional manner. They include the Source Code Control System (SCCS), which provides facilities to store, update and retrieve all versions of source code modules; YACC, which generates parsers; LEX, which builds lexical analyzers; and other utilities which enhance programmer productivity and the quality of work.

VERSAdos

The M68000 Real-time Operating System (VERSAdos) provides complete real-time, multitask support for the EXORmacs User. Features included in the VERSAdos are:

- Real-time multitasking executive
- Device independent I/O
- Floppy and hard disk support
- Sequential, random, and index sequential file capabilities.

CRT Text Editor

The EXORmacs CRT-oriented Text Editor runs under the supervision of the Operating System and provides the capability to create and modify source programs. The editor supports both command and cursor editing, utilizing the cursor, control characters and function keys of the EXORterm 155.

Structured Assembler

The M68000 Structured Macro Assembler translates source statements into relocatable machine code, assigns storage locations to instructions and data, performs auxiliary assembler actions designated by the programmer, and optionally produces a cross-reference listing. The M68000 resident assembler includes macro and conditional assembly capabilities plus certain control constructs that permit structured programming at the assembly language level.

Linkage Editor

The Linkage Editor provides the capability of merging two or more separately-compiled object units into a loadable object module file.

Symbolic Debug

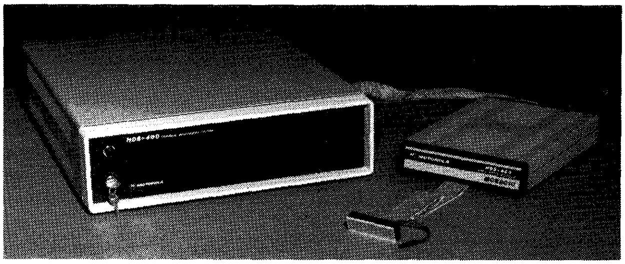
The SYMbug/A program is used to debug other programs, whose source code may have been written in Motorola-provided assembler language, for execution on the M68000. The language processors, in cooperation with the Linkage Editor, supply symbolic information to SYMbug/A. This permits the user to describe the debugging requirements to SYMbug in terms close to the language in which the source program was written.

Pascal Compiler (Optional) M68K0PASCALH

Pascal is a block structured high order language that promotes good programming technique, is self-documenting, and simplifies program writing.

FORTRAN Compiler (Optional) M68K0FORTRNH

Motorola's FORTRAN exceeds ANSI FORTRAN 77 subset language specification, providing real-time processing capabilities.



Control Station
M68KHDS400
M68KHDS400A

Family Board
M68KHDS16FB

Emulators
M68000HDS4
M68008HDS4-8
M68010HDS4-8

Software
M68KHDS4-XX

HDS-400 MICROPROCESSOR HARDWARE/SOFTWARE DEVELOPMENT STATION

Design Features

- 12.5 MHz Real-Time Emulation for MC68000 MPUs
- 10 MHz Real-Time Emulation with no Wait Cycles for MC68000 MPUs
- 8.0 MHz Emulation for MC68008 and MC68010 MPUs
- No User Target System Restrictions
- 32K bytes of 10 MHz No Wait Cycle Emulation RAM is Standard
- Emulation RAM Expandable to 64K, 128K or 256K bytes
- Full Symbolic Debug with EXORmacs and VME/10 Hosts
- Unrestricted User Memory Map
- One-Line Assembler/Disassembler
- Automatic Self-Test of Development Station Hardware
- M68KHDS400 Interfaces with EXORmacs Development System
- M68KHDS400A Interfaces with Motorola VME/10 and DEC VAX Hosts
- Compatible with Real-Time Bus State Analyzer

Major Benefits

- Reduces Development Costs
- Shortens Product Development Cycle
- Brings Product to Market Faster
- Versatility Protects Against Obsolescence

The HDS-400 Microprocessor Hardware/Software Development Station, in conjunction with a Motorola EXORmacs Development System or VME/10 Microcomputer System, or a DEC VAX Computer, provides a complete hardware/software development system for the Motorola M68000 family of microprocessors. It consists of a Control Station, with all the support circuitry for complete MPU emulation, and a separate Emulator Module with an internal microprocessor to match the particular MPU it is expected to emulate.

Two key capabilities of the HDS-400 make it very useful as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected MPU in the user's target system. By plugging the HDS-400 into the socket on the prototype hardware, it allows efficient testing and debugging of both hardware and software. The second capability is the rapid debug and integration of the target system for the production of prototypes. This is accomplished by the use of the powerful set of commands in the HDS-400. The user may execute the commands by either entering the command code and its parameters, or by sequentially depressing function keys which provide a "fill-in-the-blanks" format with parameters such as file name, address, data, etc. When a single function key or a combination of function keys is pressed, a command code is automatically generated and the command syntax is displayed by the system.

EXORmacs® and VME/10 are trademarks of Motorola Inc.
DEC and VAX are trademarks of Digital Equipment Corporation.

Typical System Configuration

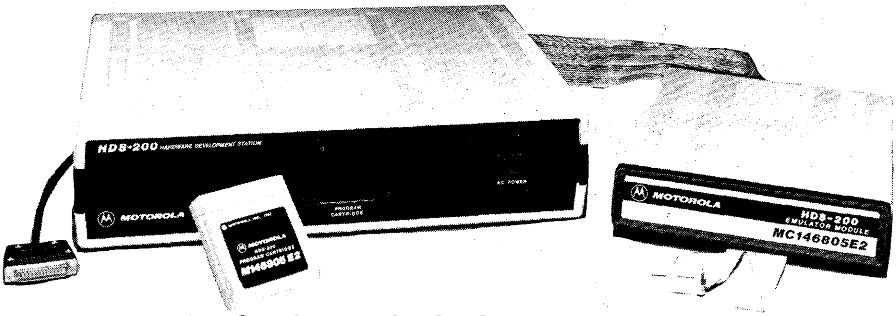
The HDS-400 Hardware/Software Development Station includes a four-slot Control Station with a built-in 30 A power supply and an Emulator Module for the specific MPU which will be used in the target system. Emulators are available for the MC68000, MC68008, and MC68010 MPUs.

The HDS-400 has been partitioned with options and part numbers that give the user versatility in defining the development system configuration. The user may choose from three host computers EXORmacs, VME/10, or DEC VAX with a variety of operating systems. Each of the HDS-400 Control Stations is delivered pre-wired to accept the optional Emulation Memory Module and the Real-Time Bus State Analyzer (BSA). EXORterm 155 is required in HDS-400 systems hosted by the EXORmacs and the VAX. The VME/10 functions as both host and terminal to the HDS-400, eliminating the need for a separate terminal in VME/10-based systems.

System Performance

The HDS-400 Development Station, when substituted for the MPU chip in the target system being debugged, performs the functions of the microprocessor being emulated — exactly as the MPU would have performed were it still in the circuit being tested. The emulator provides the interfacing with the RAM, ROM, and I/O devices and operates at the same speed as the MPU. There are no restrictions on the use of emulation memory that are not imposed by the MPU itself, and the memory may be mapped to the target system or to the emulator module.

The standard 32K bytes of emulation RAM provided in the Family Interface Module may be expanded with one of three optional Emulation Memory Modules. The three memory expansion modules available increase the 10 MHz no wait cycle emulation RAM to 64K, 128K or 256K bytes.



Control Station
M68HDS201

Emulators
M6804P2HM
M6805P234HM
M6805RU23HM
M6805S2HM
M6805T2HM
M146805E2HM
M146805F2HM
M146805G2HM

HDS-200 MICROCOMPUTER HARDWARE/SOFTWARE DEVELOPMENT STATION

Design Features

- Real-time emulation for M6804/M6805/M146805 MCUs.
- Sixteen programmed breakpoints.
- Prioritized breakpoints.
- Line-by-line assembler/disassembler.
- Program trace commands.
- Commands displayed for operator HELP.
- Memory map display.
- Macro commands stored for re-use.
- Transparent mode for host communication.
- Emulates more than 20 MCUs.

Operating Features

- Compatible with EXORmacs, EXORciser and EXORset software development systems.
- Low cost.
- Stand alone operation — frees software development system for parallel use.
- Easy to use.
- Operates with any standard RS-232C terminal and most host systems.

The HDS-200 Hardware Development System, in conjunction with a Motorola EXORset, EXORmacs or EXORciser software development system, provides a complete hardware/software development system for the Motorola M6804/M6805/M146805 families of microprocessors. It consists of a Control Station, with all the functional circuitry to complete MCU emulation, and a separate Emulator Module with an internal microcomputer and memory capacity to match that of the particular MCU it is expected to emulate.

Two major factors contribute to the HDS-200's usefulness as a systems development tool. The first is the ability to serve as a fully functional substitute for the selected MCU in the target system. By plugging the HDS-200 into the socket on the prototype hardware, it allows efficient testing of hardware as well as software debugging. The second factor is its powerful list of analysis commands. These easy-to-use, plain language commands enable the user to rapidly debug, integrate the target system and produce prototype systems.

Systems Development and Integration

The initial stages of developing an MCU-based system normally involve two parallel, rather independent, efforts. One is the hardware design — the other the software design. These efforts are frequently accomplished by two different teams of personnel, resulting in debugging problems that are often difficult and time-consuming. The HDS-200 simplifies this process because of its ability to bring the hardware and software development processes into intimate relationship with each other throughout the development cycle. Moreover, with the HDS-200 it becomes economically feasible to test alternate design approaches in order to determine the best solution.

The complete HDS-200 Hardware/Software Development System consists of three separate items — the HDS-200 Control Station, the Emulator Module, and an associated Firmware Cartridge.

HDS-200 Control Station

The station contains an internal power supply, logic circuits, clock and an MC6809 MPU. The MPU runs the monitor, controls the ports and interfaces with the emulators. It has two RS-232C communication ports for interconnecting with a host computer and a suitable terminal. Another cable connects the station to an associated outboard Emulator Module.

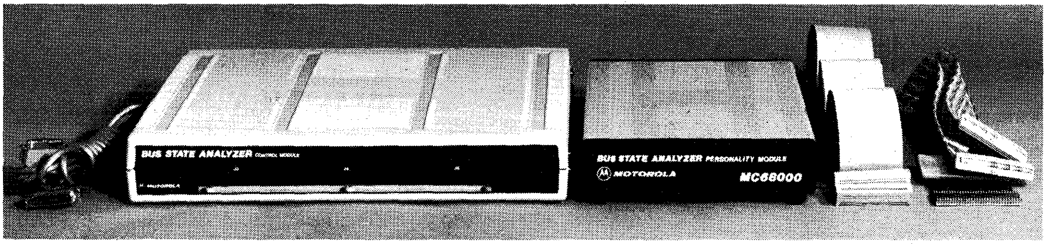
The Emulator Module

The module's output to the user's system is by a short, noisefree ribbon cable terminating in a plug to mate with the target hardware MCU socket. The emulator contains the target processor and various I/O interfaces to provide a compatible link between the Control Station and target hardware MCU/MPU socket.

Different modules are available for specific microprocessor family types. The various MCU Emulator Modules available include the M6804, M6805, M68705, and M146805 families. Each module comes with a matching Firmware Cartridge and an emulator cable/connector assembly.

The Firmware Cartridge

Paired with each Emulator Module is a small cartridge which is easily plugged into the HDS-200 station. This cartridge contains the necessary programs on ROM to enable the HDS-200 to adapt to the specific "personality" of the selected MCU type.



REAL-TIME BUS STATE ANALYZER

The Real-Time Bus State Analyzer (BSA) is a highly intelligent diagnostic tool that is designed specifically for use with microprocessors. It consists of a Control Module plus one of several "Personality Modules." The Control Module contains the analyzer hardware, control firmware, and I/O ports. The Personality Modules interface to selected MPU/MCU, EXORbus, or VERSAbus signals. The BSA stores data which appears on between 55 and 79 different lines.

In order to facilitate the gathering of pertinent data from the MPU/MCU or bus, a set of qualified triggering modes are provided. These modes can be broken into three categories: Continuous Trace Mode, Sequential Trigger Mode, and Window Trigger Mode.

CONTINUOUS TRACE MODE samples signals and stores signal information continuously on each occurrence of the clocking signal. It is primarily a default mode which the BSA automatically enters when power is first applied. There are no qualifications for the BSA to begin gathering information, so it will always be storing the signal states. This default mode is particularly useful when a sudden catastrophic failure occurs during a debugging session, before the user is able to configure the BSA. It is very likely that the events leading up to the failure will be stored in the BSA's trace store buffer.

SEQUENTIAL TRIGGER MODE requires that a series of events occur before the instrument triggers and starts to gather data; or conversely triggers and stops gathering data. Sequence Terms, as these events are called, must occur in order of specification, or triggering will not take place. A Sequence Reset Term can also be specified to reset the BSA and cause the instrument to begin looking for the Sequence Terms again. Sequential triggering will be most useful for debugging complex software, including loops, nested subroutines and complex branches.

WINDOW TRIGGER MODE provides a means of causing signal states to be stored if address accesses occur inside or outside of a particular address range. Both the upper and lower bounds of the range are programmable, and the size is variable from a single address to the full range of the memory map. Window triggering is useful for following programs that suddenly and unexpectedly leave the memory area in which they should be operating. It is also applicable for observing access violations in a multiple user environment.

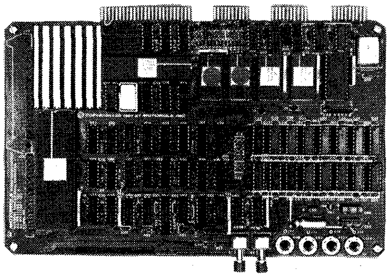
SOFTWARE PERFORMANCE HISTOGRAMS are also provided to give an indication of the relative frequency of memory accesses within a particular memory range, with the exact range specified by the user. This histogram provides a means of determining where a program spends the greatest amount of time. The resulting information can then be used to compress inefficient code. A hardware performance histogram is provided to display the relative frequency of combinations of four user-selected signals within a user system.

In order to service these triggering modes and provide a complete set of operating features, an MC6809 microprocessor is located on the Control Module with local intelligence running from an operating system based in ROM. This operating system provides the data analysis and formatting functions for the operator including the interface to the hardware sampling the bus.

To reduce system redundancy, the terminal used by the operator to communicate with the development system will also serve to link him to the intelligence aboard the analyzer (it is a requirement of the CRT-based analyzer operating system that the terminal used be an EXORterm 155 Display Console). This communication will be achieved by means of a phantom or transparent serial link feeding from the terminal through the Bus State Analyzer control board and then to the normal terminal input channel of the development system. The logic onboard the BSA determines whether the information traveling over the link is destined for the development system, the Bus State Analyzer or the terminal. This allows the operating system or the user's software to run in the development system while analysis is being performed. Additionally, a means is provided for the BSA to operate in a stand-alone mode with only a terminal connected.

Part Number	Description
M68BSAC	BSA Control Module for use with BSA Personality Modules
M68BSACE	BSA Control Module with Enclosure
M68BSA1-1	BSA Personality Module for MC68000, MC68010 and MC68451
M68BSA2	BSA Personality Module for MC6800, MC6809, and MC6829
M68BSA3	BSA Personality Module for MC68008
M68BSA4	BSA Personality Module for MC6801 and MC68120
M68BSA5	BSA Personality Module for VERSABus
M68BSA6	BSA Personality Module for EXORbus

MOTOROLA
MICROCOMPUTER DEVELOPMENT SYSTEMS



LOW COST MPU/MCU EVALUATION MODULES

A series of inexpensive evaluation modules are available for Motorola's line of microprocessors and microcomputers. Evaluation modules allow the user to prepare, debug, and run software in the resident microcomputer. Even though the cost is low, an onboard ROM contains extensive commands for controlling I/O and debug operations, including down-up load S-record transfers.

Memory, internal registers, and I/O registers may be displayed and modified. Program execution may be traced one step at a time or breakpoints may be inserted for program interruption. Circuitry and firmware are included to allow the MCU's EPROM to be programmed.

MC68000 Educational Computer Board MEX68KECB

The MC68000 Educational Computer Board (ECB) serves as an economical introduction to systems based on the M68000 family of microcomputer products.

The ECB is based around a 4 MHz MC68000 MPU. Also provided are 32K bytes of RAM, arranged as 16K x 16. The firmware is contained in two 8K by 8 ROMs, addressed as an 8K by 16 block of memory. Two RS-232C serial ports are implemented with MC6850 ACIA's and an MC14411 baud rate generator, allowing selection of data rates from 110 to 9600 baud.

One of the M68000 peripherals, the MC68230 Parallel Interface and Timer (PI/T), provides a Centronix-type parallel printer interface and an audio cassette interface. An audio cassette recorder may be used to store and retrieve user programs.

The ECB uses a terminal, interfaced via one RS-232C port. Also, a small wirewrap area is provided for system I/O modification and buffering.

MC6801 Evaluation Module MEX6801EVM

The MC6801 Microcomputer Evaluation Module is a completely self-contained microcomputer on a single printed circuit card, providing the user with the means of evaluating the MC6801 microcomputer. As configured, the MC6801 may be evaluated in the Single-Chip mode by attaching an RS-232C-compatible terminal to the serial port of the module. Thus, the minimum functioning system consists of only the MC6801 and an MC1488 and MC1489 (RS-232C interface).

In the Expanded mode, the customer may add an ACIA, PTM, 4K bytes RAM or 2K EPROM and a programmable gate array for address configuration.

MC68701 Programming Module MEX6801EV1

This module has the same features as the MC6801 module but is also populated with an MC68701, 2K bytes of RAM, a programmed gate array, and a DEbug monitor (PRObug) which also provides the programming capability for the MC68701 EPROM device. It, also, can be used to evaluate the MC6801 microcomputer.

MC68120 Evaluation Module M68120EVM

The M68120 Evaluation Module is designed to assist the potential user of an MC68120 Intelligent Peripheral Controller (IPC) chip in developing software, performing limited circuit emulation, and operating as a serially-linked design on an EXORbus compatible board format.

All data communications are accomplished via two RS-232C ports. Consequently, the Evaluation module can be operated in a stand-alone configuration with only power brought in on the EXORbus connector. An optional operating configuration allows the M68120 Evaluation Module to be plugged into an EXORciser II or an EXORmacs VERSAbus System via a VERSAbus Adapter Module (VAM). The dual-ported 128-byte RAM can then be mapped into a local map or system map.

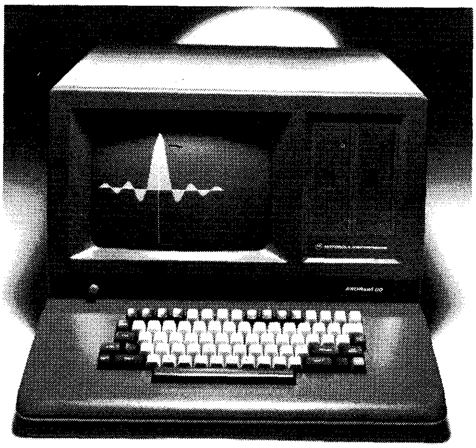
There is 4K of RAM populated on the board local bus along with decoding to permit an additional 4K RAM to be implemented in the user wirewrap area. The RAM allows user software development and debug for future programming of 2K, 4K, and 8K EPROMs to be inserted on the M68120 Evaluation Module. The Module has 64K bytes of address space on the local bus and 256 bytes of address space on the system bus.

MC68705 Evaluation Module M68705EVM
MC1468705 Evaluation Module M1468705EVM

Operation of an MC68705 or MC1468705 is simulated by the resident MC6805 or MC146805 MCU. Data transfer within the EVM is controlled by the monitor ROM firmware. In turn, this ROM is controlled from an external RS-232C compatible user terminal. User object code may be down-loaded to the user program RAM via the host port; a cassette port is also provided for this purpose. The host and terminal port ACIAs are baud rate strap-selectable from 110 bps to 19.2 Kbps in eight steps.

The MCU parallel I/O ports allow the user to connect externally to the simulated MCU I/O lines. These lines are also used to control the MC68705 or MC1468705 MCUs on-chip EPROM programmer. This is accomplished by inserting the MCU into the programmer socket and executing the appropriate monitor commands.

EXORset 110



EXORset 110 Features

- MC6809 high performance microprocessor.
- Full ASCII Keyboard with 16 user-definable function keys.
- 12" CRT displaying 22 lines of 80 characters, or switchable to 16 lines of 40 characters and/or full graphics. 2K bytes of static RAM are included for CRT character refresh.
- 56K bytes of RAM and three sockets for up to 24K bytes of EPROM/ROM.
- Three card slots for EXORciser/Micromodule boards, four if no disk controller needed.
- Printer interface.
- Serial I/O port.
- EXORbug monitor/debug ROM included. An additional EPROM/ROM socket is available if user does not require EXORbug.
- Triple 16-bit programmable counter/timer included with input Gate and Clock signals and output signals available to the user.
- Meets FCC compliance for a Class A computing device.

A High Performance Processor

The EXORset controller is based on the new generation 16/8-bit microprocessor MC6809. The expanded instruction set, addressing modes, and architecture make execution of software particularly efficient and allow sophisticated programming techniques such as structured programming, position independent code, re-entrant routines and real-time operations. These capabilities make the 6809 microprocessor suitable for high-level language program development.

A CRT Display and Keyboard

The EXORset unit provides the user with a complete man/machine interface consisting of a full-size ASCII keyboard and 16 user-assigned function keys and a high resolution 12" CRT display capable of displaying 22 lines of 80 or 16 lines of 40 upper or lower case characters and simultaneously a full 320 x 256 dot graphic image.

Memory Flexibility

The EXORset controller allows for flexibility in the type and amount of memory to be used in the application. Three versions are available that provide optional amounts of mass storage: no floppy disk drives, with one double-sided mini-floppy disk drive for 160K bytes of mass storage and with two disk drives for 320K bytes of mass storage. All three versions include 2K bytes of dynamic RAM for CRT character refresh, 56K bytes of dynamic RAM and three strappable sockets that can be configured for 1K, 2K, 4K or 8K ROMs or EPROMs. A fourth socket, normally containing the 4K EXORbus firmware, can be configured for a user-designed monitor routine. The EXORset memory map is defined by PROMs, allowing the user to easily reconfigure the architecture of the system. Optional configurations information is available by contacting your local Motorola sales office.

On-Board Input/Output Ports

The EXORset unit provides three on-board input/output ports. An asynchronous serial communication port is provided with strap-selectable interface options of RS-232C, RS-422 or RS-423 and can be configured as a terminal or as a modem. The baud rate is software programmable from 110 to 19.2K baud. The user may also replace the asynchronous device with an SSDA device for synchronous communication application.

A 16-bit data plus four handshake control lines parallel input/output port is provided. This parallel port consists of a fully-buffered PIA device with a pinout that is compatible with a standard Centronics printer type interface. An optional adapter kit is available to interconnect this port to the industry standard optically-isolated solid-state relay mounting racks.

A triple 16-bit programmable counter/timer device is included, with each section's clock, gate and output signals available to the user. The output signals can be strapped to generate a system IRQ, FIRQ or NMI if required.

Add-On Input/Output Flexibility

The EXORset Controller has a four-slot card cage with bus connectors for installing additional EXORbus compatible modules available from Motorola as well as a number of other vendors. The Floppy Disk Controller Module occupies one of these four slots.

Development Systems

EXORciser For 8-Bit Prototype Development

M6800 EXOR
M6809 EXOR

The EXORciser is an expandable development system that allows development of any 8-bit Motorola microprocessor or microcomputer configuration, from the simplest to the most elaborate. It comes with an MPU Module that provides system timing and a DEbug Module that contains system firmware. Both MC68B00 or MC68B09 MPU versions are offered in the EXORciser Development System.

With optional accessories, the EXORciser design and diagnostic functions can be extended to other members of the Motorola family of microprocessors and microcomputers.

The EXORciser with a USE (User System Emulator) option can be used to test and evaluate equipment external to its chassis. By removing the microprocessing unit from the user's system and connecting the USE cable from the EXORciser into the MPU's socket, the EXORciser with its EXbug firmware can be used to debug and troubleshoot microprocessor systems.

The basic EXORciser consists of a rugged cabinet with a built-in power supply, and a prewired bus-oriented 14-slot Backplane with MPU and DEbug Modules. Together these elements form a development microcomputer, with the capability of adapting the unit to a specific design problem by adding optional I/O and memory modules. Additional Motorola memory modules for the EXORciser can be selected to suit varying system configurations; for example, to meet the increased memory requirements of sophisticated high order language based systems. The concept of add-on modules permits the user to match the functional requirements of the systems being developed. Using one slot each for a floppy disk and printer function, ten slots remain for memory and I/O expansion. The EXORciser is a system that is never out-of-date, being at all times upgradable when new and expanded microcomputer functions become available.

Accessories for EXORciser

PROM Programmer — M68PP5

The PROM Programmer is designed to program a variety of MOS PROMs, EPROMs and bipolar PROMs. It can verify data from the PROM, transfer data from the PROM to the development system RAM memory, and transfer blocks of data from one memory location to another. Programming time depends upon the PROM used.

The M68PP5 is a powerful new EEPROM/PROM/EPROM programmer, designed to provide all of the functionality of the M68PP3 and more. A powerful feature of the M68PP5 is that it does not require removal of the EXORset or EXORciser covers during operation. This is accomplished via the Remote Socket Module. This module can be conveniently positioned by the user for his needs. It also has many other new or improved features. Such features include: programming even or odd byte PROMs/EPROMs, commands to display and modify data, attach printer command to send all responses to a hard copy printer. The power of the M68PP5 is further enhanced by the increased list of standard devices which it can program.

Software on diskette for both M6800 and M6809-based systems is included with the PROM Programmer.

System Analyzer — MEX68SA2 (6800) M6809SA (6809)

This unique instrument can be used to enhance the capabilities of the EXORciser as a design tool, or as an independent, portable, low-cost unit for field service of bus-compatible equipment.

In field service applications, the System Analyzer derives operating power and I/O signals directly from the system under test. It can stop the system at any point in its program, step through the program, change the contents of the system memory, and monitor and record the MPU's operation during a selected portion of the program without shutting down the operation.

In EXORciser applications, it complements the system's inherent program development capabilities. In conjunction with the EXORciser and USE, it offers a powerful combination of development and diagnostic tools available for microcomputer work.

MC6801 Development System — MEX6801

This product upgrades EXORciser and EXORset for development of MC6801-based systems. All three modes of MC6801 operation — single-chip, expanded multiplexed and expanded non-multiplexed — are supported by this system.

This system is fully compatible with all current supporting hardware and software and includes the USE function. It allows real-time emulation of the MC6801 application hardware and facilitates the debugging of software.

Resident System Software

8-Bit Assemblers, Editors and Monitors

M6800 and M6809 Development System Software Package

Supplied with the Motorola floppy disk subsystem, EXORdisk, is a basic software development package consisting of the Motorola Disk Operating System (MDOS), CRT Editor, Macro Assembler and Linking Loader.

M6800/01/09 UP/Down Load — M6800UPDWNLD, M6809UPDWNLD

Permits a user to download software developed in a host system into an EXORciser or Micromodule; alternatively, memory-to-memory uploads are permitted between EXORcisers, and a memory-to-file upload to an EXORmacs; provided in both a 2K PROM and two 1K PROMS.

8-Bit High Level Languages and Cross Assemblers

M6800/M6809 MPL Compiler — M68MPLR020M/ M6809MPL

A high-level, user-oriented system programming language for the MC6800 and MC6809 MPU's, MPL is a block-structured language with features chosen for applicability to the microprocessor environment. This compiler is designed to operate in an EXORterm or EXORciser floppy disk-based environment with MDOS.

M6800/M6809 FORTRAN — M68FTNR012M/ M6809FORTRN

Resident FORTRAN is a high-level programming language widely used for scientific and engineering problem solving. This FORTRAN Compiler, which is a subset of the ANSI standard FORTRAN IV, translates the source program into a relocatable object module. The Linking Loader converts the relocatable object code into an executable object file.

M6800 Real-Time FORTRAN Compiler — M68RTFR02M

This FORTRAN Compiler enables the user to write real-time software in a high-level language for use in M6800-based Micromodule systems. It also contains an execution-time operating system, allowing several queues of tasks to be performed, with an ability to respond to real-time interrupts and to generate delays.

M6800 Resident BASIC Interpreter — M68BASR010M

The Resident BASIC Interpreter provides another problem-solving tool to the M6800 microcomputer family of products. BASIC is high-level programming language widely-used for education, general-purpose, and certain business-related applications. Decimal arithmetic, string variables and arrays, string functions, and printer output are several of the features.

M6809 BASIC-M Interactive Compiler — M6809BASMR

The BASIC-M Interactive Compiler provides an extension over standard BASIC in two major directions. It improves considerably the capabilities of the BASIC programming language and generates executable codes that can be used independently of the compiler itself. The compiler is available for M6809-based EXORciser.

M6809 Pascal — M6809PASCLC

M6809 Pascal produces relocatable object modules that may be linked with other separately compiled Pascal modules and/or assembly language modules. The object code is position independent, re-entrant and ROMable. Both a Compiler which produces a relocatable object module, and an Interpreter version are available for M6809-based EXORciser systems.

EXORciser CRT Editor — M68EDITM

EDITM is a memory resident record key oriented text editor that can do CRT editing on a line-by-line basis using only the left and right cursor functions and a few easy to remember control key sequences. EDITM can edit up thru a 132 character line, can be run under CHAIN command control, has error recovery procedures, and is USER configurable for different CRTs and default conditions. 6800 and 6809 versions of EDITM are included. 32K RAM minimum.

EXORset/EXORciser Document Processor — M68XDOC/ M68MDOC

DOC is a powerful text processing program. Any editor may be used to imbed the DOC processor commands with DOC interprets and formats in the output text. Among the many features are: file concatenation (book chapters), multiple file input (form letter/address file), automatic table of contents generation, automatic page numbering, left/right/ center text justification, conditional text, and multiple line spacing. 24K RAM minimum.

Peripherals

Use of appropriate peripheral devices can generate savings by affording faster program development. Each Motorola peripheral is supplied with the necessary circuitry to perform the necessary development system interface function.

EXORterm 155 — M68SXD10155A

EXORterm 155 is a video terminal which facilitates the exchange of data between the user and the development system via a high quality video interface in combination with keyboard entry and a serial communications link using speeds up to 9600 baud.

EXORterm 155 uses LSI components of the M6800 family to provide control of the display attributes, communication facility, terminal switch/indicator control, and keyboard inputs. The keyboard provides cursor control keys and special keys to invoke functions unique to the EXORciser and EXORmacs Operating Systems. These keys can also be used by the designer for special routines. An additional Text Edit mode feature permits multiuser editing.

EXORterm 155 may be connected for either RS-232C or 20/60 mA current loop operation. Like the EXORterm Development System, this display console contains a high-quality CRT with a full 1920-character screen and 7 x 9 ASCII characters.

The EXORdisk — M68DSK2, M68DSK3, M68KFD1102

The EXORdisk is a dual floppy disk storage system with its own package of development software. EXORdisk is designed to support either MDOS (the EXORciser Disk Operating System) or VERSAdos (the EXORmacs Disk Operating System). It facilitates high-speed data transfers through fast headsettling time and logical sector arrangement. An interface card connects this storage system to the EXORmacs, EXORciser or EXORterm Development Systems.

EXORdisk is available in various storage capacities. EXORdisk II offers 512K bytes of storage. It is a single-sided/single-density dual drive system with up to 256K bytes of memory per diskette. EXORdisk III is a double-sided/single-density dual drive system with total storage of 1 million bytes. An expansion unit is available for EXORdisk III which adds one additional disk drive and interconnecting cable to increase storage to 2 million bytes.

Hard Disk — M68KHDS16-1, M68KHDS32-1, M68KHDS50-1, M68KHDS96-1

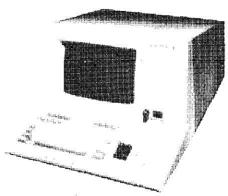
The longer, more complex programs written for advanced 16-bit processors like the MC68000 make much higher data transfer speed and larger storage capacity a necessity. New Hard Disk systems offer the EXORmacs user a choice of high speed mass storage.

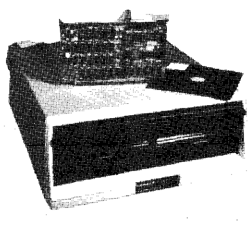
For multiuser operation in the EXORmacs system, Hard Disk is required to provide rapid storage and retrieval for a large number of files. Hard Disk storage greatly enhances and increases file access performance over a floppy disk-based system.

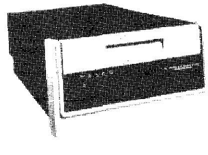
Dot-Matrix Printers — M68SP702C10, MPRINT703

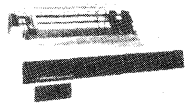
Motorola dot-matrix printers are equipped with an interface module and/or an interconnection cable assembly that specifically adapts them to the various Motorola microcomputer development systems, including the EXORmacs, EXORciser, EXORterm and EXORset. In addition these interface accessories permit the printers to be used with Motorola Micromodules to provide more complete availability of microcomputer system components. Printer specifications are as follows:

FUNCTION	703	702
Print Speed (cps)	180	120
Lines-per-Minute (80 characters)	90	65
Bidirectional Printing	Yes	Yes
Dot-Matrix	7 x 7	7 x 7
ASCII Character Set	96	64
Tractor Feed	Yes	Yes
Condensed Print (10–16.5 cpi)	—	—









REFERENCE GUIDE: Selection by MPU/MCU Supported

PRODUCT CATEGORY: EXORmacs (68000 only)

Type Number	Description
M68KVM10-3	VERSAbus RAM 128K Bytes
M68KVM11-1	VERSAbus RAM 256K Bytes
M68KVM11-2	VERSAbus RAM 512K Bytes
M68KHDD16-1	16 MB Hard Disk
M68KHDD32-1	32 MB Hard Disk
M68KHDD50-1	50 MB Hard Disk
M68KHDD96-1	96 MB Hard Disk
M68KMCCM	Multichannel Communications Module
M68KEXTM	VERSAbus Extender Module
M68KFD1102	EXORdisk III for EXORmacs
M68KVAM	VERSAbus Adapter Module
M68KWW	VERSAbus Wirewrap Module
M68K703LP1	EXORmacs Printer 703, 110 V
M68KRDS1	EXORmacs Remote Development Station with USE
M68KRDS2	EXORmacs Remote Development Station without USE
M68KMACSRK	EXORmacs Rack Mount Kit

PRODUCT CATEGORY: Systems Products

Type Number	6800	6801	6809	6805	68000	68010	Description
M68K101-1						X	VME/10 Microcomputer System — 5 MB
M68K102B1						X	VME/10 Microcomputer System — 15 MB
MVMC682-114					X		VMC 68/2 Microcomputer System
M6809SET110	X	X	X	X			EXORset Microcomputer System

PRODUCT CATEGORY: Instrumentation

Type Number	68000	68008	68010	68120	6804	6805	146805	68705	6800 Family	Description
M68KHDS400	X	X	X							HDS-400 Control Station
M68KHDS16FB	X	X	X							HDS-400 Personality Module
M68000HDS4	X									MC68000 Emulator Module
M68008HDS4-8		X								MC68008 Emulator Module
M68010HDS4-8			X							MC68010 Emulator Module
M68HDS201					X	X	X	X		HDS-200 Control Station
M6804P2HM					X					MC6804P2 Emulator Module
M6805P234HM						X		X		MC6805P2,P4,P6, MC68705P3,P5 Emulator Module
MC6805RU234HM						X		X		MC6805R2,R3,U2,U3, MC68705R3,U3 Emulator Module
M6805S2HM						X				MC6805S2 Emulator Module
M6805T2HM						X				MC6805T2 Emulator Module
M146805E2HM							X			MC146805E2 Emulator Module
M146805F2HM							X			MC146805F2, M1468705F2 Emulator Module
M146805G2HM						X		X		MC146805G2, M1468705G2 Emulator Module
M68BSAC	X									Bus State Analyzer Control Module
M68BSA1-1	X									MC68000 BSA Personality module
M68BSA2									X	M6800 Family Personality Module
M68BSA4				X					X	MC6801, MC68121 Personality Module
M68BSA5	X	X	X							VERSAbus Personality Module
M68BSA6									X	EXORbus Personality Module
MEX6801EVM									X	MC6801 Evaluation Module
M68120EVM				X						MC68120 Evaluation Module
M68705EVM							X			MC68705 Evaluation Module
M1468705EVM							X			MC1468705 Evaluation Module

PRODUCT CATEGORY: EXORciser

Type Number	141000	68000	6801	6809	6805	6800/02	Description
MEX68IC2				X	X	X	I/O Interconnect Cable (Use with MEX6821-2)
MEX68RK2		X		X	X	X	Rack Mounting Kit EXORciser I & II
MEX68RR		X		X	X	X	EPROM/RAM Module
MEX68SA						X	System Analyzer
MEX68SA2						X	System Analyzer II
MEX68USEC						X	User System Evaluator
MEX68USM				X		X	Universal Support Module
MEX68WW				X	X	X	Wirewrap Module
MEX68XT3				X	X	X	Extender Module
MEX6801EVM			X				Evaluation Module
MEX6801EVM1			X				68701 Programming Module
MEX6801			X				Development System
MEX6802-46						X	MC6802/46 Support Module
MEX6808-22		X	X	X	X	X	8K Static RAM Module with Parity
MEX6809KT				X			6809 Upgrade for EXORciser or EXORterm
MEX6812-1		X	X	X	X	X	2K Static RAM Module
MEX6816-1HR		X	X	X	X	X	16K Dynamic RAM Module with Hidden Refresh
MEX6816-22D		X	X	X	X	X	16K Dynamic RAM Module with Parity
MEX6816-22S		X	X	X	X	X	16K Static RAM Module with Parity
MEX6821-2		X	X	X	X	X	Input/Output II Module
MEX6832-1HR		X	X	X	X	X	32K Dynamic RAM Module with Hidden Refresh
MEX6832-22		X	X	X	X	X	32K Dynamic RAM Module with Parity
MEX6845		X	X	X	X	X	MC6845 CRT Controller Module
MEX6848-1HR		X	X	X	X	X	48K Dynamic RAM Module with Hidden Refresh
MEX6848-22		X	X	X	X	X	48K Dynamic RAM Module with Parity
MEX6850		X	X	X	X	X	ACIA Module
MEX6850-2		X	X	X	X	X	ACIA/SSDA Module
MEX6854		X	X	X	X	X	MC6854 ADLC Support Module
MEX6864-1HR		X	X	X	X	X	64K Dynamic RAM Module with Hidden Refresh
MEX6864-22		X	X	X	X	X	64 Dynamic RAM Memory with Parity
MEX68488		X	X	X	X	X	MC68488 GPIA Support Module
M68BASR010M						X	Resident BASIC Interpreter on 6800 MDOS Diskette
M68FTNR012M						X	Resident FORTRAN Compiler and Linking Loader on 6800 MDOS Diskette
M68MPLR020M						X	Resident MPL Compiler on 6800 MDOS Diskette
M68PANEL220		X	X	X	X	X	6809 Front Panel Conv. of EXORterm 200
M68PP3		X	X	X		X	PROM Programmer III
M68PP3-1		X	X	X		X	Personality Module & Software for PPIII to allow Programming of MCM2532 and MCM68764
M68RTFR02M						X	Resident Real-Time FORTRAN Compiler on MDOS Diskette for 6800
M6800DOWNLD			X			X	6800/6801 Down-Line-Load ROM
M6800EXOR						X	M6800 EXORciser II Development
M6800EXORU						X	M6800 EXORciser II USE Development System 110 V
M6800SMDOS						X	6800 CRT Editor/Macro Assembler with MDOS
M6800XASMBL1			X			X	6800/6801 Cross Macro Assembler
M6805MASC01M					X		6805 Cross Macro Assembler and Linking Loader on MDOS Diskette
M6809DOWNLD				X			6809 Down-Line-Load ROM
M6809EXOR				X			M6809 EXORciser II Development System 110 V
M6809FORTRN				X			6809 Resident FORTRAN Compiler
M6809MASC01M				X			6809 Cross Macro Assembler and Linking Loader on MDOS Diskette
M6809MPL				X			6809 Resident MPL Compiler on MDOS Diskette
M6809PASCLC				X			6809 Resident PASCAL Compiler
M6809SA				X			System Analyzer II
M6809SMDOS				X			6809 CRT Editor/Macro Assembler with MDOS
M6809USE				X			User System Evaluator
M6833		X	X	X	X	X	Blank Diskettes (SS/SD)
M6834		X	X	X	X	X	Blank Diskette (DS/SD)

PRODUCT CATEGORY: PERIPHERALS

Type Number	68000	6809	6805	6802	6800	Description
M68DSK2		X	X	X	X	EXORdisk II 110 V
M68DSK3		X	X	X	X	EXORdisk III 110 V
M68SFDRK3		X	X	X	X	Rack Mounting Kit, EXORdisk II and III
M68SFDU1102E		X	X	X	X	EXORdisk IIIE Expansion Unit, 110 V
M68SP702C10		X	X	X	X	Microsystems Printer 702, 110 V
MPRINT703	X	X	X	X	X	Microsystems Printer 703, 110 V
M68SXD10155A	X	X	X	X	X	EXORterm 155
M68KHDS32-1	X					32MB Hard Disk
M68KHDS96-1	X					96MB Hard Disk
M68KHDE32-1	X					32MB Hard Disk Expansion
M68KHDE96-1	X					96MB Hard Disk Expansion
M68CART	X					Hard Disk Cartridge

PRODUCT CATEGORY: VMEmodules (68000 family)

Type Number	Description
MVME101	68000 Monoboard Microcomputer
MVME110	68000 Monoboard Microcomputer with I/O Channel Interface
MVME200/201	64K and 256K Byte Dynamic Memory
MVME210	Static RAM/ROM Board
MVME300/310	GPIB Controller Modules
MVME310	Universal Intelligent Peripheral Controller
MVME315	Intelligent DMA SASI Interface and Floppy Disk Controller
MVME930	VMEbus Extender Board
MVME931	VMEbus Wirewrap Board

PRODUCT CATEGORY: VERSAmodules (68000 family)

Type Number	Description
M68KORMS68K	M68000 Real-Time Multitasking, Software (Object) on EXORmacs Diskette
M68KVM01A1	68000 16-Bit Monoboard Microcomputer, 32K RAM
M68KVM01A2	68000 16-Bit Monoboard Microcomputer, 64K RAM
M68KVM02	68000 16-Bit Monoboard Microcomputer, 128K RAM
M68KVM03	68010 16-Bit Monoboard Microcomputer, 10 MHz, 256K RAM
M68KVMCC1	4-Slot Card Cage
M68KVMCH1-1	VERSAmodule System Chassis, 15 Amps-5 Vdc, 110 V
M68KVM10-3	128K Byte Dynamic RAM Module
M68KVM11	256/512K Byte Dynamic RAM
M68KVM20	Floppy Disk Controller Module
M68KVM21	Universal Disk Controller
M68KVM30	4-Channel Serial Communication Module
M68KVM60	Universal Intelligent Peripheral Controller Module
M68KVBUG	VERSAbug Debug Monitor Firmware Package

PRODUCT CATEGORY: MICROMODULES

Type Number	6809	6802	6800	Description
M68BASRC1			X	Resident BASIC Interpreter ROM Set (MINIBUG II-Based)
M68BASRC2			X	Resident BASIC Interpreter ROM SEt (MICRObug-Based)
M68BASRM2			X	Resident BASIC Interpreter Module (Micromodules)
M68EAB1			X	Resident Editor/Assembler and BASIC Interpreter Module (MINIBUG II-Based)
M68EAB2			X	Resident Editor/Assembler and BASIC Interpreter Module (Micromodules)
M68MMCC05	X	X	X	Card Cage, 5-Card
M68MMCC10	X	X	X	Card Cage, 10-Card
M68MMFLC1	X	X	X	Front Load Chassis, 14 Card, 110 V
M68MMFLK	X	X	X	Rack Mounting Slide Kit, FLC
M68MMLC1	X	X	X	Long Chassis, 10-Card, 110 V
M68MMLK	X	X	X	Rack Mounting Kit, Long Chassis
M68MMPS1-1	X	X	X	Micromodule, EXORciser, EXORterm, DC Power Supply, 110 V
M68MM01A2			X	Monoboard Microcomputer (with four 2K x 8 EPROM/ROM Sockets)
M68MM01B1A		X		Monoboard Microcomputer
M68MM01D			X	Monoboard Microcomputer
M68MM03	X	X	X	32/32 Input/Output Module
M68MM03-1	X	X	X	32/32 Input/Output Module (with 4.7K Termination Option)
M68MM03-2	X	X	X	32/32 Input/Output Module (with 330/220 Termination Option)
M68MM04A	X	X	X	16 Socket EPROM, ROM or RAM Module
M68MM05A	X	X	X	8-Channel, 12-Bit Differential Input A/D Module
M68MM05B	X	X	X	16-Channel, 12-Bit Single Ended Input A/D Module
M68MM05C	X	X	X	Quad 12-Bit D/A Module
M68MM07	X	X	X	Quad Communication Module
M68MM08A			X	MICRObug Module-Consisting of MICRObug ROM (Use with MM01A2)
M68MM09	X	X	X	4K CMOS RAM with Battery Backup
M68MM10B	X	X	X	Power Fail Detect Module with Battery Backed-up CMOS Time-of-Day Clock/Calendar
M68MM11	X	X	X	RS-232C to TTY Adapter Module
M68MM12		X	X	GPIB Listener/Talker/Controller Module (with 6800 Firmware)
M68MM12-1	X			GPIB Listener/Talker/Controller Module (with 6809 Firmware)
M68MM12A	X	X	X	GPIB Listener/Talker Module
M68MM13A	X	X	X	Digital-Output (Contact Closure) Module — 16 Outputs
M68MM13B	X	X	X	Digital-Output (Contact Closures) Modules — 32 Outputs
M68MM13C	X	X	X	Optically Isolated Digital Input Module-24 Voltage Inputs
M68MM13D	X	X	X	Optically Isolated Digital Input Module-24 Contact Closure Inputs
M68MM14	X	X	X	2 MHz Hardware Arithmetic Processor Unit
M68MM14A	X	X	X	3 MHz Hardware Arithmetic Processor Unit
M68MM15A	X	X	X	High-Level A/D Module 16 Channel
M68MM15A1	X	X	X	High-Level A/D Module 32 Channel
M68MM15B	X	X	X	Low-Level A/D Module
M68MM15CV4	X	X	X	High-Level Voltage D/A Module 4 Channel
M68MM15CI4	X	X	X	Current D/A Module 4 Channel
M68MM16	X	X	X	Combo ROM, RAM and I/O (Parallel and Serial) (1 or 2 MHz)
M68MM17	X			6809 Monoboard Microcomputer
M68MM19A	X			6809 Monoboard Microcomputer (2 MHz) (For new designs use MM19A1, up to 32K EPROM)
M68MM19SB	X			SUPERbug Firmware ROM
M68MMI/OC	X	X	X	Parallel I/O Adapter Set
M68XEARC1		X	X	Resident Editor/Assembler ROM Set (MINIbug II/MICRObug-Based)

VMEmodules

VMEmodules from Motorola incorporate the high performance MC68000, the internationally accepted Eurocard format, the defacto industry standard 16-bit VMEbus, and the new and flexible I/O Channel, all combined in the most versatile and latest state-of-the-art approach to the modular systems concept.

The MC68000 MPU

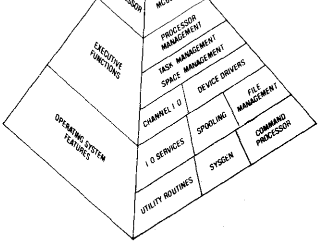
You've seen the benchmarks, and the results — MC68000 has emerged as the acknowledged microprocessor leader in the 16/32 bit performance class. Its architecture is designed for optimal support of the latest high-level languages, and it directly addresses 16 Megabytes of memory (instead of one Megabyte for most of the competition). Its 32-bit internal features mean easy growth to full 32-bit capability as your needs grow into the future. VMEmodule products put the MC68000 MPU to work in a modular structure that has achieved worldwide acceptance and support, both by users and manufacturers of microcomputer subsystems.

Worldwide Standard Package: Eurocard

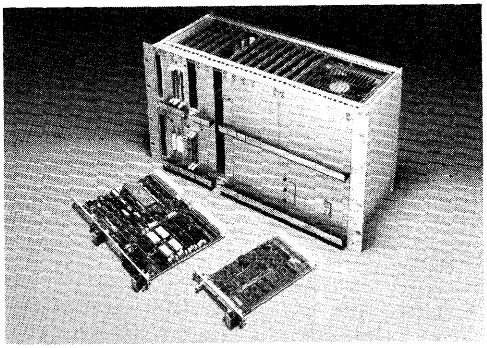
Developed as a de facto standard in Europe, the Eurocard mechanical format is rapidly gaining worldwide acceptance of modular applications in a broad range of laboratory and industrial automation environments. And for good reason — the Single and Double Eurocard circuit boards and card cages in the VMEmodule product line offer a convenient size, plus pin-and-socket bus connectors to give you an extra margin of confidence of reliability in the more severe application environments.

Multiprocessing 16/32 Bit VME Bus

The VME bus doesn't lock you into today's technology. It has the inherent power and capabilities to adapt to any number or types of popular processors for true multiprocessing applications; and, you can use as many bus masters as you need.



BASIC-M, I/Omodule, RMS68K, VMEbus, VERSAbug, VERSAdos, VERSAmodule and VMEmodule are trademarks of Motorola Inc.



With the VME bus, you can mix 8, 16, and 32-bit processors in the VME backplane. It operates asynchronously at high speed, and provides 7 interrupt plus 4 bus arbitration priority levels to allow total flexibility.

I/O Channel Expands Capabilities

The VMEmodule system architecture supports the I/O Channel feature described elsewhere in this publication. Briefly, the I/O Channel is a buffered extension of the onboard processor bus, allowing the system to be easily custom-tailored with the addition of input/output functions in small modular amounts both within and external to the VME card cage. The I/O Channel promotes efficient system utilization by allowing I/O transfers to proceed at rates up to 2 megabytes per second, independently of other on-going activity in the higher-speed VMEbus system interconnect.

Powered by High Performance Software

VMEmodule products are designed for demanding lab and industrial automation environments where quick, accurate response to multiple random events is essential — and Motorola's RMS68K Real-Time Multitasking Executive software for the VMEmodule Monoboard Microcomputer provides the nucleus around which complete real-time applications can be built. For those applications where large data files and mass storage resources must be handled efficiently, there's the full-featured VERSAdos Operating System. Standard device drivers are provided with both VERSAdos and RMS68K for interfaces and devices supplied by Motorola, and both systems make provisions for easy addition of user-supplied device drivers. Both the RMS68K Executive and the full VERSAdos System are rapidly emerging as the standard real-time system structure for MC68000-based applications.

To provide diversified programming capabilities for VMEmodule-based projects, Motorola supplies not only an advanced Structured Macro Assembler, but also efficient Pascal and FORTRAN Compilers.

VMEmodules

And to offer streamlined debugging capabilities, the VMEbug Debug Monitor firmware is available either in ROM or on disk for use with the VMEmodule Monoboard Microcomputer.

Modular Subsystems elevate the starting point for microcomputer system design from the "components" level to the board level. And, just as there are variations in microprocessors for different end-use requirements, there are families of modular subsystems to best serve these varying demands. Thus, the VMEmodule family joins the existing Motorola Micromodule 8-bit family and the VERSAmodule 16-bit family of modular microsystem products to let the user tailor his system to his specific needs.

VMEmodules provide a degree of performance and flexibility that bridges the gap between the lower-level 8-bit processing tasks (the Micromodule domain) and the highend computation and memory-intensive challenges that are the domain of the physically larger and more complex 16/32-bit VERSAmodules. This spectrum of microsystem products offers the most cost effective solution to complex systems — perhaps distributed control systems — with the right performance elements at each processing node of the system.

The Intangible Extras —

When you select Motorola microsystem products for your system design, you get not just the hardware and software, but a host of built-in benefits of almost equal importance. Among these:

- A field-proven line of thoroughly tested products that assure highly reliable system operation.
- A time-tested set of support tools and documentation that simplify system design and operation.
- A nationwide field-sales and service network that offers design and applications support before, during and after the sale.

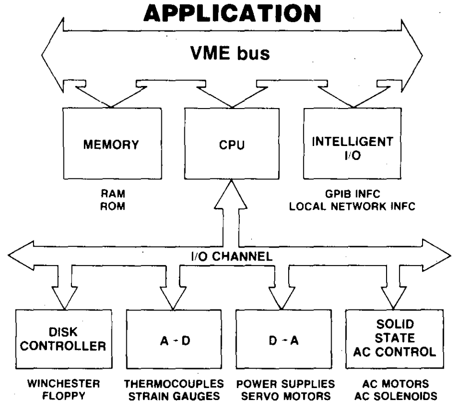
A mature training program at various levels that offers group training at specified locations as well as in the customer's own establishment.

A product line that continues to expand to take full advantage of new developments for increasing capabilities, improving performance and allowing more efficient operation.

Multiple Sources of VME Compatible Products — Worldwide

Development of the VME bus structure represents the combined technical efforts of Motorola and a number of other major international electronics companies. The initial announcement in Europe met with very positive reactions from potential users and vendors the world over, with the result that the original participants are being joined by increasig numbers of companies planning to supply such products. These sources are united through the activities of the VME Bus Manufacturers Group, which meets four times a year in technical forum to help assure the user community of a high degree of technical compatibility between products, and to make available to the public a comprehensive list of suppliers.

TYPICAL VMEmodule APPLICATION



VMEmodule Line*

VMEmodules — VMEbus compatible, Double Eurocard Format.

MVME101 — MC68000 Monoboard Microcomputer with two serial ports and one parallel port on board.

MVME110 — MC68000 Monoboard Microcomputer with I/O Channel support for extended I/O functions.

MVME200/201 — 64K byte and 256K byte Dynamic RAM Modules with data parity check.

MVME210 — Static RAM/ROM Board providing up to 128K bytes storage capacity.

MVME300 — GPIB Controller meeting full IEEE 488-1978 standard.

MVME310 — Universal Intelligent Peripheral Controller with 35% of board area in wirewrap for customer applications.

MVME315 — Intelligent DMA SASI interface and floppy disk controller.

MVME930 — VMEbus Extender Board

MVME931 — VMEbus Wirewrap Board

*See also the list of I/O modules on another page in this catalog for additional I/O functional elements supporting the VMEmodule line.

Software

MVMEBUG — Debugging Packages for VMEmodule Monoboard Microcomputer with single-line Assembler/Disassembler.

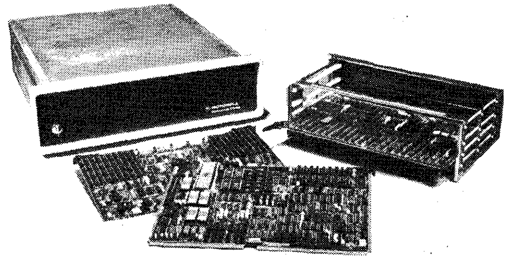
M68KORMS68K — M68000 Real-Time Multitasking Executive provides task scheduling and synchronization for any number of tasks.

M68KOVDOS — OEM VERSAdos Operating System is a real-time multitasking MC68000 based system oriented to hard disk operation.

Packaging

VMEmodule and I/Omodule Card Cages, Chassis, Power Supplies and Backplanes.

VERSAmodules



VERSAmodule circuit boards are microcomputer building blocks from Motorola, based on the state-of-the-art 16-bit MC68000 Microprocessor. They are part of a family of modular building block products that provide the system designer ready-to-run hardware and software. VERSAmodule building blocks drastically reduce the total cost of bringing together a fully configured custom microcomputer-based system . . . by saving development time, engineering talent, and money as well.

With VERSAmodule products, you minimize the risks of design limitations and system obsolescence while keeping your system tied to the leading edge of technology. Your system is built around the most advanced 16-bit microcomputer available today . . . incorporating sophisticated architectural features to enhance system performance. The full range of available software products and applications development tools assure early system completion. And Motorola's experienced support staff is available to help, any time.

Use Today's Most Advanced 16-Bit Microcomputers

The VERSAmodule Monoboard Microcomputers (VM01A and VM02) are the most powerful and versatile 16-bit single-board microcomputers available. They achieve a higher degree of computing power, memory capacity and tailorability by combining the MC68000 MPU with other on-board features. Such on-board features as I/O Channel interface, VERSAbus interface, bus arbitration logic, dual port RAM, multiprotocol serial I/O, parallel I/O, programmable timer/counters, and RAM with battery back-up capability enable these VERSAmodule Monoboards to handle applications ranging from those using a single processor through those requiring complex multiprocessing structures.

VERSAbus Architecture Enhances System Performance

VERSAmodule boards are interconnected in a system using the VERSAbus interconnect standard. The high-speed VERSAbus interconnect is characterized by asynchronous operation supporting direct memory addressing and true multiprocessor operation. Unlike other popular bus structures, VERSAbus architecture does not limit the number or types of processors that can be used in multiprocessing applications. The number of "bus masters" or main processor boards is limited only by the number of card slots in the particular VERSAbus backplane being used. Furthermore, several lines within the VERSAbus structure enhance system reliability and integrity by providing for efficient self-diagnosis . . . resulting in minimum system downtime.

BASIC-M, I/Omodule, RMS68K, EXORmacs, EXORbus, VERSAmodule, VERSAbus, VERSAdos, and VERSAbug are trademarks of Motorola, Inc.

Cost-Effective I/O Channel Increases System Flexibility

The I/O Channel is an advanced architectural feature of VERSAmodule Systems that allows greater system flexibility and low incremental cost for I/O expansion. The I/O Channel has a 12-bit address bus, 8-bit bidirectional data bus, 4K Bytes of memory-mapped I/O, and a data transfer rate of up to 2 Megabytes per second.

VERSAdos Real-Time Disk Operating System

The VERSAdos Operating System Software employs modular design of its major programs to allow easy addition of user functions with minimal cost. It contains a file management package and additional device-independent I/O support. The VERSAdos System is available with software drivers for both floppy and hard disk storage, and incorporates redundant safeguards against system failures. Optimum processor and memory utilization are achieved through true multitasking and dynamic memory allocation/deallocation.

RMS68K Real-Time Multitasking Executive

For real-time applications that do not require auxiliary mass storage (disk), and efficient Real-Time Executive may provide all the required systems functions.

The RMS68K Real-Time Multitasking Executive provides the nucleus around which real-time applications can be built. It allows a wide variety of application systems without large expenditures for complex real-time and multitask control functions. RMS68K is ROMable, meaning that the executable code for your entire system could be placed in ROM. In addition, the RMS68K System customizes your system by allowing you to add your own device drivers and select only those functions that you need. Compatibility with VERSAdos and debug software packages helps reduce the cost of software maintenance over the life of your system.

VERSAbug Debugging Packages

The VERSAbug debug package provides a powerful evaluation and debugging tool for VERSAmodule Systems. It permits full-speed execution of system and user-developed programs operated in a VERSAmodule Monoboard Microcomputer environment under complete operator control.

VERSAbug software is available as a system debug monitor, in a pre-configured EPROM resident package, or as source and relocatable object modules, packaged on diskette or cartridge disk, allowing you to easily create your own application-specific version in a matter of hours. In either package, VERSAbug software gives you a powerful tool for reducing system development and continuing maintenance costs.

VERSAmodules

Complete Your System . . . On Schedule

With VERSAmodule products, the lion's share of your system's hardware design, debug, assembly and test is done for you. The mature operating system software is already developed and debugged, too. You can begin developing your applications software immediately, in order to respond faster to customer requirements, penetrate fast moving market windows, or automate a critical activity sooner. The result . . . higher profitability.

Use Your Resources Efficiently

Since your costly and often limited technical resources are not needed to design or debug the basic computer system hardware, you can concentrate on the value-added areas of applications software and any unique hardware requirements. In other words, you apply your scarce resources to the area you know best . . . your application.

Lower Your Non-Recurring Costs

The rising costs to design, develop and debug basic system hardware are reduced by using VERSAmodule products. But the cost savings don't stop here. The powerful applications development tools supporting the VERSAmodule family greatly facilitate the development and debugging of your applications software and any specialized hardware. This allows you to get it right the first time . . . avoiding costly redesigns and project delays.

VERSAmodule CIRCUIT BOARDS

Monoboard Microcomputers

VM01A Monoboard Microcomputer — MC68000 MPU, 32/64K Byte RAM, Sockets for 64K Byte ROM, four parallel I/O ports, two serial I/O ports.

VM02 Monoboard Microcomputer — MC68000 MPU, 128K Byte dual-port RAM, Sockets for 64K Byte ROM, two Multiprotocol serial I/O ports. I/O Channel Interface.

VM03 Monoboard Microcomputer — MC68010 MPU at 10 MHz, MC68451 MMU, 256K DRAM, Sockets for 64K Byte ROM, two Multiprotocol serial I/O ports and I/O Channel Interface.

Memory Modules

VM10 Random Access Memory — 128K Byte Dynamic RAM, Byte Parity, 16-bit data/word length.

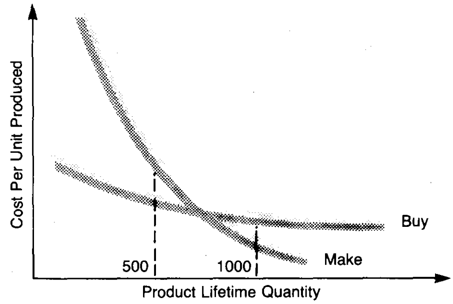
VM11 Random Access Memory — 256/512K Dynamic RAM, Error detection and correction, 16/32-bit data word length.

VM80 Combination ROM/RAM/I-O — 0/128K Byte Dynamic RAM, Sockets for 256K Byte ROM, six parallel I/O ports, two Multiprotocol serial I/O ports.

Controllers

VM20 Floppy Disk Controller — Up to 4 floppy disks, 2M Byte formatted floppy capacity, On-board IPC with data buffer.

VM60 Universal Intelligent Peripheral Controller — 4K Bytes on-board RAM, Up to 32K Bytes ROM, DMA data transfers, Wire wrap area for custom user interface.



VM21 Universal Disk Controller — UP to 4 floppy disks, Up to 2 SMD compatible hard disks, Up to 516 M Byte formatted disk capacity, On-board IPC with data buffer.

VM30 Multi-Channel Communications Module — four RS-232C serial I/O ports, One parallel printer port, ON-board IPC with data buffer.

Support

RSC1 Remote Serial Conversion Module — RS-232C to RS-449 or multidrop port, Synchronous or asynchronous operation, Half or full duplex, Eurocard form factor.

System Packaging and Accessories — 5¼ inch Chassis, Stand-Alone Card Cage, Power Supplies, Cabling Options, I/Omodule Card Cage, Mass Storage Enclosure, Industrial Card Cage System Package, VERSAbus Adapter Module.

Addition I/O

All of the I/Omodules described under I/Omodules in this catalog are compatible with the I/O Channel on VERSAmodule 02, thus extending many additional I/O and control functions to the VERSAmodule product family.

FUTURE VERSAmodule PRODUCTS

Motorola currently offers more than 20 individual hardware and software products in the VERSAmodule and I/Omodule product lines. But beyond these, Motorola engineers are at work planning and designing future products to ensure continual expansion of the VERSAmodule product line. New hardware and software products will incorporate the latest technology in easy-to-use building-block form. Future family members will include higher-performance single board computers, higher-density memory modules, and new intelligent device controllers . . . all of which take advantage of advancements in LSI technology. I/Omodule products will expand the offering of popular industry interfaces and new software will bring advanced tools like applications-oriented languages and multiprocessor capability for the VERSAdos Operating System. Other announcements from Motorola, plus those from independent vendors offering VERSAbus-compatible products, will assure an even broader selection of useful products in the future.

I/Omodules

The I/O Channel is a new system architectural concept supported in Motorola Microsystem products which allows modular I/O expansion on the local processor bus.

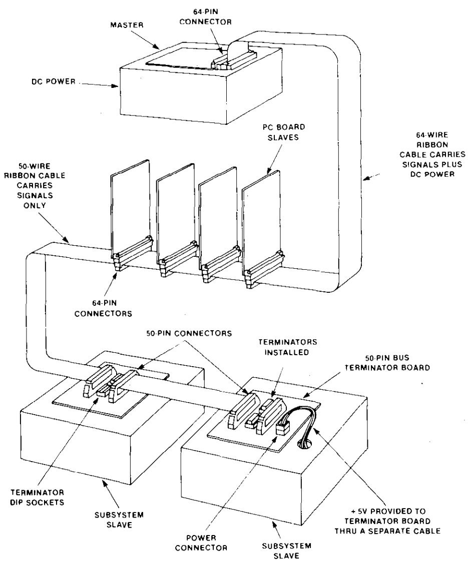
This frees the system bus to handle simultaneous high-speed data exchange and multiprocessor access requirements while permitting most lower speed system I/O activity to take place through the local I/O Channel. Thus, the advanced I/O Channel architecture affords great flexibility in I/O intensive applications such as high speed data acquisition and distributed control.

More than a dozen defined I/Omodule products already support the Motorola modular product line. The family will grow into the future with additional offerings from Motorola, and with a variety of I/O Channel compatible products from other vendors. Should you desire to design custom I/O Channel modules for your specific needs, that task is made easier by a comprehensive I/O Channel Specification Manual available from Motorola. (M68RIO1/D1)

The I/O Channel provides the following features:

- 12-bit address bus
- 8-bit bidirectional data bus
- Asynchronous operation
- Up to 2 megabyte transfer rate
- Four interrupt lines
- Reset line
- 4 MHz free running clock line

The figure below illustrates how a system might be configured using a ribbon cable bus I/O Channel. The bus master is typically a computer, but may also include a DMA controller for transferring blocks of data to or from a slave device at high speed.



I/Omodule Product Line

A. I/Omodules — I/O Channel Compatible, Single Eurocard Format.

MVME400 — Dual Channel RS-232C Serial Port providing two independent, full-duplex serial input/output ports.

MVME410 — Dual Channel 16-bit Parallel Port, four independent 8-bit ports jumper or software configurable as inputs or outputs.

MVME420 — SASI™ Peripheral Adapter provides interface to SA1400 Shugart Associates SASI Bus.

MVME435 — Buffered 9-Track Magnetic Tape Adapter to interface industry standard 800/1600 BPI, ½″ Magnetic Tape Formatter.

MVME600 — Analog Input Module with 16 channel single-ended or 8 channel differential operation.

MVME605 — Analog Output Module with 4 independent channels and 12-bit resolution.

MVME610/615/616 — Opto Isolated 120V/240V AC Input/Output modules with eight independent I/O Channels.

MVME620/625 — Opto Isolated 3VDC Input/Output modules with eight channels and isolation to 2500 Volts.

MVME932 — I/O Channel Extender Board.

MVME933 — I/O Channel Wirewrap Board.

MVME935 — I/O Channel Extender Board which converts DIN connector to 50-pin dual row header.

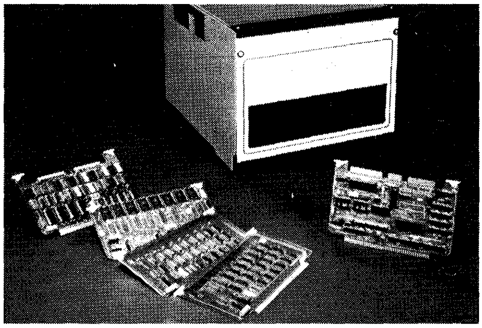
B. I/Omodules — I/O Channel Compatible, Non-Eurocard Format.

M68RWIN1-1, M68RWIN1-2 — Winchester Disk Controller for 5½″ or 8″ Winchester and Floppy Disk drive combinations.

M68RI01 — Remote Input/Output Module provides parallel I/O oprations and will accept up to 16 compatible solid state relay input and output modules.

M68RAD1 — Remote Intelligent Analog-to-Digital Conversion Module controlled by an on-board Intelligent Peripheral Controller.

Micromodules



The Motorola line of Micromodules offers a selection of modular subsystems that permits a high degree of end-product customization. It is supplemented by a sophisticated library of development software with high-level language interface to simplify man-machine interaction. An array of packaging accessories provides the proper physical environment for the system assembly.

Utilizing Motorola's extensive family of 8-bit MPU-compatible chips, Micromodules are tailored to meet the performance objectives of most industrial automation and data acquisition applications. They are priced to compete favorably with in-house development and manufacturing costs and, in many instances, they represent the most cost-effective means for rapid, reliable system implementation (or even for prototyping chip-implemented systems.)

The Modular Building Blocks

The Micromodule Family is based on a selection of differently configured *single-board microcomputers.* These vary in capabilities and applications as a result of differences in on-board microprocessors, as well as memory and I/O content. For some requirements, a single monoboard microcomputer module, supplemented by a suitable enclosure, a power supply and your applications program, will adequately serve your total needs. For other more demanding applications, the Family offers a wide range of expansion modules which tailor the system to your ultimate requirements.

Software Support

To ease programming load and allow programmers to concentrate on the end product application, incorporate the **M6809 Real-time Multitasking System (RMS09)** as the executive kernal around which a real-time applications system can be built. RMS09 is a flexible collection of systems routines from which the user can customize or 'sysgen' supervisor routines and interrupt handling routines tailored as simple or as complex as the application system requires.

Also available for MC6809-based systems is SUPERbug, a high performance monitor which also provides the facility for linking relocatable modular software routines that can be independently written and executed from EPROM, ROM or RAM. For MC6800-based Micromodules, there is MICRObug Monitor, with system software and hardware debugging capability. Also available are Editor/Assembler and Basic Interpreter packages.

MONOBOARD MICROCOMPUTER SELECTION GUIDE

Part No.	Parallel I/O	Serial I/O			Memory		Options
		RS-232C	RS-422	20mA	ROM Capacity	RAM (Bytes)	
MC6800/MC6802 Based, 1 MHz Clock Rate							
M68MM01B	1 PIA 1 PTM				To 4K**		Low Cost, Self-contained Not Expandable
M68MM01	3 PIAs				To 4K**	1K	
M68MM01A2	2 PIAs	1 ACIA		*	To 8K**	1K	
M68MM01B1A	1 PIA 1 PTM	1 ACIA		*	To 4K**	384	Cassette I/O
M68MM01D	Printer Port 1 PTM	1 ACIA	(Opt) +	*	To 10K**		Use 2K RAMs in ROM Sockets
MC6809-Based; Clock Rate 1 MHz, except M68MM19A1-2MHz							
M68MM17	1 PIA 1 PTM	2 ACIA		*	To 64K**	To 64K**	Use RAMs in ROM Sockets
M68MM19-1 19A1	1 PIA 1 PTM	1 ACIA or SSDA	(Opt) +	*	To 32K**	2K	Replace ACIA With SSDA +

NOTES:
+ = Option requires minor board modifications
* = Option requires addition of Micromodule MM11 (RS-232C to 20-mA Current-Loop Adapter)
** = User supplied

D-A Converters—All inputs are TTL compatible.

Part No.	No. of Channels	Input	Analog Output Range		Comments
			Voltage	Current	
M68MM05C	4	12-Bit Binary	0-5, 0-10 ± 2.5, ± 5 ± 10		Output Voltage Range option is strap selectable.
M68MM15CV	1 to 4*	12-Bit Binary or two's complement	0-5, 0-10, ± 5, ± 10		Input Code and Output Voltage Range Options are strap selectable.
M68MM15CI	1 to 4*	12-Bit Binary or two's complement	0-5, 0-10, ± 5, ± 10	4 to 20mA	Voltage or Current output device with strap selectable current or voltage range options.

*Add suffix 1 through 4 to part number to denote number of channels required.

A-D Converters

Part No.	No. of Channels		Resolution No. of Bits	Input Voltage (full scale, dc)	Comments
	Diff.	S.E.			
High Level M68MM15A	8	16	12	0-5 Vdc, 0-10 Vdc, ± 5 Vdc, ± 10 Vdc.	V_{in} is strap selectable
M68MM15A1	16	32	12	same as above	
M68MM05A	8		12	± 10mV to ± 10V	
M68MM05B		16	12	same as above	
Low Level M68MM15B	1		15 plus sign	± 25 mV, ± 55 mV, ± 80 mV	Expandable to 16 channels with Expander Circuits
M68MM15BEX	1 to 4		Channel Expander for above		

Micromodules

EXORbus-Compatible Memory Modules

	Capacity (K Bytes)	Features
RAM-Static, NMOS		
M68MM06	2	Clock Speed—1 MHz
MEX6808-22	8	
MEX6816-22S	16	With Parity. Clock speed – 1 or 2 MHz (with stretched Phase 2)
RAM-Static, CMOS		
M68MM09	4	Clock speed = 1.5 or 2 MHz (with stretched phase 2) On-board ckt. for user-installed parity.
M68MM21	8	Optional parity.
M68MM21-1	16	Optional parity.
RAM-Dynamic, with parity		
MEX6816-22D	16	Jumper selectable 1-, 1.5-, or 2- MHz speed; Row-addressable in 16K byte blocks.
MEX6832-22	32	
MEX6848-22	48	
MEX6864-22	64	
RAM-Dynamic; with hidden refresh; clock speed = 1 MHz; all with parity.		
MEX6816-1HR	16	Organized into independently addressable rows of 16K bytes each.
MEX6832-1HR	32	
MEX6848-1HR	48	
MEX6864-1HR	64	

Unpopulated Modules—User supplies chip set

	Number Of Sockets	EPROM/ROM CAPACITY (Memory Chip Number)	OPTION RAM CAPACITY (Memory Chip Number)
MEX68RR	20	16K (MCM68708/6830)	512 (MCM6810)
M68MM04	16	16K (MCM68708 6830)	0
M68MM04A	16	64K (1K, 2K or 4K capacity)	8K (1K or 2K capacity)

Serial-Format Digital I/O

ACIA Modules — MEX6850, 50-2

Offers both TTY and RS-232C data terminal interface, with eight switch-selectable baud rates between 110 and 9600 baud. MEX6850 operates at 1MHz and is configured with Modem output; 6850-2, at 2MHz, is configured with 20 mA TTY output.

Quad Serial I/O — M68MM07

Supplied with four MC6850 ACIAs, or with user supplied MC6852 SSDAs for either asynchronous or synchronous operation. Strap options permit RS-232C, RS-422, RS-423 or 20mA interface and baud-rate selection for each of the four ports.

RS-232C to TTY Adapter — M68MM11

Converts RS-232C output to 20 mA TTY operation.

8 Channel Serial I/O Module — M68MM18

Provides eight asynchronous RS-232C channels. Each channel is strap selectable to baud rates from 75 to 115K BPS. Memory location is strap selectable in a block of eight channels.

GPIB Modules

Provide interface between various MPUs and the IEEE STD 488-1978 interface bus. MM12A provides Listener/Talker functions for sending and receiving data bytes, requesting service and responding to parallel and serial polls. MM12 and 12-1 add the controller function that permits the system to send commands and conduct serial and parallel polls.

Listener/Talker for MC6800-type systems	**M68MM12A**
Listener/Talker/Controller for MC6800-type systems	**M68MM12**
Listener/Talker/Controller for MC6809-type systems	**M68MM12-1**

Memory-I/O-Timer Expansion Module — M68MM16-1, 2, 3

Provides functional expansion of Monoboard MM01 (version 16-1), or MM19 (version 16-2), and can be used as MM19 Emulator in an EXORset Development System (version 16-3). Includes asynchronous serial data port with strap-selectable RS-232C, RS-422, or RS-423 interface, parallel interface port with 16 data lines and 2K of static RAM, four control lines, three 16-bit programmable counter/timers, and four sockets for user installed, single 5-volt-supply MOS or bipolar memories.

Parallel-Format Digital I/O

Universal PIA-Controlled I/O — MEX6820, 21-2

Contains two MC6820 Peripheral Interface Adapters (PIA's) for a total of four separate 8-bit I/O ports for peripheral interfacing.

32-In/32-Out Expansion Module — M68MM03

Contains 32 bits of parallel input and 32 bits of parallel output in four continguous 8-bit bytes. Used for simultaneous transfer of 4 bytes of informtion between an MPU and an external system to speed up the data transfer cycle.

16/32-Channel Relay Output — M68MM13A, B

Contains 16 (MM13A) or 32 (MM13B) on-board reed relay output channels to isolate the microcomputer from the system(s) being controlled.

24-Channel Optically Isolated Input Modules — M68MM13C, D

Provides three byte-oriented (8-bit) input channels that have high electrical isolation between microcomputer and equipment being monitored. Input voltages in excess of 17 volts are read as logical "1"; 4 volts or less represent logical "0." MM13D provides an on-board wetting source for applications requiring switch and relay inputs.

Quad Parallel Interface Adapter — M68MM22

Utilizes four PIAs in a versatile buffered I/O configuration that allows up to 64 high-voltage (200 Vdc or 280 Vac) or high-current (to 3A) signals to be monitored or controlled.

Packaging Hardware

Part No.	Description
M68MMCC05	5-Card Cardcage
M68MMCC10	10-Card Cardcage
M68MMFLC1	Front Load Chassis, 14 Card, 110 Vac
M68MMLC1	Long Chassis, 10 Card, 110 Vac
M68MMSC1	Short Chassis, 5-Card, 110 Vac
M68MMPS-1	Power Supply, 110 Vac

A16664 Printed in USA 2-84 TMP 60,000 DL-133