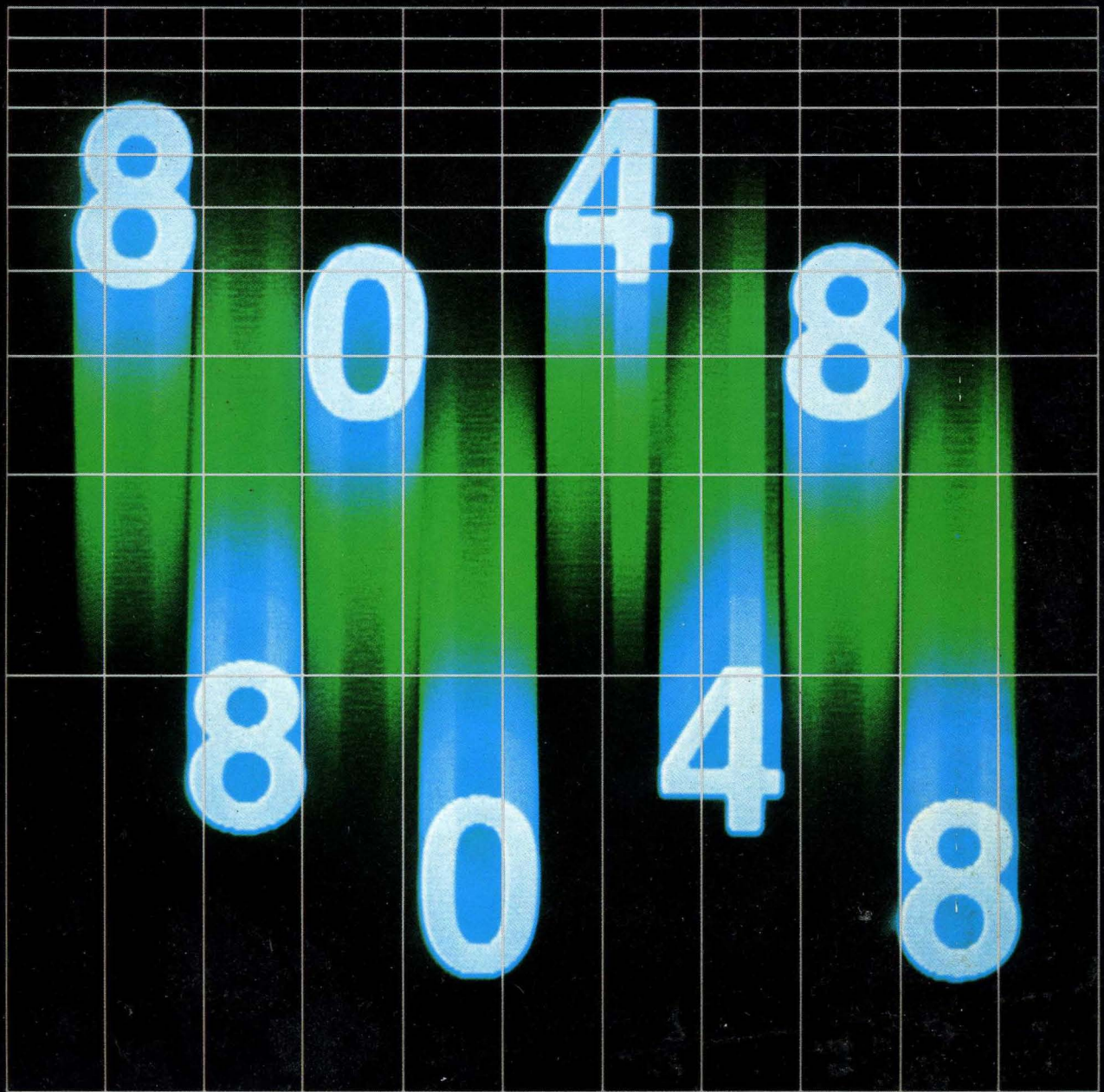
intel

MCS-48® Family of Single Chip Microcomputers User's Manual



intel

MCS-48™ FAMILY OF SINGLE CHIP MICROCOMPUTERS USER'S MANUAL

September 1980

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Table of Contents

CHAPTER 6

MCS-48™ Component Specifications

CHAPTER 7

MCS-51™ Component Specification

CHAPTER 8

Compatible MCS-48™ Components

CHAPTER 9

Peripherals

CHAPTER 10

Support Products

APPENDICES

Introduction

1

INTRODUCTION

1.0 Introduction to MCS-48™

Recent advances in NMOS technology have allowed Intel for the first time to place enough capability on a single silicon die to create a true single-chip microcomputer containing all the functions required in a digital processing system. A set of such microcomputers on single chips, their variations, and optional peripherals are collectively called the MCS-48 microcomputer family. These products are fully described in this manual.

The head of the family is the 8048 microcomputer which contains the following functions in a single 40 pin package:

8-Bit CPU
1K x 8 ROM Program Memory
64 x 8 RAM Data Memory
27 I/O Lines
8-Bit Timer/Event Counter

A 2.5 or 5.0 microsecond cycle time and a repertoire of over 90 instructions, each consisting of either one or two cycles, makes the single chip 8048 the equal in performance of most presently available multi-chip NMOS microprocessors. The 8048 is, however, a true "low-cost" microcomputer. A single 5V supply requirement for all MCS-48 components assures that "low cost" also applies to the power supply in your system.

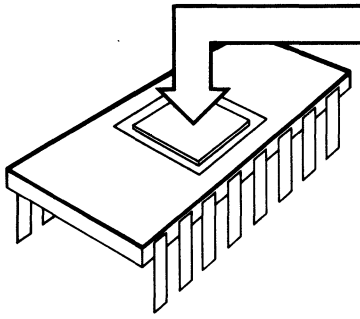
New Family Members

The MCS-48 family of microcomputers which began with the 8048 and 8748 has now been expanded with new members which provide either more capability or lower cost than the original family members. While broadening the applications possible with a single chip microcomputer, these new microcomputers share both the instruction set and development support of the 8048.

The 8049 is a single-chip microcomputer which is completely interchangeable with the 8048, but contains twice the program memory and twice the data memory of the 8048.

The 8022 is an 8021-based microcomputer with additional memory, I/O, and an A/D converter.

The 8021 is a new very low cost MCS-48 family member which contains a subset of



8049	8048	8021	8022	FEATURES
✓	✓	✓	✓	8-BIT CPU
2K x 8	1K x 8	1K x 8	2K x 8	PROGRAM MEMORY
128 x 8	64 x 8	64 x 8	64 x 8	DATA RAM
27	27	21	28	I/O LINES
✓	✓	✓	✓	TIMER COUNTER
✓	✓	✓	✓	OSCILLATOR AND CLOCK
✓	✓	✓	✓	RESET CIRCUIT
✓	✓		✓	INTERRUPT

ON CHIP FEATURES

the 8048's instruction set and incorporates several new features critical in low cost applications.

Even with low component costs, however, a project may be jeopardized by high development and rework costs resulting from an inflexible production design. Intel has solved this problem by creating two pin-compatible versions of the 8048 microcomputer: the 8048 with mask Programmable ROM program memory for low cost production and the 8748 with user programmable and erasable EPROM program memory for prototype development. The 8748 is essentially a single chip microcomputer "breadboard" which can be modified over and over again during development and pre-production then replaced by the low cost 8021*, 8048, or 8049 ROM for volume production. The 8748 provides a very easy transition from development to production and also provides an easy vehicle for temporary field updates while new ROMs are being made.

SPECIAL FEATURES

- **SINGLE 5V SUPPLY**
- **40 PIN DIP OR 28 PIN DIP**
- **PIN COMPATIBLE ROM AND EPROM**
- **2.5, 5.0 AND 10.0 μsec CYCLE VERSIONS**
- **ALL INSTRUCTIONS 1 OR 2 CYCLES**
- **SINGLE STEP**
- **8 LEVEL STACK**
- **2 WORKING REGISTER BANKS**
- **LC, XTAL, OR EXTERNAL FREQUENCY SOURCE**
- **OPTIONAL CLOCK OUTPUT**
- **POWER DOWN STANDBY MODE**

To allow the MCS-48 to solve a wide range of problems and to provide for future expansion, all 8048 and 8049 functions have been made externally expandable using either special expanders or standard memories and peripherals. An efficient low cost means of I/O expansion is provided by either the 8243 I/O Expander or standard TTL or CMOS circuits. The 8243 provides 16 I/O lines in a 24 pin package. For systems with large I/O requirements, multiple 8243s can be used.

For such applications as Keyboards, Displays, Serial communication lines, etc. standard MCS-80/85 peripheral circuits may be added. Program and data memory may be expanded using standard memories or the 8355 and 8155 memories that also include programmable I/O lines and timing functions.

For applications which require a more custom tailored interface, the 8041 or 8741 Universal Peripheral Interface (UPI-41) devices can be used. The UPI-41 devices are available in both ROM and EPROM versions and are essentially slave versions of the 8048/8748 which are designed to interface directly with expandable MCS-48 processors and provide flexible intelligent I/O capability. The 8041/8741 share the instruction set of the MCS-48 family of processors.

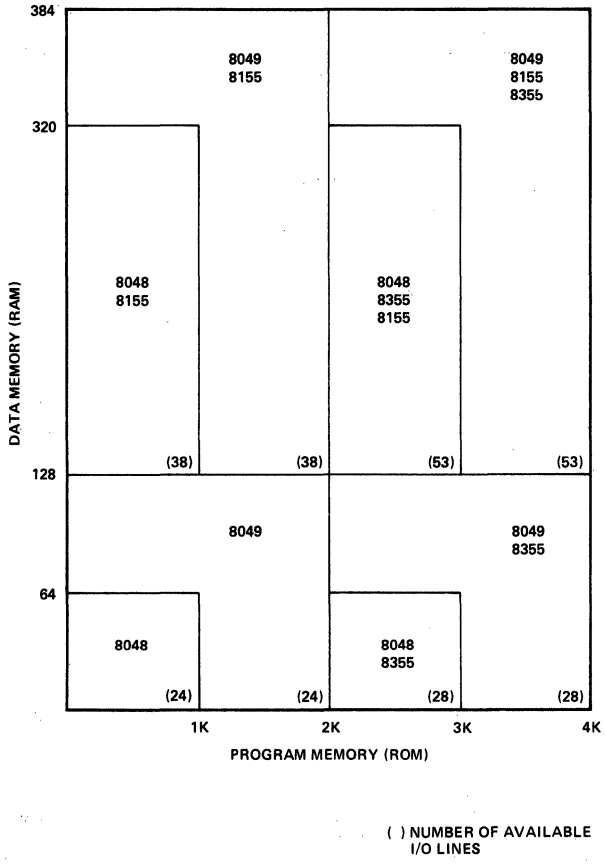
The 8035 and 8039 are an 8048 or 8049 respectively without internal program memory that allows the user to match his program memory requirements exactly by using a wide variety of external memories. The 8035 and 8039 allow the user to select a minimum cost system no matter what his program memory requirements. The 8035L is an 8035 with the powerdown mode of the 8048.

The MCS-48 processors are designed to be efficient control processors as well as arithmetic processors. They provide an instruction set which allows the user to directly set and reset individual lines within its I/O ports as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions make these processors very efficient in implementing standard logic functions. Also, special attention has been given to code efficiency. Over 70% of the instructions are a single byte long and all others are only two bytes long. This means many functions requiring 1.5K to 2.0K bytes in other computers may very well be compressed into the 1K words resident in the 8048 or up to 3K to 4K equivalent bytes may be compressed into the 8049.

*The 8021 is code compatible but not pin compatible with the 8748.

	FUNCTION	PART NUMBER	DESCRIPTION	COMMENTS
MCS-48™	Microcomputers	8021 8022 8048 8035 8035L 8049 8039 8748-8 8035-8 8748-6	1K ROM Program Memory 2K ROM Program Memory with A/D } 10 μsec Cycle 1K ROM Program Memory No Program Memory 64 x 8 RAM } 2.5 μsec Cycle 8035 with Power Down Mode 2K ROM Program Memory No Program Memory 128 x 8 RAM } 1.36 μsec Cycle 1K EPROM Program Memory No Program Memory } 5.0 μsec Cycle 1K EPROM 0°C–55°C, 6 MHz	Compatible versions of the single chip microcomputers provide mask programmed, light erasable, or no internal program memory.
	Memory and I/O Expanders	8355 8755A 8155/56 8185	2K x 8 ROM with 16 I/O Lines 2K x 8 EPROM with 16 I/O Lines 256 x 8 RAM with 22 I/O Lines and Timer 1K x 8 RAM	Compatible devices allow direct expansion of MCS-48 functions with no additional external components.
	I/O Expander	8243	16 Line I/O Expander	Low cost I/O expander.
COMPATIBLE MCS-80/85™ COMPONENTS	Standard ROMs	2308 2316E 2332	1K x 8 450 ns 2K x 8 450 ns 4K x 8 450 ns	Allows low cost external expansion of Program Memory. Each ROM is interchangeable with an EPROM.
	Standard EPROMs	2708 2716 2732	1K x 8 450 ns Light Erasable 2K x 8 450 ns Light Erasable 4K x 8 450 ns Light Erasable	User programmable and erasable.
	Standard RAMs	2111A-4 2101A-4	256 x 4 450 ns Common I/O 256 x 4 450 ns Separate I/O	Data memory can be easily expanded using standard NMOS RAMs.
	Standard I/O	8212 8255A 8251A 8273	8-Bit I/O Port Programmable Peripheral Interface Programmable Communication Interface Programmable HDLC/SDLC Controller	Serves as Address Latch or I/O port. Three 8-bit programmable I/O ports. Serial Communications Receiver/Transmitters.
	Standard Peripherals	8205 8214 8216 8226 8253 8279/78 8291 8294 8295	1-of-8 Binary Decoder Priority Interrupt Controller Bidirectional Bus Driver Bidirectional Bus Driver (Inverting) Programmable Interval Timer Programmable Keyboard/Display Interface (64 Keys/128 Keys) GPIB Talker/Listener Data Encryption Unit Dot Matrix Printer Controller	MCS-80 peripheral devices are compatible with the MCS-48, allowing easy addition of such specialized interfaces as the 8279 Keyboard/Display Interface. Future MCS-80/85 devices will also be compatible.
	Universal Peripheral Interface	8041A 8741A	ROM Program Memory EPROM Program Memory	User programmable to perform any custom I/O and control functions.

MCS-48™ MICROCOMPUTER COMPONENTS



THE EXPANDED MCS-48™ SYSTEM

The chart above shows the expansion possibilities using the 8048 and 8049 in various combinations with the Intel® 8355/8755 Program Memory and I/O Expander and the 8155 Data Memory and I/O Expander. Data Memory can be expanded beyond the resident words in blocks of 256 by adding 8155's. Program Memory can be expanded beyond the resident 1K or 2K in blocks of 2K by using the 8355/8755 in combination with the 8048 or 8049. If all external memory is desired, the 8035 or 8039 can be substituted for the 8048 and 8049.

1.1 The Function of a Computer

This chapter introduces certain basic computer concepts. It provides background information and definitions which will be useful in later chapters of this manual. Those already familiar with computers may skip this material, at their option.

1.1.1 A Typical Computer System

A typical digital computer consists of:

A central processor unit (CPU)
Program Memory
Data Memory
Input/output (I/O) ports

The processor memory serves as a place to store Instructions, the coded pieces of information that direct the activities of the CPU, while Memory stores the Data, the coded pieces of information that are processed by the CPU. A group of logically related instructions stored in memory is referred to as a Program. The CPU "reads" each instruction from memory in a logically determined sequence, and uses it to initiate processing actions. If the program sequence is coherent and logical, processing the program will produce intelligible and useful results. The program must be organized such that the CPU does not read a non-instruction word when it expects to see an instruction.

The CPU can rapidly access any data stored in memory; but often the memory is not large enough to store the entire data bank required for a particular application. The problem can be resolved by providing the computer with one or more Input Ports. The CPU can address these ports and input the data contained there. The addition of input ports enables the computer to receive information from external equipment (such as a paper tape reader or floppy disk) at high rates of speed and in large volumes.

A computer also requires one or more Output Ports that permit the CPU to communicate the result of its processing to the outside world. The output may go to a display, for use by a human operator, to a peripheral device that produces "hard-copy", such as a line-printer, to a peripheral storage device, such as a floppy disk unit, or the output may constitute process control signals that direct the operations of another system, such as an automated assembly line. Like input ports, output ports are addressable. The input and output ports together permit the processor to communicate with the outside world.

The CPU unifies the system. It controls the functions performed by the other components. The CPU must be able to fetch instructions from memory, decode their binary contents and execute them. It must also be able to reference memory and I/O ports as necessary in the execution of instructions. In addition, the CPU should be able to recognize and respond to certain external control signals, such as INTERRUPT requests. The functional units within a CPU that enable it to perform these functions are described below.

1.1.2 The Architecture of a CPU

A typical central processor unit (CPU) consists of the following interconnected functional units:

Registers
Arithmetic/Logic Unit (ALU)
Control Circuitry

Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. Other registers, such as the accumulator, are for more general purpose use.

Accumulator

The accumulator usually stores one of the operands to be manipulated by the ALU. A typical instruction might direct the ALU to add the contents of some other register to the contents of the accumulator and store the result in the accumulator itself. In general, the accumulator is both a source (operand) and a destination (result) register. Often a CPU will include a number of additional general purpose registers that can be used to store operands or intermediate data. The availability of general purpose registers

eliminates the need to "shuffle" intermediate results back and forth between memory and the accumulator, thus improving processing speed and efficiency.

Program Counter (Jumps, Subroutines and the Stack):

The instructions that make up a program are stored in the system's memory. The central processor references the contents of memory in order to determine what action is appropriate. This means that the processor must know which location contains the next instruction.

Each of the locations in memory is numbered, to distinguish it from all other locations in memory. The number which identifies a memory location is called its Address. The processor maintains a counter which contains the address of the next program instruction. This register is called the Program Counter. The processor updates the program counter by adding "1" to the counter each time it fetches an instruction, so that the program counter is always current (pointing to the next instruction).

The programmer therefore stores his instructions in numerically adjacent addresses, so that the lower addresses contain the first instructions to be executed and the higher addresses contain later instructions. The only time the programmer may violate this sequential rule is when an instruction in one section of memory is a Jump instruction to another section of memory.

A jump instruction contains the address of the instruction which is to follow it. The next instruction may be stored in any memory location, as long as the programmed jump specifies the correct address. During the execution of a jump instruction, the processor replaces the contents of its program counter with the address embodied in the Jump. Thus, the logical continuity of the program is maintained.

A special kind of program jump occurs when the stored program "Calls" a subroutine. In this kind of jump, the processor is required to "remember" the contents of the program counter at the time that the jump occurs. This enables the processor to resume execution of the main program when it is finished with the last instruction of the subroutine.

A Subroutine is a program within a program. Usually it is a general-purpose set of instructions that must be executed repeatedly in the course of a main program. Routines which calculate the square, the sine, or the logarithm of a program variable are good examples of functions often written as subroutines. Other examples might be programs designed for inputting data to a particular peripheral device.

The processor has a special way of handling subroutines, in order to insure an orderly return to the main program. When the processor receives a Call instruction, it increments the Program Counter and stores the counter's contents in a reserved memory area known as the Stack. The Stack thus saves the address of the instruction to be executed after the subroutine is completed. Then the processor loads the address specified in the Call into its Program Counter. The next instruction fetched will therefore be the first step of the subroutine.

The last instruction in any subroutine is a Return. Such an instruction need specify no address. When the processor fetches a Return instruction, it simply replaces the current contents of the Program Counter with the address on the top of the stack. This causes the processor to resume execution of the calling program at the point immediately following the original Call instruction.

Subroutines are often Nested; that is, one subroutine will sometimes call a second subroutine. The second may call a third, and so on. This is perfectly acceptable, as long as the processor has enough capacity to store the necessary return addresses, and the logical provision for doing so. In other words, the maximum depth of nesting is determined by the depth of the stack itself. If the stack has space for storing three return addresses, then

three levels of subroutines may be accommodated.

Instruction Register and Decoder

Every computer has a Word Length that is characteristic of that machine. A computer's word length is usually determined by the size of its internal storage elements and interconnecting paths (referred to as Buses); for example, a computer whose registers and buses can store and transfer 8-bits of information has a characteristic word length of 8-bits and is referred to as an 8-bit parallel processor. An 8-bit parallel processor generally finds it most efficient to deal with 8-bit binary fields, and the memory associated with such a processor is therefore organized to store 8-bits in each addressable memory location. Data and instructions are stored in memory as 8-bit binary numbers, or as numbers that are integral multiples of 8-bits: 16-bits, 24-bits, and so on. This characteristic 8-bit field is often referred to as a Byte. If however, efficient handling of 4 or even 1-bit data is necessary special processor instructions can provide this capability.

Each operation that the processor can perform is identified by a unique byte of data known as an Instruction Code or Operation Code. An 8-bit word used as an instruction code can distinguish between 256 alternative actions, more than adequate for most processors.

The processor fetches an instruction in two distinct operations. First, the processor transmits the address in its Program Counter to the program memory. Then the program memory returns the addressed byte to the processor. The CPU stores this instruction byte in a register known as the Instruction Register, and uses it to direct activities during the remainder of the instruction execution.

The 8-bits stored in the instruction register can be decoded and used to selectively activate one of a number of output lines. Each line represents a set of activities associated with execution of a particular instruction code. The enabled line can be combined with selected timing pulses, to develop electrical signals that can then be used to initiate specific actions. This translation of code into action is performed by the Instruction Decoder and by the associated control circuitry.

An 8-bit instruction code is often sufficient to specify a particular processing action. There are times, however, when execution of the instruction requires more information than 8-bits can convey.

One example of this is when the instruction references a memory location. The basic instruction code identifies the operation to be performed, but cannot specify the object address as well. In a case like this, a two byte instruction must be used. Successive instruction bytes are stored in sequentially adjacent memory locations, and the processor performs two fetches in succession to obtain the full instruction. The first byte retrieved from memory is placed in the processor's instruction register, and subsequent byte is placed in temporary storage; the processor then proceeds with the execution phase.

Address Register(s)

A CPU may use a register to hold the address of a memory location that is to be accessed for data. If the address register is Programmable, (i.e., if there are instructions that allow the programmer to alter the contents of the register) the program can "build" an address in the address register prior to executing a Memory Reference instruction (i.e., an instruction that reads data from memory, writes data to memory or operates on data stored in memory).

Arithmetic/Logic Unit (ALU)

All processors contain an arithmetic/logic unit, which is often referred to simply as the ALU. The ALU, as its name implies, is that portion of the CPU hardware which performs the arithmetic and logical operations on the binary data.

The ALU must contain an Adder which is capable of combining the contents of two registers in accordance with the logic of binary arithmetic. This provision permits the

processor to perform arithmetic manipulations on the data it obtains from memory and from its other inputs.

Using only the basic adder a capable programmer can write routines which will subtract, multiply and divide, giving the machine complete arithmetic capabilities. In practice, however, most ALUs provide other built-in functions, including boolean logic operations, and shift capabilities.

The ALU contains Flag Bits which specify certain conditions that arise in the course of arithmetic and logical manipulations. It is possible to program jumps which are conditionally dependent on the status of one or more flags. Thus, for example, the program may be designed to jump to a special routine if the carry bit is set following an addition instruction.

Control Circuitry

The control circuitry is the primary functional unit within a CPU. Using clock inputs, the control circuitry maintains the proper sequence of events required for any processing task. After an instruction is fetched and decoded, the control circuitry issues the appropriate signals (to units both internal and external to the CPU) for initiating the proper processing action. Often the control circuitry will be capable of responding to external signals, such as an interrupt. An Interrupt request will cause the control circuitry to temporarily interrupt main program execution, jump to a special routine to service the interrupting device, then automatically return to the main program.

1.1.3 Computer Operations

There are certain operations that are basic to almost any computer. A sound understanding of these basic operations is a necessary prerequisite to examining the specific operations of a particular computer.

Timing

The activities of the central processor are cyclical. The processor fetches an instruction, performs the operations required, fetches the next instruction, and so on. This orderly sequence of events requires precise timing, and the CPU therefore requires a free running oscillator clock which furnishes the reference for all processor actions. The combined fetch and execution of a single instruction is referred to as an Instruction Cycle. The portion of a cycle identified with a clearly defined activity is called a State. And the interval between pulses of the timing oscillator is referred to as a Clock Period. As a general rule, one or more clock periods are necessary for the completion of a state, and there are several states in a cycle.

Instruction Fetch

The first state(s) of any instruction cycle will be dedicated to fetching the next instruction. The CPU issues a read signal and the contents of the program counter are sent to program memory, which responds by returning the next instruction word. The first byte of the instruction is placed in the instruction register. If the instruction consists of more than one byte, additional states are required to fetch the second byte of the instruction. When the entire instruction is present in the CPU, the program counter is incremented (in preparation for the next instruction fetch) and the instruction is decoded. The operation specified in the instruction will be executed in the remaining states of the instruction cycle. The instruction may call for a data memory read or write, an input or output and/or an internal CPU operation, such as a register-to-register transfer or an add operation.

Memory Read

An instruction fetch is merely a special program memory read operation that brings the instruction to the CPU's instruction register. The instruction fetched may then call for data to be read from data memory into the CPU. The CPU again issues a read signal and sends the proper memory address; memory responds by returning the requested word. The data received is placed in the accumulator or one of the other general purpose registers (not the instruction register).

Memory Write

A memory write operation is similar to a read except for the direction of data flow. The CPU issues a write signal, sends the proper memory address, then sends the data word to be written into the addressed data memory location.

Input/Output

Input and Output operations are similar to memory read and write operations with the exception that an I/O port is addressed instead of a memory location. The CPU issues the appropriate input or output control signal, sends the proper address and either receives the data being input or sends the data to be output.

Data can be input/output in either parallel or serial form. All data within a digital computer is represented in binary coded form. A binary data word consists of a group of bits; each bit is either a one or a zero. Parallel I/O consists of transferring all bits in the word at the same time, one bit per line. Serial I/O consists of transferring one bit at a time on a single line. Naturally serial I/O is much slower, but it requires considerable less hardware than does parallel I/O.

Interrupts

Interrupt provisions are included on many central processors, as a means of improving the processor's efficiency. Consider the case of a computer that is processing a large volume of data, portions of which are to be output to a printer. The CPU can output a byte of data within a single machine cycle but it may take the printer the equivalent of many machine cycles to actually print the character specified by the data byte. The CPU could then remain idle waiting until the printer can accept the next data byte. If an interrupt capability is implemented on the computer, the CPU can output a data byte then return to data processing. When the printer is ready to accept the next data byte, it can request an interrupt. When the CPU acknowledges the interrupt, it suspends main program execution and automatically branches to a routine that will output the next data byte. After the byte is output, the CPU continues with main program execution. Note that this is, in principle, quite similar to a subroutine call, except that the jump is initiated externally rather than by the program.

More complex interrupt structures are possible, in which several interrupting devices share the same processor but have different priority levels. Interruptive processing is an important feature that enables maximum utilization of a processor's capacity for high system throughput.

1.2 Programming a Microcomputer

1.2.1 Machine Language Programming

A microprocessor is instructed what to do by programming it with a series of instructions stored in Program Memory. The processor fetches these instructions one at a time and performs the operation indicated. These instructions must be stored in a form that the processor can understand. This format is referred to as Machine Language. For most microprocessors this instruction is a group of 8 binary bits (1's and 0's) called a word (also called a byte if the word is 8-bits). Some instructions require more than one location in Program Memory. To execute a multi-byte instruction, the processor must execute multiple fetches of program memory before performing the instruction. Because multi-byte instructions take more Program Memory and take longer to execute than single byte instructions their use is usually kept to a minimum.

A processor may be programmed by writing a sequence of instructions in the binary code (ones and zeros) which the machine can interpret directly. This is machine language programming and it is very useful where the program to be written is small and the application requires that the designer have an intimate knowledge of the microprocessor. Machine language programming allows the user, because of his detailed knowledge, to use many programming "tricks" to produce the most compact and efficient code possible.

The following is an example of a machine language program: This program reads 5 sequential 8-bit words in from an I/O port and stores them sequentially in data memory. The program starts by initializing two registers, one which determines where the data is to be stored and another which counts the number of words to be stored. When finished the processor continues on to the next instructions.

Step Number	Machine Code	Explanation
0	1011 1000	Load decimal 32 in register R0
1	0010 0000	
2	1011 1010	Load decimal 5 in register R2
3	0000 0101	
4	0000 1001	Load Port 1 to accumulator
5	1010 0000	Transfer contents of accumulator to register addressed by register 0
6	0001 1000	Increment R0 by 1
7	1110 1010	Decrement register 2 by 1, if result is zero continue to step 9, if not go to step 4
8	0000 0100	
9	—	
10	—	

As you can see, writing machine instructions in ones and zeros can be very laborious and subject to error. It is almost always more efficient to represent each 8-bits of machine language code in a shorthand format called Hexadecimal. The term hexadecimal results from the character set used in hexadecimal notation. Hexadecimal is merely an extension of the normal decimal numbers by the addition of the first six letters of the alphabet. This gives a total of 16 different characters. Each hexadecimal "digit" can represent 16 values or the equivalent of four binary bits; therefore, each 8-bit machine language word can be represented by 2 hexadecimal (hex for short) digits. The correspondence among the decimal, binary, and hex number systems is given below:

Decimal	Hex	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Our machine language program then becomes:

Step	Hex Code
0	B8
1	20
2	BA
3	05
4	09
5	F0
6	18
7	EA
8	04

This coding is now quite efficient to write and read and coding errors are much easier to detect. Hex coding is usually very efficient for small programs (a few hundred lines of code). However, it does have two major limitations in larger programs:

1. Hex coding is not self-documenting, that is, the code itself does not give any indication in human terms of the operation to be performed. The user must learn each code or constantly use a Program Reference Card to convert.

2. Hex coding is absolute, that is, the program will work only when stored in a specific location in program memory. This is because the branch or jump instructions in the program reference specific addresses elsewhere in the program. In the example above steps 7 and 8 reference step (or address) 4. If the program were to be moved, step 8 would have to be changed to refer to the new address of step 4.

1.2.2 Assembly Language Programming

Assembly language overcomes the disadvantages of machine language by allowing the use of alphanumeric symbols to represent machine operation codes, branch addresses, and other operands. For example, the instruction to increment the contents of register 0 becomes INC R0 instead of the hex 18, giving the user at a glance the meaning of the instruction. Our example program can be written in assembly language as follows:

Step No.	Hex Code		Assembly Code
0	B8		MOV R0, #32
1	20		
2	BA		MOV R2, #05
3	05		
4	09	INP:	IN A, P1
5	AO		MOV @R0, A
6	18		INC R0
7	EA		DJNZ R2, INP
8	04		

The first statement can be verbalized as follows: Move to Register 0 the decimal number 32. Move instructions are always structured such that the destination is first and the source is second. The pound sign "#" indicates that the source is "immediate" data (data contained in the following byte of program memory). In this case data was specified as a decimal 32, however, this could have been written as a hex 20H or a binary 0010 0000B since the assembler will accept either form. Notice also that in this instance two lines of hex code are represented by one line of assembly code.

The input instruction IN A, P1 has the same form as a MOV instruction indicating that the contents of Port 1 are to be transferred to the accumulator. In front of the input instruction is an address label which is delineated by a colon. This label allows the program to be written in a form independent of its final location in program memory since the branch instruction at the end of the program can refer to this label rather than a specific address. This is a very important advantage of assembly language programs since it

allows instructions to be added or deleted throughout the program during debugging without requiring that any jump addresses be changed.

The next instruction MOV @R0, A can be verbalized as, Move to the data memory location addressed by R0, the contents of the accumulator. The @ sign indicates an indirect operation whereby the contents of either register 0 or register 1 acts as a pointer to the data memory location to be operated on.

The last instruction is a Decrement and Jump if Not Zero instruction which acts in combination with the specified register as a loop counter. In this case register 2 is loaded with 5 initially and then decremented by one each time the loop is executed. If the result of the decrement is not zero, the program jumps to INP and executes another input operation. The fifth time thru the loop the result is zero and execution falls through to whatever routine follows the DJNZ instruction.

In addition to the normal features provided by assemblers, more advanced assemblers such as that for the MCS-48 offer such things as evaluation of expressions at assembly time, conditional assembly, and macro capability.

1. Evaluation of Expressions - Certain assemblers allow the use of arithmetic expressions and multiple symbols in the operand portion of instructions. For instance the MCS-48 assembler accepts instructions such as:

```
ADD A, # ALFA*BETA/2
```

ALFA and BETA are two previously defined symbols. At assembly time the expression ALFA*BETA/2 will be evaluated and the resulting number (which is the average of ALFA and BETA) will be treated as immediate data and designated as the second byte of the ADD immediate instruction. This expression has allowed the immediate data of this instruction to be defined in a single statement and eliminated the need for a third symbol equal to ALFA*BETA/2.

2. Conditional Assembly - Conditional assembly allows the programmer to select only certain portions of his assembly language (source) program for conversion to machine (object) code at assembly time. This allows for instance, the inclusion of various "debug" routines to be included in the program during development. Using conditional assembly, they can then be left out when the final assembly is done.

Conditional assembly also allows several versions of one basic program to be generated by selecting various portions of a larger program at assembly time.

3. Macro's - A macro instruction is essentially a symbol which is recognized by the assembler to represent a specific sequence of several standard instructions. A macro is a shorthand way of generating the same sequence of instructions at several locations in a program without having to rewrite the sequence each time it is used. For example, a typical macro instruction might be one which performs a subtract operation. The 8048 does not have a subtract instruction as such but the operation can be performed easily with three instructions:

```
CPL A
ADD A, REG
CPL A
```

This routine subtracts a register from the accumulator and leaves the result in the accumulator. This sequence can be defined as a macro with the name SUB and an operand which can be R0 to R7. To subtract R7 from the accumulator then, the programmer merely has to write:

```
SUB R7
```

and the assembler will automatically insert the three instructions above with R7 substituted for REG.

Once the assembly language source code is written it can be converted to machine executable object code by passing it through an assembler program. The MCS-48 assembler is a program which runs on the 8080-based Intellec MDS system explained in the next section.

1.3 Developing An MCS-48™ Based Product

Although the development of a microcomputer based product may differ in detail from the development cycle of a product based on TTL logic or relays, the basic procedures are the same — only the tools are different.

1.3.1 Education

The first step of course is to become familiar with what the microcomputer is and what it can do. The first step in this education is this document, the MCS-48™ User's Manual. The user's manual gives a detailed description of the MCS-48 family of components and how they may be used in various system configurations. Also included is a description of the 8048 instruction set and examples of how the instructions may be used. For a more complete discussion of the instruction set and programming techniques the MCS-48 Assembly Language Manual is also available.

If time is critical in getting started in microcomputers, individuals can attend one of many Intel sponsored 5-day training courses which give basic instruction in the MCS-48 as well as hands-on experience with MCS-48 development systems. These courses are a convenient means of getting started with the MCS-48, particularly for those not familiar with microprocessors.

After general familiarization is complete, either through self-instruction or a training course, the next step is to gain a better "feel" for what a microprocessor can do in your own applications by writing several exercise programs which perform basic functions. You may require such things as I/O routines, delays, counting functions, look-up tables, arithmetic functions, and logical operations which can serve as a set of building blocks for future applications programs. Several basic programming examples are included in the MCS-48 Assembly Language Manual while the Intel User's Library is a source of more specific applications routines.

1.3.2 Function Definition

After a thorough understanding of the microprocessor is achieved, the functions to be implemented can be defined using a flowchart method to describe each basic system function and the sequence in which the processor executes these functions. Once the system is flowcharted, critical time-related functions can be identified and sample programs written to verify that performance requirements can be met.

1.3.3 Hardware Configuration

The next step involves the definition of the microcomputer hardware required to implement the function. Input/Output capability must be defined in terms of number of inputs, number of outputs, bi-directional lines, latching or non-latching I/O, output drive capability, and input impedance. The number of words of RAM storage required for intermediate results and data storage must then be determined. The type of system will dictate whether battery backup is needed to maintain data RAM during power failure.

Probably the most difficult parameter to define initially is the amount of program memory needed to store the applications program. Although previously written exercise programs will make this estimate more accurate, a generous amount of "breathing room" should be allowed in program memory until coding is complete and the exact requirements are known. Many special functions such as serial communications (TTY) or keyboard/display interfaces may be implemented in software (programs); however, in cases where these functions place a severe load on the processor in terms of time or program memory, special peripheral interface circuits such as the 8251, Universal Synchronous or Asychronous Receiver/Transmitter (USART) or 8279 Keyboard/Display interface may be used.

1.3.4 Code Generation

The writing of the final program code for the application can begin once the system function and hardware have been defined and can be generated in parallel with the detailed hardware design (PC card layout, power supply, etc.)

At this point, there are two paths available to the designer/programmer and two types of design development aids provided by Intel to simplify the procedures. One system, called PROMPT 48, is a low cost development system which supports machine language programming and the second is the Intellec Microcomputer Development System which supports both machine and assembly languages. For those of you unfamiliar with the advantages and disadvantages of machine and assembly languages see Section 1.2.

1.3.5 PROMPT 48

PROMPT 48 is a low cost design aid consisting of: an 8748 processor to execute programs, control circuitry to provide debug functions such as single step and break points, a monitor program stored in ROM, an EPROM programmer, and a hexadecimal keyboard and display. There are two processor sockets on the front of PROMPT 48, one for programming the 8748 and one in which a programmed 8748 executes its program while under control of the monitor routine.

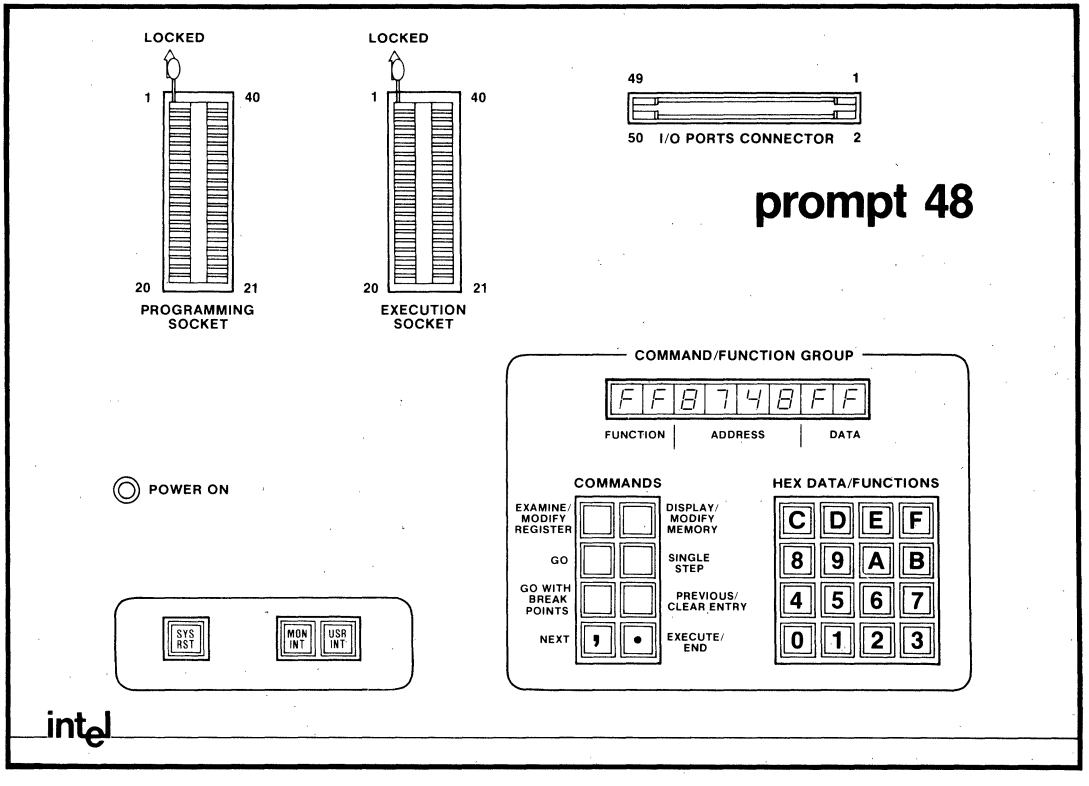
Use of PROMPT 48 involves the following steps:

1. Loading an application program into the PROMPT RAM memory via Hex keyboard or external terminal (TTY and RS232 interface provided).

2. Inserting an erased 8748 in the programming socket and transferring the application program to its internal EPROM.

3. Transferring programmed 8748 to execution socket where program is executed and debugged under control of the monitor.

The monitor routine allows the user to single step this processor, examine or modify all internal registers and data memory; or to run at full speed and stop the processor at predetermined breakpoints. PROMPT 48



also provides 1K of writeable program memory which may be used to debug user programs. A multiple single step feature is also provided in which the processor steps through its program dumping all internal contents to external RAM where it may be later displayed or typed out on an external terminal. Paper tape input and output in Intel's hexadecimal format is also available through the TTY.

1.3.6 Intellec Development System

The Intellec Microcomputer Development System is a modular development system which can be expanded as necessary to meet the requirements of your design cycle. The system consists of the processor unit which is based on Intel's 8080A microprocessor, and several optional units such as the UPP Universal PROM Programmer, the PTR High Speed Paper tape reader, the DOS Disk Operating System, and the Intellec CRT terminal.

To support the development of MCS-48 systems a macro-assembler ASM 48 is available for the Intellec System as well as a personality module for the UPP which will program the EPROM of the 8748. Also available is in-circuit emulation capability with ICE-49 which will allow emulation and debug of user's 8048 application programs on the 8080A-based Intellec Development System.

The Intellec system is a flexible high performance development system which can support Intel's various microcomputer families with various optional modules. The macro-assembler and text editor programs provided allow the designer to write and edit his programs in assembly language and then generate the machine language output necessary to program the 8748 EPROM. The availability of a high speed CRT and a diskette operating system eliminates the laborious input and output of paper tape files normally required during the assembly process. Finally, ICE 48 allows the user to extend the resources of his entire Intellec system into the 8048 socket of his own system and use all its emulation, debug, and display facilities directly.

1.3.7 Production

Once a working program has been achieved, a preproduction phase usually follows where several prototype systems are evaluated in simulated situations or in actual operation in the field. During this period the use of the 8748 EPROM allows quick alteration of the application program when problems or suggested changes arise. Depending on the magnitude and number of future changes anticipated, the first production units may also be shipped with EPROM processor. However, to achieve the maximum cost reduction potential in high volume applications, a conversion to the 8048 ROM is usually necessary. This is an easy transition since the 8048 and 8748 are pin and machine code compatible equivalents. The user merely develops a hexadecimal tape of his 8748 program memory contents using his Intellec System or PROMPT 48 development aid and sends it to Intel along with his 8048 order. As the 8048 ROM's arrive they can immediately replace the 8748 EPROMs.

The Single Component MCS-48™ System

2

THE SINGLE COMPONENT MCS-48™ SYSTEM

2.0 Summary

Sections 2.1 through 2.4 describe in detail the functional characteristics of the 8748 EPROM, 8048/8049 ROM and 8035/8039 single component microcomputers. Unless otherwise noted, details within these sections apply to all versions. Sections 2.5 through 2.11 describe the operation of the 8021, while Sections 2.12 through 2.21 describe the 8022. This chapter is limited to those functions useful in single-chip implementations of the MCS-48. Chapter 3 discusses functions which allow expansion of program memory, data memory, and input-output capability.

2.1 Architecture

The following sections break the 8048 into functional blocks and describe each in detail.

2.1.1 Arithmetic Section

The arithmetic section of the processor contains the basic data manipulation functions of the 8048 and can be divided into the following blocks:

Arithmetic Logic Unit (ALU)
Accumulator
Carry Flag
Instruction Decoder

In a typical operation data stored in the accumulator is combined in the ALU with data from another source on the internal bus (such as a register or I/O port) and the result is stored in the accumulator or another register. The following is a more detailed description of the function of each block:

Instruction Decoder

The operation code (op code) portion of each program instruction is stored in the Instruction Decoder and converted to outputs which control the function of each of the blocks of the Arithmetic Section. These lines control the source of data and the destination register as well as the function performed in the ALU.

Arithmetic Logic Unit

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under control of the Instruction Decoder. The ALU can perform the following functions:

Add With or Without Carry
AND, OR, Exclusive OR
Increment/Decrement
Bit Complement
Rotate Left, Right
Swap Nibbles
BCD Decimal Adjust

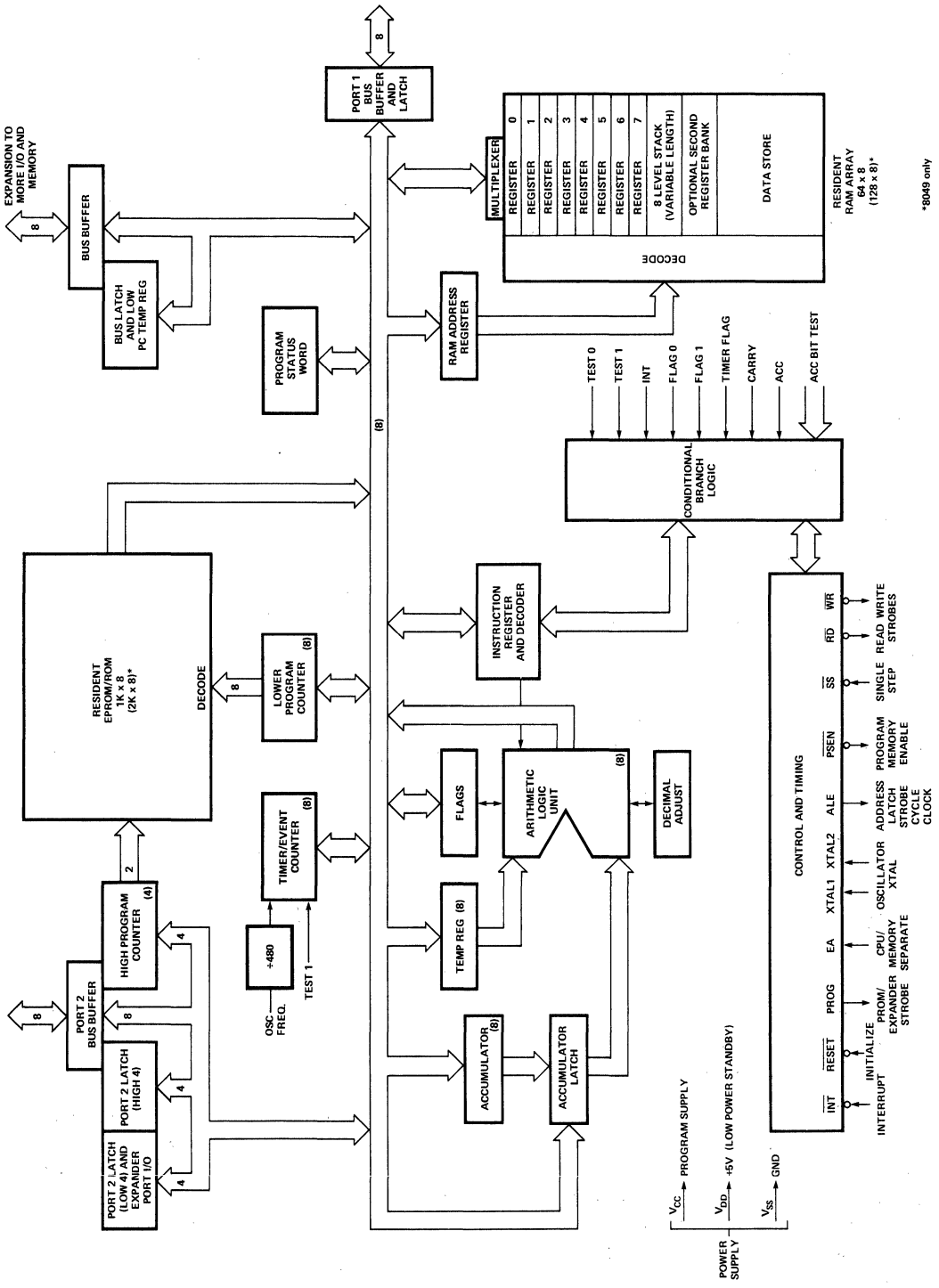
If the operation performed by the ALU results in a value represented by more than 8 bits (overflow of most significant bit) a Carry Flag is set in the Program Status Word.

Accumulator

The accumulator is the single most important data register in the processor, being one of the sources of input to the ALU and often the destination of the result of operations performed in the ALU. Data to and from I/O ports and memory also normally passes through the accumulator.

2.1.2 Program Memory

Resident program memory consists of 1024 or 2048 words eight bits wide which are addressed by the program counter. In the 8748 this memory is user programmable and erasable EPROM; in the 8048/8049 the memory is ROM which is mask programmable at the factory. The 8035/8039 has no internal program memory and is used with external devices. Program code is completely interchangeable among the various versions. See Section 2.3 for EPROM programming techniques.



8048/8049 BLOCK DIAGRAM

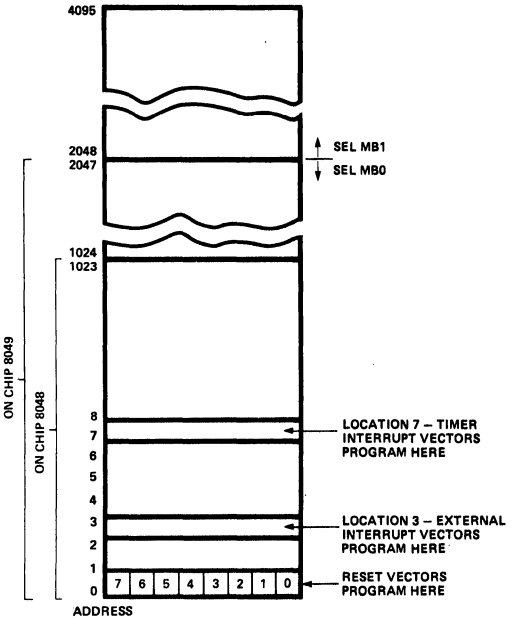
There are three locations in Program Memory of special importance:

LOCATION 0
Activating the Reset line of the processor causes the first instruction to be fetched from location 0.

LOCATION 3
Activating the Interrupt input line of the processor (if interrupt is enabled) causes a jump to subroutine at location 3.

LOCATION 7
A timer/counter interrupt resulting from timer/counter overflow (if enabled) causes a jump to subroutine at location 7.

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service subroutine is stored in location 3, and the first word of a timer/counter service routine is stored in location 7. Program memory can be used to store constants as well as program instructions. Instructions such as MOVP and MOVP3 allow easy access to data "lookup" tables.

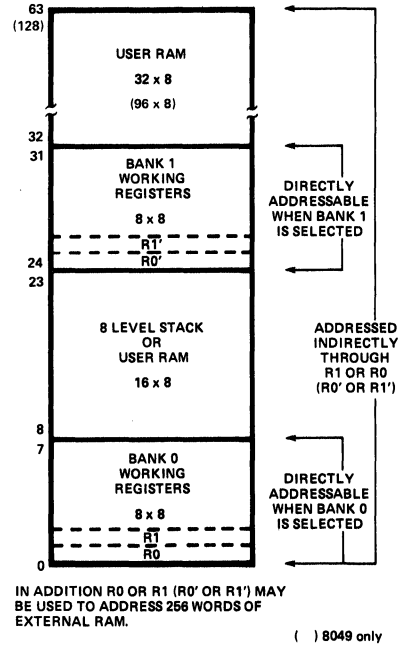


MCS-48™ PROGRAM MEMORY MAP

Data Memory

Resident data memory is organized as 64 or 128 words 8-bits wide. All locations are indirectly addressable through either of two RAM Pointer Registers which reside at address 0 and 1 of the register array. In addition, the first 8 locations (0-7) of the array are designated as working registers and are directly addressable by several instructions. Since these registers are more easily addressed, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

By executing a Register Bank Switch instruction (SEL RB) RAM locations 24-31 are designated as the working registers in place of locations 0-7 and are then directly addressable. This second bank of working registers may be used as an extension of the first bank or reserved for use during interrupt service



DATA MEMORY MAP

subroutines allowing the registers of Bank 0 used in the main program to be instantly "saved" by a Bank Switch. Note that if this second bank is not used, locations 24-31 are still addressable as general purpose RAM. Since the two RAM pointer Registers R0 and R1 are a part of the working register array, bank switching effectively creates two more pointer registers (R0' and R1') which can be used with R0 and R1 to easily access up to four separate working areas in Ram at one time. RAM locations (8-23) also serve a dual role in that they contain the program counter stack as explained in Sec. 2.1.6. These locations are addressed by the Stack Pointer during subroutine calls as well as by RAM Pointer Registers R0 and R1. If the level of subroutine nesting is less than 8, all stack registers are not required and can be used as general purpose RAM locations. Each level of subroutine nesting not used provides the user with two additional RAM locations.

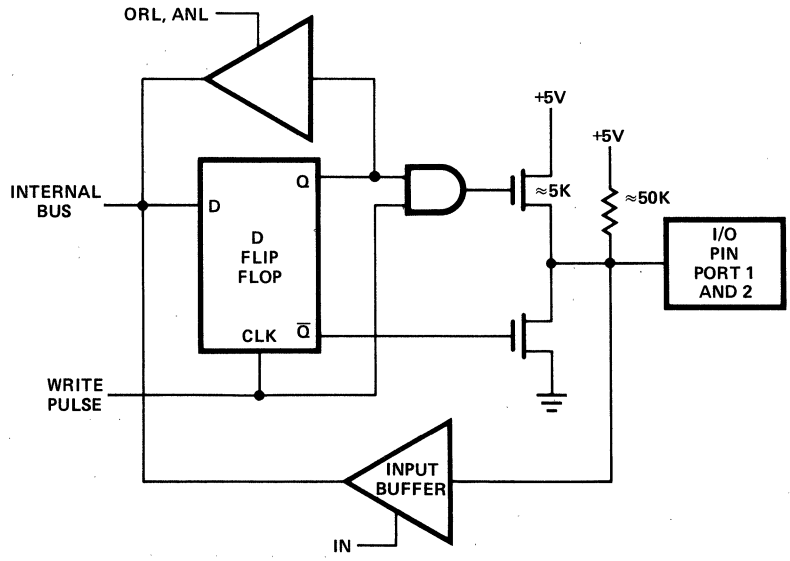
2.1.4 Input/Output

The 8048 has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs or bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non latching, i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The figure shows the circuit configuration in detail. Each line is continuously pulled up to +5v through a resistive device of relatively high impedance (~50KΩ). This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition a relatively low



"QUASI BI DIRECTIONAL" PORT STRUCTURE

impedance device (~5KΩ) is switched in momentarily (~500ns) whenever a "1" is written to the line. When a "0" is written to the line a low impedance (~300Ω) device overcomes the light pullup and provides TTL current sinking capability. Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state. This structure allows input and output on the same pin and also allows a mix of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor. See also Section 3.7.

Bus

Bus is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, Bus can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed however.

As a static port, data is written and latched using the OUTL instruction and inputted using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$. When not being written or read, the BUS lines are in a high impedance state. See also Sections 3.6 and 3.7.

2.1.5 Test and $\overline{INT}$ Inputs

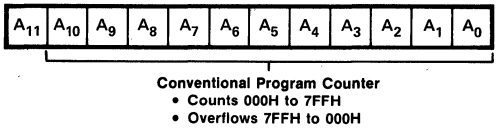
Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and $\overline{INT}$. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and $\overline{INT}$ pins have other possible functions as well. See the pin description in Sec. 2.2.

2.1.6 Program Counter and Stack

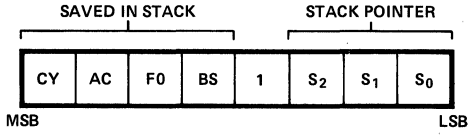
The Program Counter is an independent counter while the Program Counter Stack is implemented using pairs of registers in the Data Memory Array. Only 10 (or 11) bits of the Program Counter are used to address the 1024 (2048) words of on-board program memory while the most significant bits are used for external Program Memory fetches. The Program Counter is initialized to zero by activating the Reset line.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit Stack Pointer which is part of the Program Status Word (PSW). Data RAM locations 8 thru 23 are available as stack registers and are used to store the Program Counter and 4 bits of PSW as shown in the figure. The Stack Pointer when initialized to 000 points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the Stack Pointer to be decremented and the contents of the resulting register pair to be transferred to the Program Counter.

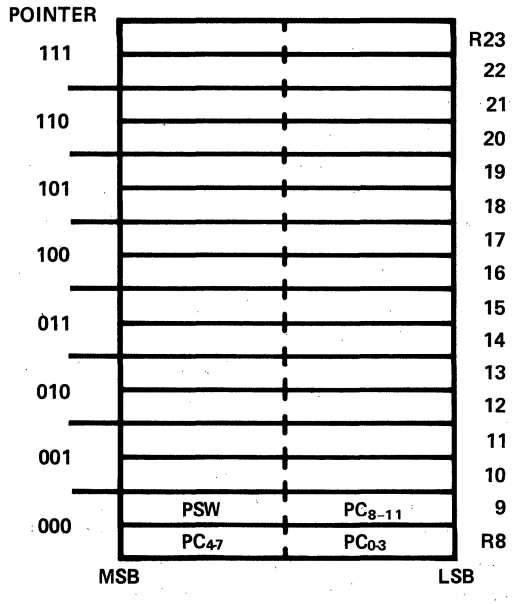


PROGRAM COUNTER



CY CARRY
AC AUXILLARY CARRY
F0 FLAG 0
BS REGISTER BANK SELECT

PROGRAM STATUS WORD (PSW)



PROGRAM COUNTER STACK

2.1.7 Program Status Word

An 8-bit status word which can be loaded to and from the accumulator exists called the Program Status Word (PSW). The accompanying figure shows the information available in the word. The Program Status Word is actually a collection of flip-flops throughout the machine which can be read or written as a whole. The ability to write to PSW allows for easy restoration of machine status after a power down sequence.

The upper four bits of PSW are stored in the Program Counter Stack with every call to subroutine or interrupt vector and are optionally restored upon return with the RETR instruction. The RET return instruction does not update PSW.

The PSW bit definitions are as follows:

Bits 0 - 2: Stack Pointer bits (S_0, S_1, S_2)

Bit 3: Not used ("1" level when read)

Bit 4: Working Register Bank Switch Bit (BS)
0 = Bank 0
1 = Bank 1

Bit 5: Flag 0 bit (F0) user controlled flag which can be complemented or cleared, and tested with the conditional jump instruction JF0.

Bit 6: Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A.

Bit 7: Carry (CY) carry flag which indicates that the previous operation has resulted in overflow of the accumulator.

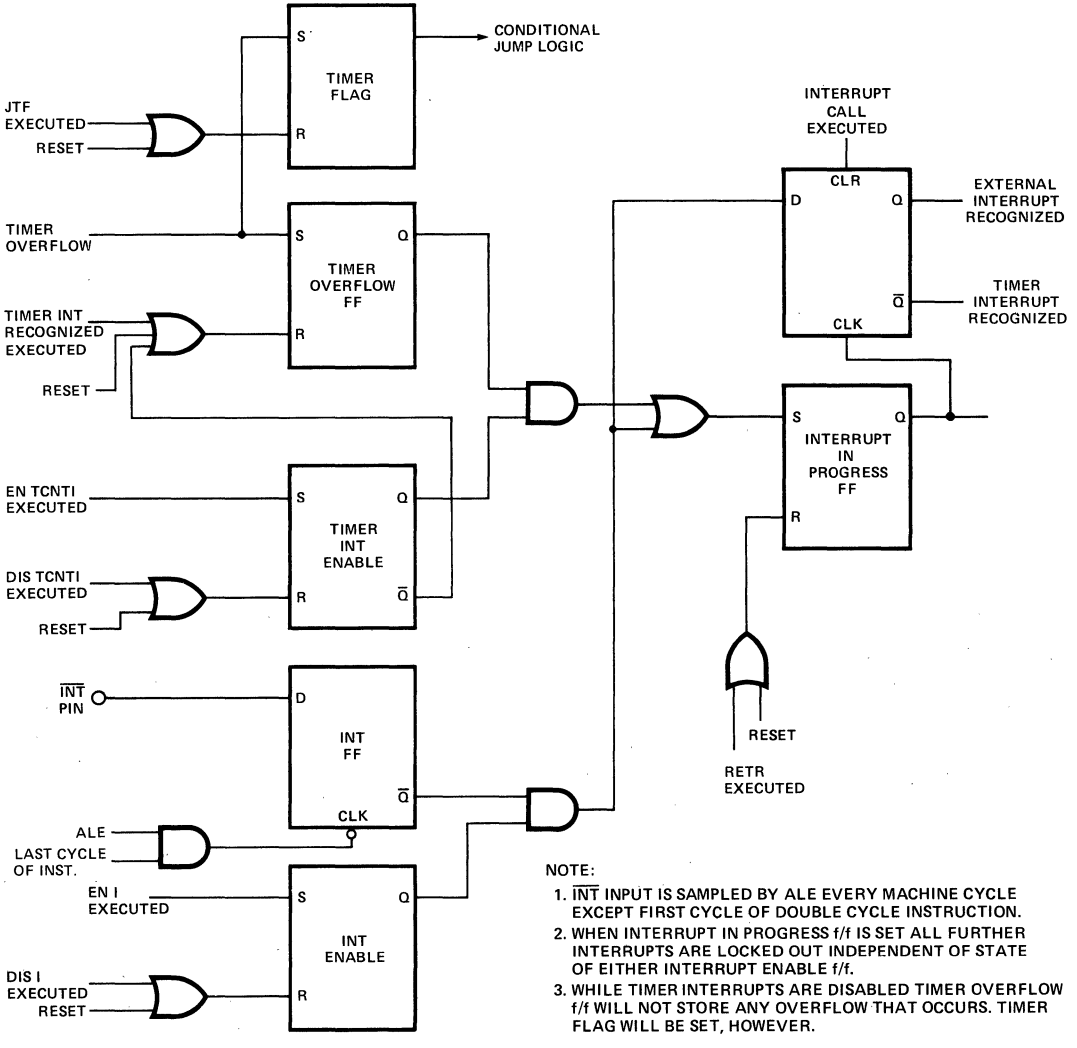
2.1.8 Conditional Branch Logic

The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. By using the conditional jump instruction the following conditions can effect a change in the sequence of the program execution.

Device Testable	Jump Conditions (Jump On)	
	All zeros	not all zeros
Accumulator	—	1
Accumulator Bit	0	1
Carry Flag	—	1
User Flags (F0, F1)	—	1
Timer Overflow Flag	0	1
Test Inputs (T0, T1)	0	1
Interrupt Input ($\overline{INT}$)	0	—

2.1.9 Interrupt

An interrupt sequence is initiated by applying a low "0" level input to the $\overline{INT}$ pin. Interrupt is level triggered and active low to allow "WIRE ORing" of several interrupt sources at the input pin. The Interrupt line is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at location 3 in program memory as soon as all cycles of the current instruction are complete. $\overline{INT}$ must be held low for at least T_{cy} to ensure proper



NOTE:

1. $\overline{INT}$ INPUT IS SAMPLED BY ALE EVERY MACHINE CYCLE EXCEPT FIRST CYCLE OF DOUBLE CYCLE INSTRUCTION.
2. WHEN INTERRUPT IN PROGRESS f/f IS SET ALL FURTHER INTERRUPTS ARE LOCKED OUT INDEPENDENT OF STATE OF EITHER INTERRUPT ENABLE f/f.
3. WHILE TIMER INTERRUPTS ARE DISABLED TIMER OVERFLOW f/f WILL NOT STORE ANY OVERFLOW THAT OCCURS. TIMER FLAG WILL BE SET, HOWEVER.

interrupt operations. As in any CALL to subroutine, the Program Counter and Program Status word are saved in the stack. For a description of this operation see the previous section, Program Counter and Stack. Program Memory location 3 usually contains an unconditional jump to an interrupt service subroutine elsewhere in program memory. The end of an interrupt service subroutine is signalled by the execution of a Return and Restore Status instruction RETR. The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR reenables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. This sequence holds true also for an internal interrupt generated by timer overflow. If an internal timer/counter generated interrupt and an external interrupt are detected at the same time, the external source will be recognized. See the following Timer/Counter section for a description of timer interrupt. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the Counter (one less than terminal count), and enabling the event counter mode. A "1" to "0" transition on the T1 input will then cause an interrupt vector to location 7.

Interrupt Timing

The interrupt input may be enabled or disabled under Program Control using the EN I and DIS I instructions. Interrupts are disabled by Reset and remain so until enabled by the users program. An interrupt request must be removed before the RETR instruction is executed upon return from the service routine otherwise the processor will re-enter the service routine immediately. Many peripheral devices prevent this situation by resetting their interrupt request line whenever the processor accesses (Reads or Writes) the peripherals data buffer register. If the interrupting device does not require access by the processor, one output line of the 8048 may be designated as an "interrupt acknowledge" which is activated by the service subroutine to reset the interrupt request. The $\overline{INT}$ pin may also be tested using the conditional jump instruction JNI. This instruction may be used to detect the presence of a pending interrupt before interrupts are enabled. If interrupt is left disabled, $\overline{INT}$ may be used as another test input like T0 and T1.

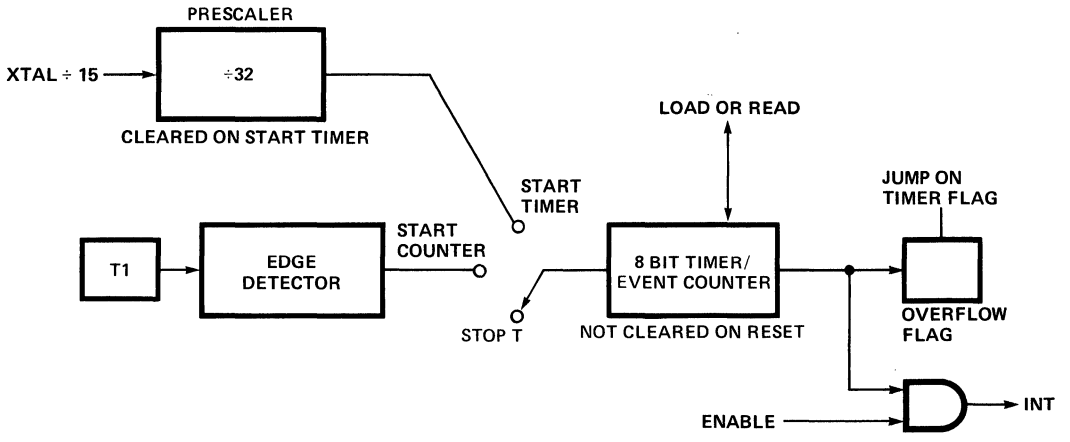
2.1.10 Timer/Counter

The 8048 contains a counter to aid the user in counting external events and generating accurate time delays without placing a burden on the processor for these functions. In both modes the counter operation is the same, the only difference being the source of the input to the counter.

Counter

The 8-bit up binary counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice versa. The counter content is not affected by Reset and is initialized solely by the MOV T,A instruction. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started as a timer by a START T instruction or as an event counter by a START CNT instruction. Once started the counter will increment to its maximum count (FF) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or Reset.

The increment from maximum count to zero (overflow) results in the setting of an overflow flag flip-flop and in the generation of an interrupt request. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by executing a JTF or by Reset. The interrupt request is stored in a latch and then ORed with the external interrupt input $\overline{INT}$. The timer interrupt may be enabled or disabled independently of external interrupt by the EN TCNTI and DIS TCNTI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer or counter service routine may be stored. If timer and external interrupts occur simultaneously, the external source will be recognized and the Call will be to location 3. Since the timer interrupt is latched it will remain pending until the external device is



TIMER/EVENT COUNTER

serviced and immediately be recognized upon return from the service routine. The pending timer interrupt is reset by the Call to location 7 or may be removed by executing a DIS TCNTI instruction.

As an Event Counter

Execution of a START CNT instruction connects the T1 input pin to the counter input and enables the counter. Subsequent high to low transitions on T1 will cause the counter to increment. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 7.5μsec when using a 6MHz crystal)—there is no minimum frequency. T1 input must remain high for at least 500ns (at 6MHz) after each transition.

As a Timer

Execution of a START T instruction connects an internal clock to the counter input and enables the counter. The internal clock is derived by passing the basic 400 KHz machine cycle clock ALE through a ÷ 32 prescaler. The prescaler is reset during the START T instruction. The resulting 12.5 KHz clock increments the counter every 80 μsec (assuming 6 MHz XTAL). Various delays between 80 μsec and 20 msec (256 counts) can be obtained by presetting the counter and detecting overflow. Times longer than 20 msec may be achieved by accumulating multiple overflows in a register under software control. For time resolution less than 80 μsec an external clock can be applied to the T1 input and the counter operated in the event counter mode. ALE divided by 3 or more can serve as this external clock. Very small delays or "fine tuning" of larger delays can be easily accomplished by software delay loops.

2.1.11 Clock and Timing Circuits

Timing generation for the 8048 is completely self-contained with the exception of a frequency reference which can be XTAL, inductor, or external clock source. The Clock and Timing circuitry can be divided into the following functional blocks:

Oscillator

The on-board oscillator is a high gain parallel resonant circuit with a frequency range of 1 to 6MHz. The X1 external pin is the input to the amplifier stage while X2 is the output. A crystal or inductor connected between X1 and X2 provides the feedback and phase shift required for oscillation. A 5.9904 MHz crystal provides for easy derivation of all standard communications frequencies. If an accurate frequency reference and maximum processor speed are not required, an induc-

tor may be used in place of the crystal. With an inductor the oscillator frequency can be approximately 3 to 5 MHz. For higher speed operation a crystal should be used. An externally generated clock may also be applied to X1-X2 as the frequency source. See the data sheet for more information.

State Counter

The output of the oscillator is divided by 3 in the State Counter to create a clock which defines the state times of the machine (CLK). CLK can be made available on the external pin T0 by executing an ENTO CLK instruction. The output of CLK on T0 is disabled by Reset of the processor.

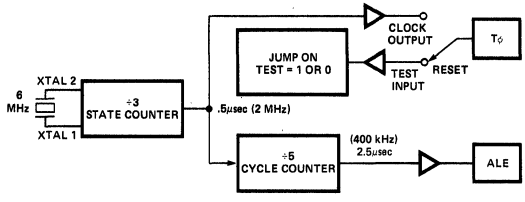
Cycle Counter

CLK is then divided by 5 in the Cycle Counter to provide a clock which defines a machine cycle consisting of 5 machine states. This clock is called Address Latch Enable (ALE) because of its function in MCS-48 systems with external memory. It is provided continuously on the ALE output pin.

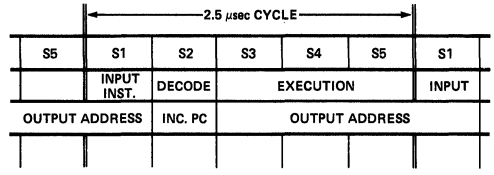
2.1.12 Reset

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pullup resistor which in combination with an external 1 μfd capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset. If the

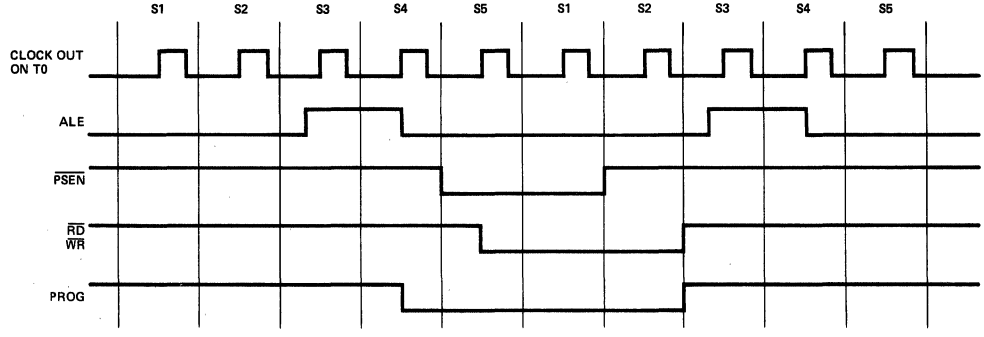
DIAGRAM OF 8048 CLOCK UTILITIES



INSTRUCTION CYCLE



MCS-48™ CYCLE TIMING



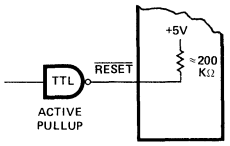
	CYCLE 1					CYCLE 2				
INSTRUCTION	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	——	——	READ PORT	* ——	——	——
OUTL P,A	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	OUTPUT TO PORT	——	——	* ——	——	——
ANL P, #DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
ORL P, #DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
INS A, BUS	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	——	——	READ PORT	* ——	——	——
OUTL BUS, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	OUTPUT TO PORT	——	——	* ——	——	——
ANL BUS, #DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
ORL BUS, #DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
MOVX @ R,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	OUTPUT DATA TO RAM	——	——	* ——	——	——
MOVX A,@R	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	——	——	READ DATA	* ——	——	——
MOVD A, P_i	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	——	——	READ P2 LOWER	* ——	——	——
MOVD P_i, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA TO P2 LOWER	——	——	* ——	——	——
ANLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	——	——	* ——	——	——
ORLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	——	——	* ——	——	——
J (CONDITIONAL)	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	SAMPLE CONDITION	INCREMENT TIMER	——	FETCH IMMEDIATE DATA	——	* UPDATE PROGRAM COUNTER	——	——
STRT T STRT CNT	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	——	START COUNTER					
STOP TCNT	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	——	STOP COUNTER					
ENI	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	ENABLE INTERRUPT	——					
DIS I	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	DISABLE INTERRUPT	——					
ENTO CLK	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	ENABLE CLOCK	——					

***VALID INSTRUCTION ADDRESSES ARE OUTPUT AT THIS TIME IF EXTERNAL PROGRAM MEMORY IS BEING ACCESSED.**

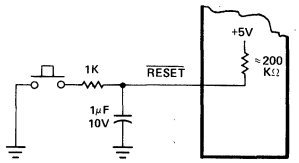
INSTRUCTION TIMING DIAGRAM

reset pulse is generated externally the RESET pin must be held at ground (.5V) for at least 10 milliseconds after the power supply is within tolerance. Only 5 machine cycles (12.5μs @ 6 MHz) are required if power is already on and the oscillator has stabilized.

EXTERNAL RESET



POWER ON RESET



Reset performs the following functions:

1. Sets program counter to zero.
2. Sets stack pointer to zero.
3. Selects register bank 0.
4. Selects memory bank 0.
5. Sets BUS to high impedance state. (except when EA = 5V)
6. Sets Ports 1 and 2 to input mode.
7. Disables interrupts (timer and external)
8. Stops timer.
9. Clears timer flag.
10. Clears F0 and F1.
11. Disables clock output from T0.

2.1.13 Single-Step

This feature provides the user with a debug capability in that the processor can be stepped through the program one instruction at a time. While stopped, the address of the next instruction to be fetched is available concurrently on BUS and the lower half of Port 2. The user can therefore follow the program through each of the instruction steps. A timing diagram, showing the interaction between output ALE and input $\overline{SS}$ is shown. The BUS buffer contents are lost during single step; however, a latch may be added to re-establish the lost I/O capability if needed. (See 2.4.1).

Timing

The 8048 operates in a single-step mode as follows:

1. The processor is requested to stop by applying a low level on $\overline{SS}$.

2. The processor responds by stopping during the instruction fetch portion of the next instruction. If a double cycle instruction is in progress when the single step command is received, both cycles will be completed before stopping.

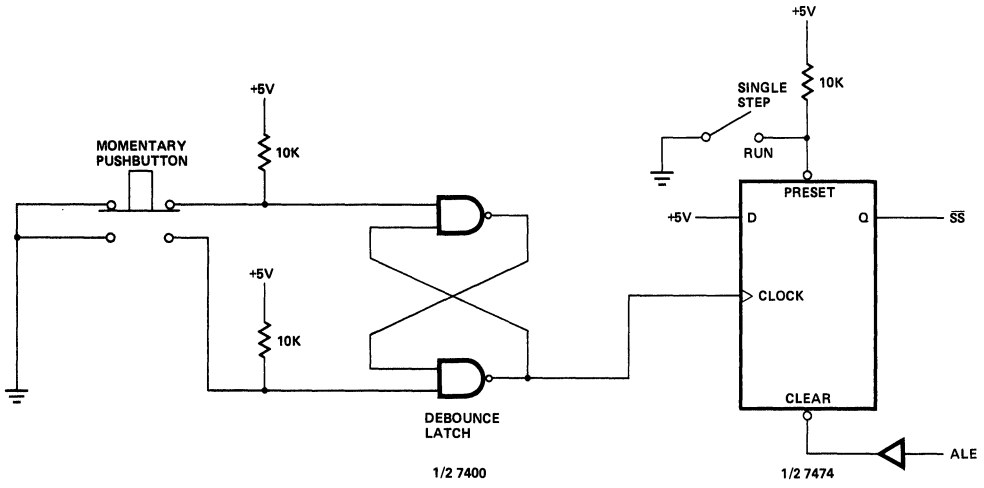
3. The processor acknowledges it has entered the stopped state by raising ALE high. In this state (which can be maintained indefinitely) the address of the next instruction to be fetched is present on BUS and the lower half of port 2.

4. $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction. The exit from stop is indicated by the processor bringing ALE low.

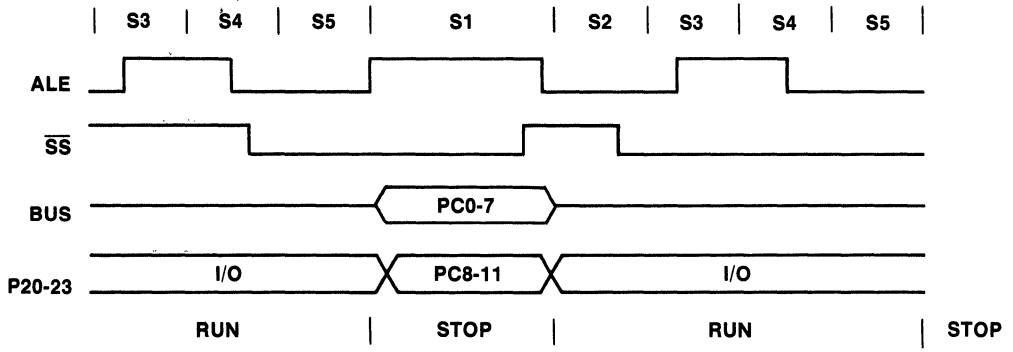
5. To stop the processor at the next instruction $\overline{SS}$ must be brought low again as soon as ALE goes low. If $\overline{SS}$ is left high the processor remains in a "Run" mode.

A diagram for implementing the single step function of the 8748 is shown. A D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the run mode $\overline{SS}$ is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single step, preset is removed allowing ALE to bring $\overline{SS}$ low via the clear input. ALE should be buffered since the clear input of an SN7474 is the equivalent of 3 TTL loads. The processor is now in the stopped state. The next instruction is initiated by clocking a "1" into the flip-flop. This "1" will not appear on $\overline{SS}$ unless ALE is high removing clear from the flip-flop. In response to $\overline{SS}$ going high the processor begins an instruction fetch which brings ALE low resetting $\overline{SS}$ through the clear input and causing the processor to again enter the stopped state.

SINGLE STEP CIRCUIT



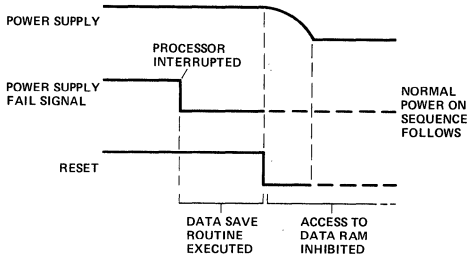
SINGLE STEP TIMING



2.1.14 Power Down Mode (8048, 8049, 8039, 8035L)

Extra circuitry has been added to the 8048 ROM version to allow power to be removed from all but the 64/128 x 8 data RAM array for low power standby operation. In the power down mode the contents of data RAM can be maintained while drawing typically 10% to 15% of normal operating power requirements.

V_{CC} serves as the 5V supply pin for the bulk of 8048 circuitry while the V_{DD} pin supplies only the RAM array. In normal operation both pins are at 5V while in standby V_{CC} is at ground and only V_{DD} is maintained at 5V. Applying Reset to the processor through the $\overline{RESET}$ pin inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from V_{CC}.



POWER DOWN SEQUENCE

A typical power down sequence occurs as follows:

1. Imminent power supply failure is detected by user defined circuitry. Signal must be early enough to allow 8048 to save all necessary data before V_{CC} falls below normal operating limits.

2. Power fail signal is used to interrupt processor and vector it to a power fail service routine.

3. Power fail routine saves all important data and machine status in the internal data RAM array. Routine may also initiate transfer of backup supply to the V_{DD} pin and indicate to external circuitry that power fail routine is complete.

4. Reset is applied to guarantee data will not be altered as the power supply falls out of limits. Reset must be held low until V_{CC} is at ground level.

Recovery from the Power Down mode can occur as any other power-on sequence with an external capacitor on the Reset input providing the necessary delay. See the previous section on Reset.

2.1.15 External Access Mode

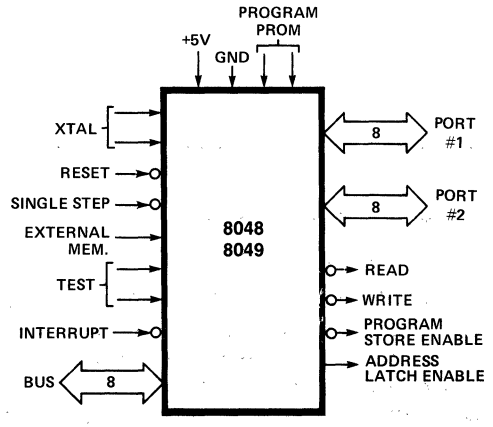
Normally the first 1K (8048) or 2K (8049) words of program memory are automatically fetched from internal ROM or EPROM. The EA input pin however allows the user to effectively disable internal program memory by forcing all program memory fetches to reference external memory. The following chapter explains how access to external program memory is accomplished.

The External Access mode is very useful in system test and debug because it allows the user to disable his internal applications program and substitute an external program of his choice—a diagnostic routine for instance. In addition, the section on Test and Debug explains how internal program memory can be read externally, independent of the processor.

A "1" level on EA initiates the external access mode. For proper operation, Reset should be applied while the EA input is changed.

2.2 Pin Description

The MCS-48 processors (except 8021) are packaged in 40 pin Dual In-Line Packages (DIP's). The following is a summary of the functions of each pin. Where it exists, the second paragraph describes each pin's function in an expanded MCS-48 system. Unless otherwise specified, each input is TTL compatible and each output will drive one standard TTL load.



8048 LOGIC SYMBOL

Designation	Pin* Number	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 ROM version
V_{CC}	40	Main power supply; +5V during operation and 8748 programming.
PROG	25	Program pulse (+23V) input pin during 8748 programming. Output strobe for 8243 I/O expander.
P10-P17 (Port 1)	27-34	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ)
P20-P27 (Port 2)	21-24 35-38	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ) P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
D0-D7 (BUS)	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENTO CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the event counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. (Active low) Used as a Read Strobe to External Data Memory.

*8048, 8748, 8049

Designation	Pin Number	Function
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during PROM programming and verification. (Active low) (Internal pullup ≈ 200KΩ)
$\overline{WR}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to external data memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active Low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active Low) (Internal pullup ≈ 300KΩ)
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active High) (Internal pullup ≈ 10MΩ on 8048/8049, 8035L, 8039 only)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source.
XTAL2	3	Other side of crystal/external source input.

Unless otherwise stated inputs do not have internal pullup resistors.

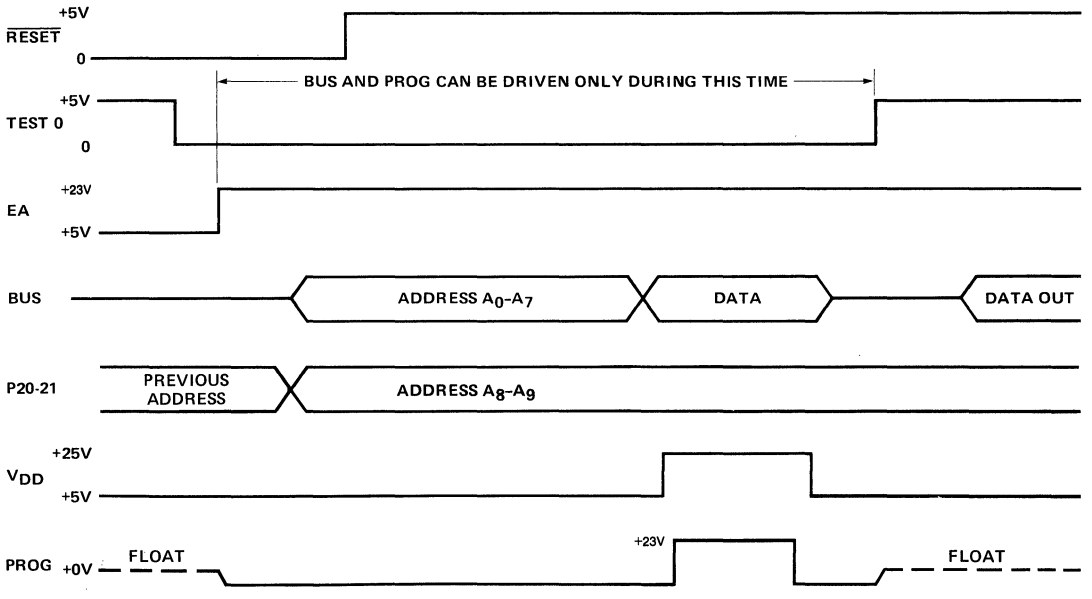
2.3 Programming, Verifying and Erasing EPROM

The internal Program Memory of the 8748 may be erased and reprogrammed by the user as explained in the following sections. See also the 8748 data sheet.

2.3.1 Programming/Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
$\overline{Reset}$	Initialization and Address Latching
Test 0	Selection of Program (0V) or Verify (5V) Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input



SEE 8048/8748 DATA SHEET (CHAPTER 6) FOR DETAIL TIMING SPECIFICATIONS.

WARNING: An attempt to program a mis-socketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

PROGRAMMING/VERIFY SEQUENCE

8748 Erasure Characteristics

The erasure characteristics of the 8748 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8748 in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8748 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8748 window to prevent unintentional erasure.

When erased, bits of the 8748 Program Memory are in the logic "0" state.

The recommended erasure procedure for the 8748 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8748 should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

The Program/Verify sequence is:

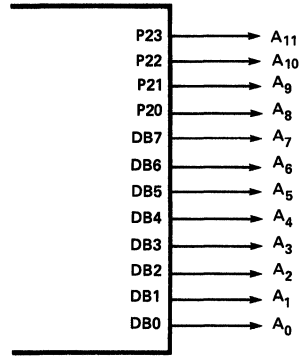
1. V_{DD} = 5v, Clock applied or internal oscillator operating, $\overline{Reset}$ = 0v Test 0 = 5v, EA = 5v, BUS and PROG floating
2. Insert 8748 in programming socket
3. Test 0 = 0v (Select Program Mode)
4. EA = 23V (Activate Program Mode)
5. Address applied to BUS and P20-1
6. $\overline{Reset}$ = 5v (Latch Address)
7. Data applied to BUS
8. V_{DD} = 25v (Programming Power)
9. PROG = 0v followed by one 50ms pulse to 23V
10. V_{DD} = 5v
11. TEST 0 = 5v (Verify Mode)
12. Read and Verify Data on BUS
13. TEST 0 = 0v
14. $\overline{Reset}$ = 0v and repeat from Step 5
15. Programmer should be at conditions of Step 1 when 8748 is removed from socket.

2.4 Test and Debug

Several MCS-48 features described in the previous sections are discussed here to emphasize their use in testing MCS-48 components and in debugging MCS-48 based systems.

2.4.1 Single Step

Single step circuitry within the microcomputer in combination with the external circuitry described in Section 2.1.13 allows the user to execute one instruction at a time whether the instruction is one or two cycles in length. After completion of the instruction the processor halts with the address of the next instruction to be fetched available on the eight lines of BUS and the lower 4-bits of port 2.



ADDRESS OUTPUT DURING SINGLE STEP

This allows the user to step through his program and note the sequence of instructions being executed.

While the processor is stopped, the I/O information on BUS and the 4-bits of port 2 is, of course, not available. I/O information is, however, valid at the leading edge of ALE and can be latched externally using this signal if necessary.

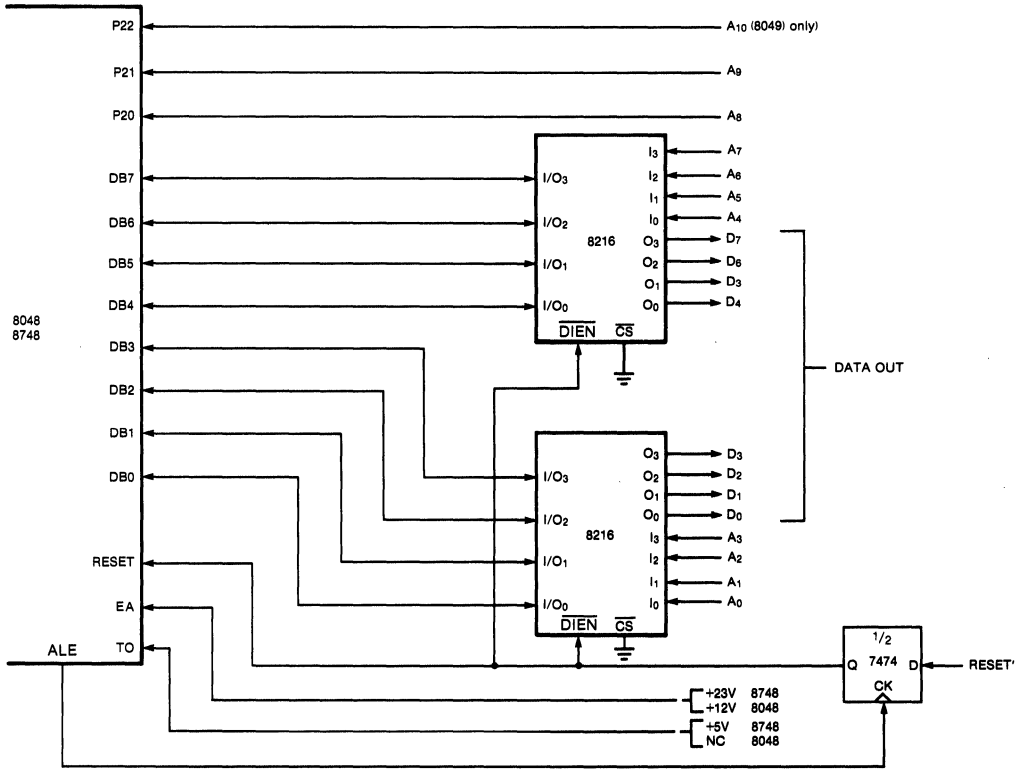
2.4.2 Disabling Internal Program Memory

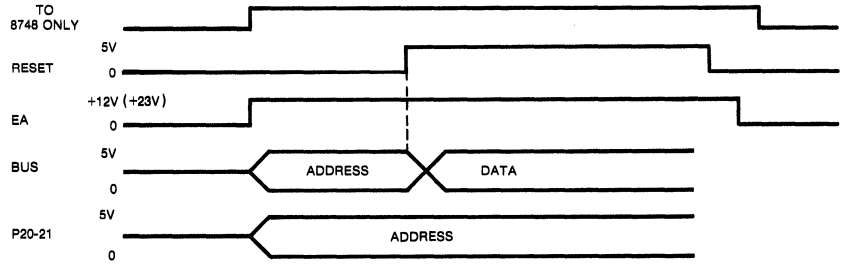
Applying +5V to the EA (external access) pin of the MCS-48 microcomputers allows the user to effectively disable internal program memory by forcing all instruction fetches to occur from an external memory. This external memory can be connected as explained in the section on program memory expansion and can contain a diagnostic routine to exercise the processor, the internal RAM, the timer, and the I/O lines. EA should be switched only when the processor is in RESET.

2.4.3 Reading Internal Program Memory

Just as the processor may be isolated from internal program memory using EA, program memory can be read independent of the processor using the verification mode described in the previous section, Programming/Verification.

The processor is placed in the READ mode by applying a high voltage (+23V for the 8748, +12V for the 8048/8049) to the EA pin and +5V to the TO (8748 only) input pin. $\overline{\text{RESET}}$ must be at 0V when voltage is applied to EA. The address of the location to be read is then applied to the same lines (TTL levels) of BUS and Port 2 which output the address during single step (see below). The address is latched by a "0" to "1" transition on $\overline{\text{RESET}}$ and a high level on $\overline{\text{RESET}}$ causes the contents of the program memory location addressed to appear on the eight lines of BUS. $\overline{\text{RESET}}$ must be brought back to 0V before leaving the READ mode.





8021 Functional Specifications

The following is a functional description of the major elements of the 8021.

2.5 Program Memory

The 8021 contains 1K x 8 of mask programmable ROM. No external ROM expansion capability is provided.

2.6 Data Memory

A 64 x 8 dynamic RAM is located on chip for data storage. All locations are indirectly addressable and eight designated locations are directly addressable. Also, included in the memory is the address stack, addressed by a 3-bit stack pointer.

Memory is organized as shown in Figure 1. The least significant 8 addresses, 0-7, are directly addressable by any of the 11 direct register instructions. The locations are readily accessible for a variety of operations with the least number of instruction bytes required for their manipulation.

Registers 0 and 1 have another function, in that they can be used to indirectly address all locations in memory, using the indirect register instructions. These indirect RAM address registers, IRAR's, are especially useful for repetitive-type operations on adjacent memory locations. The indirect register instruction specifies which IRAR to use, and the contents of the IRAR is used to address a location in RAM. The contents of the addressed location is used during the execution of the instruction and may be modified. A value larger than 63 should not be preset in the IRAR when selected by an indirect register instruction. IRAR's may point to address 0-7, if desired.

Locations 8-23 may be used as the address stack. The address stack enables the processor to keep track of the return addresses generated from CALL instructions. A 3-bit stack pointer (SP) supplies the address of the locations to be loaded with the next return address generated. The SP to this pushdown stack is incremented by one after a return address is stored, and decremented by one before an address is fetched during a RET. The unincremented program counter address is stored in the address stack. The stack contents is incremented before being loaded into the program counter during a return from subroutine. A total of 8 levels of nesting is possible. The SP is initialized to location 8 upon RESET. Since each address is 10-bits long, two bytes must be used to store a single address. The SP is incremented and decremented by one, but each increment or decrement moves the address pointed to by two. Therefore, only even numbered addresses are pointed to.

If a particular application does not require 8 levels of nesting, the unused portion of the stack may be used as any other indirectly addressable scratchpad location. For example, if only 3 levels of subroutine nesting are used, then only locations 8-13 need be reserved for the address stack, and locations 14-63 can be used for data storage.

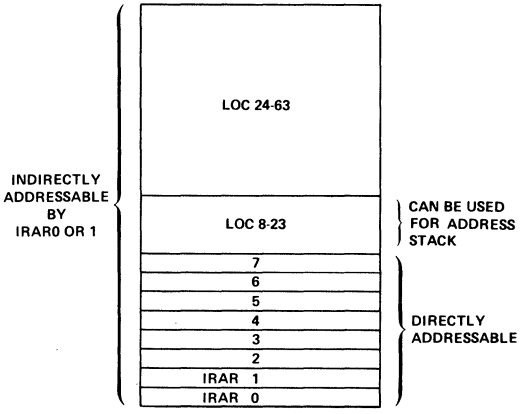


FIGURE 1. INTERNAL RAM ORGANIZATION

2.7 Oscillator and Clock

The 8021 contains its own onboard oscillator and clock circuit, requiring only an external timing control element. This control element can be a crystal, inductor, or clock in. The capacitor normally required in inductor timing control operation is integrated onto the 8021. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. Pins XTAL1 and XTAL2 are used to input the particular control element. An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. (See Figure 2) Therefore, to obtain a 10 μsec instruction cycle, a 3MHz crystal should be used. An oscillator frequency of approximately 3MHz may also be obtained by connecting a 470 μH inductor between XTAL1 and XTAL2. Note that the required inductance may vary and should be adjusted as necessary.

The 8021 utilizes dynamic RAM and certain other dynamic logic. Due to the clocking required with dynamic circuits, the oscillator frequency must be equal to or greater than 600K Hz, or improper operation may occur.

2.8 Timer/Event Counter

The 8021 has internal timer/event counter circuits that can monitor elapsed time or count external events that occur during program execution. The circuit has an 8-bit binary up-counter that is presettable and readable with two MOV instructions. These instructions transfer the contents of the accumulator to the counter and vice-versa. The counter content is not affected by Reset, and is initialized solely by the MOV T,A instruction. The counter is stopped by a RESET or STOP TCNT instruction and remains stopped until started as a timer by a STRT T instruction or as an event counter by a STRT CNT instruction. Once started,

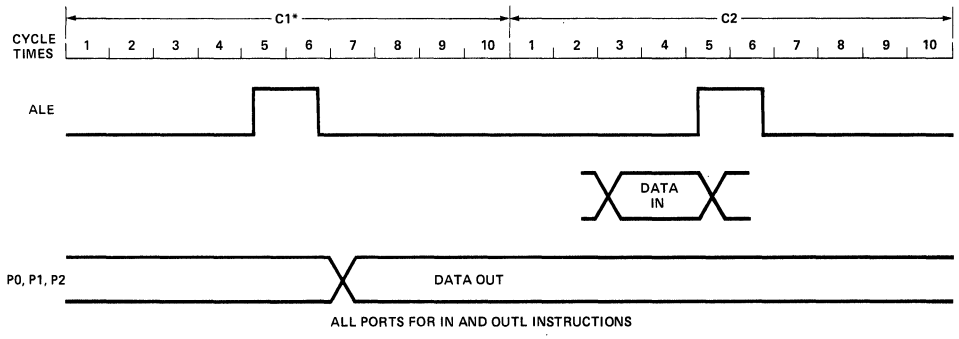


FIGURE 2. 8021 TIMING DIAGRAM

the counter increments to its maximum count (FF), and overflows to zero. The count continues until stopped by a STOP TCNT instruction or RESET. The increment from maximum count to zero (overflow) sets an overflow flag. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by JTF but not by executing a RESET, unlike the 8748.

By a MOV T,A instruction, the contents of the accumulator are loaded to the timer. At the STRT T command an internal prescaler is zeroed and thereafter increments once each 30 input clocks (once each single cycle instruction, twice each double cycle instruction). The prescaler is a divide by 32. At the (11111) to (00000) transition the timer is incremented. The timer is 8-bits and an overflow (FFH) to (00H) timer flag is set. A conditional branch instruction (JTF) is available for testing this flag, the flag being reset each test. Total count capacity for the timer is $2^8 \times 2^5 = 8192$ or 81.9 msec at a 10 μsec cycle time. Contents of the timer are moved to the accumulator by the MOV A,T instruction without disturbing the counting process. The timer stops upon the STOP TCNT instruction.

The STRT CNT instruction connects the T1 input pin to the event counter input and enables the counter. Subsequent high-to-low transitions on T1 increment the counter. The maximum rate at which the counter can increment is once per three instruction cycles (30μs for a 3 MHz oscillator). There is no minimum frequency. T1 input must remain high for at least 500ns after each transition. The event counter is stopped by a STOP TCNT instruction.

2.9 Input/Output Capabilities

The 8021 I/O configurations are highly flexible. A number of different configurations are possible, tailoring an 8021 to a given task. Other than the power supply and dedicated pins, all other pins (20) can be used for input, output, or both, depending on the configuration.

All ports are quasi-bidirectional to facilitate stand-alone use. A simplified schematic of the quasi-bidirectional interface is shown in Figure 3. This configuration allows buffered outputs, and also allows external input. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction. When writing a "0" or low value to these ports, the large pulldown device sinks an external TTL load. When writing a "1", a large current is supplied through the large pullup device to allow a fast data transfer. After a short time (less than one instruction cycle), the large device is shut off and the small pullup maintains the "1" level indefinitely. However, in this situation, an input device capable of overriding the small amount of sustaining current supplied by the pullup device can be read. (Alternatively, the data written can be read). So, by writing a "1" to any particular pin, that pin can serve either as a true high-level latched output pin, or as just a pullup resistor on an input. This allows maximum user flexibility in selecting his input or latched output pins, with a minimum of external components.

Port 00-07 is also quasi-bidirectional, except there is no large pullup device. As outputs, this port is essentially open drain.

By mask option the small pullup devices on P00-P07 may be deleted on any pin providing a true open drain output. This is useful in driving analog circuits and certain loads, such as keyboards.

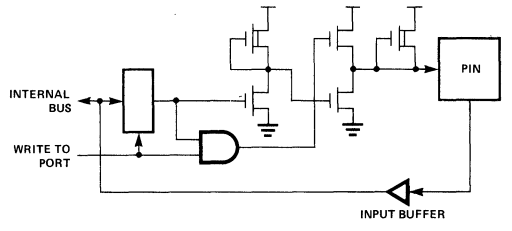


FIGURE 3. QUASI-BIDIRECTIONAL PORT STRUCTURE

2.9.1 T1 Input

The 8021 T1 input line can be used as an input for the following functions:

- Event Counter (external input)
- Test input for branch instructions
- Zero voltage crossing detection

The operation of T1 as an input to the Event Counter is described in the Timer/Event Counter section. When used as a test input, the JT1 and JNT1 instructions test for 1 and 0 levels, respectively.

The T1 pin can also be used to detect the zero crossing of slowly moving AC signals (60 Hz). The self-biasing circuit shown in Figure 4 permits the Test 1 input to detect when the input voltage crosses zero within ±5%; the voltage is then coupled through a 1.0μf capacitor. Maximum input voltage is 3V peak-to-peak. The zero cross detection is especially useful in SCR control of 60 Hz power and in developing time-of-day and other timing routines. As a ROM mask option there is a pullup resistor that is useful for switch contact input or standard TTL.

2.9.2 High Current Outputs

Very high current drive is desirable for minimizing external parts required to do high power control. P10 and P11 have been designated high drive outputs capable of sinking 7mA at V_{SS} + 2.5 volts. (For clarity, this is 7mA to V_{SS} with a 2.5 volt drop across the buffer.) These pins may, of course, be paralleled for 14mA drive if the output logic states are always the same.

2.9.3 Expanded I/O

The 8021 can be used with the 8243 I/O expander chip, which provides additional I/O capability with a limited number of overhead pins. This chip has 4 directly addressable 4-bit ports. It connects to the PROG pin, which provides a clock, and pins P20-P23, which provide address and data. These ports can be written with a MOVD P,A; ANLD P,A; and ORLD P,A for Ports 4-7. A high to low transition on PROG signifies that address and control are available on P20-P23. The previous data on P20-P23 before an output expander instruction is lost. Therefore, when using an output expander P20-P23 are not useful for general input/output. Reading is via the MOVD A,P. This circuit configuration is shown in Figure 5. The timing diagram is shown in Figure 6.

The 8021 can also use standard low cost TTL to expand the number of I/O lines. An example is shown in the Applications section.

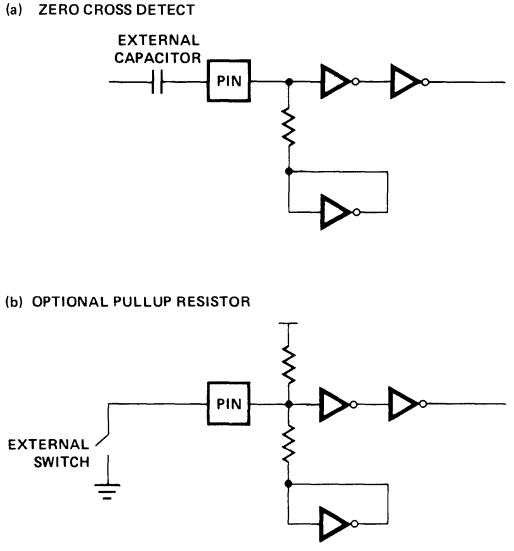


FIGURE 4. TEST 1 PIN

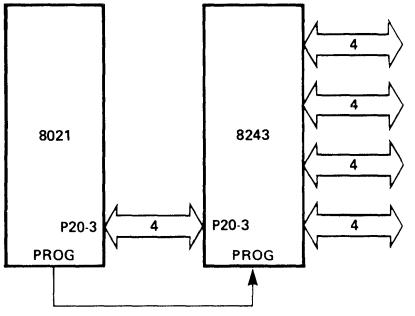


FIGURE 5. I/O EXPANDER INTERFACE

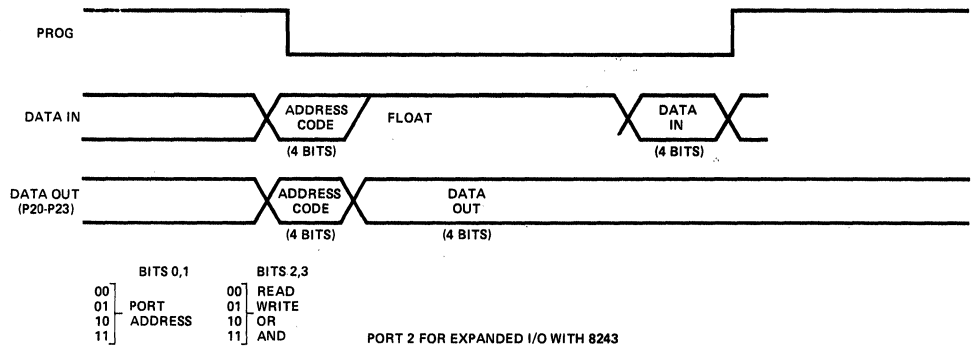


FIGURE 6. EXPANDED I/O TIMING DIAGRAM

2.10 CPU

The 8021 CPU has arithmetic and logical capability. A wide variety of arithmetic and logic instructions may be exercised, which affect the contents of the accumulator, and/or direct or indirect scratchpad locations. Provisions have been made for simplified BCD arithmetic capability using the DAA, SWAP A, and XCHD instructions. In addition, MOVP A,@A allows table lookup for display formating and constants. The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. Use the conditional jump instructions with the tests listed below to effect a change in the program execution sequence.

Test	Jump Condition		Jump Instructions
Accumulator	A=0	A≠0	JZ JNZ
Carry Flag	0	1	, JC
Timer Overflow Flag	—	1	JTF
Test Input-T1	0	1	JNT1, JT1

2.11 Reset

A positive-going signal to the RESET input resets the necessary miscellaneous flip-flops and sets the program counter and stack pointer to zero.

8022 Functional Specifications

The 8022's architecture is based upon the 8021, and many functions of the two parts are identical.

2.12 Program Memory

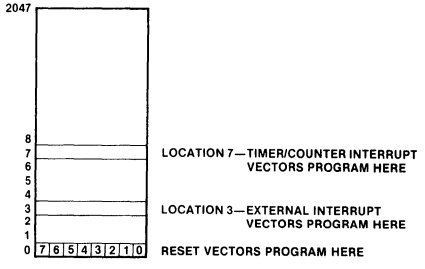
The 8022 program memory consists of 2048 words 8 bits wide which are addressed by the program counter. The memory is ROM which is mask programmable at the factory. No external ROM expansion capability is provided. There are three locations in program memory of special importance.

Location 0: Activating the RESET line of the processor causes the first instruction to be fetched from location 0.

Location 3: Activating the interrupt input line (T0) of the processor (if interrupt is enabled) causes a jump to subroutine.

Location 7: A timer/event counter interrupt resulting from a timer/counter overflow causes a jump to subroutine (if timer/counter interrupt is enabled).

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service routine is stored in location 3, and the first word of a timer/event counter interrupt service routine is stored in location 7.



PROGRAM MEMORY MAP

Program memory can be used to store constants as well as program instructions. The MOVP instruction allows easy table lookup for constants and display formatting.

2.13 Data Memory

On-chip data memory is organized as 64 words eight bits wide. All locations are indirectly addressable and eight designated locations are directly addressable. Also included in the data memory is the program counter stack, addressed by a 3-bit stack pointer.

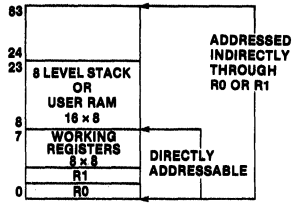
The first eight locations (0–7) of the array are designated as working registers and are directly addressable by any of the 11 direct register instructions. These locations are readily accessible for a variety of operations with a minimum number of instruction bytes required for their manipulation. Thus, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

Registers 0 and 1 have yet another function in that they can be used to indirectly address all locations in the data memory using the indirect register instructions. These two RAM pointer registers are especially useful for repetitive type operations on adjacent memory locations. The indirect register instruction specifies which register is used to address a location in RAM. The contents of the addressed location are used during the execution of the instruction and may be modified. The pointer registers may also point to registers 0–7, if desired.

Locations 8–23 serve a dual role in that they contain the 8-level program counter stack, two RAM locations per level. The program counter stack enables the processor to keep track of the return addresses generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the program counter stack's eight register pairs will be loaded with the next return address generated. The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. The first subroutine CALL or interrupt results in the program counter contents being transferred to locations 8 and 9. The stack pointer is then incremented by one and points to locations 10 and 11 in anticipation of another CALL. The end of a subroutine, which is signaled by a return instruction (RET or RETI), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Unlike the 8048, in the 8022 the unincremented program counter address is stored in the address stack. The stack contents are then incremented before being loaded into the program counter during a return (RET) from subroutine. However, during a return (RETI) from interrupt, the stack contents are loaded directly into the program counter. This difference makes it imperative to use only RETI's to return from interrupts, and RET's to return from subroutines.

Since the program counter's addresses are 11 bits long, two bytes or registers must be used to store a single address. Thus, the 16-byte program counter stack permits up to a total of 8 levels of subroutine nesting without overflowing the stack. If overflow does occur, the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111. If a particular application does not require 8 levels of nesting, the unused portion of the progam counter stack may be used as any other indirectly addressable RAM location. For example, if only 3 levels of subroutine nesting are used, then only locations 8–13 need be reserved for the program counter stack, and locations 14–23 can be used for data storage.



DATA MEMORY MAP

2.14 Oscillator and Clock

The 8022 contains its own on-board oscillator and clock circuit, requiring only an external timing control element. This control element can be a crystal, inductor, or clock. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. Therefore, to obtain a 10 μs instruction cycle, a 3 MHz crystal should be used. The minimum instruction cycle time of 8.4 μsec corresponds to a 3.58 mHz crystal.

2.15 Timer/Event Counter

Like the other MCS-48 microcomputers, the 8022 has an internal timer/event counter. This circuit can monitor elapsed time or count external events that occur during program execution.

See the 8021 description, Section 2.8, for a complete explanation.

The 8022 has 26 lines which can be used for digital input or output functions. These lines are organized as 3 ports of 8 lines, each of which serve as either inputs, outputs, or bidirectional ports, and 2 test inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2 have identical operating characteristics and are both quasi-bidirectional. That is, each line may serve as an input, an output, or both. Data written to these ports is statically latched and remains unchanged until rewritten. As inputs, these lines are non-latching; i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and all outputs will drive at least one standard TTL load. See Section 2.1.4 for a more complete description of the quasi-bidirectional structure.

2.16.1 Port 0 Comparator Inputs

Port 0 has been modified from the standard quasi-bidirectional structure to allow an optional open drain configuration with comparator inputs. The low impedance pullup device has been eliminated and the high impedance pullup is optional. Thus, the user can choose via a mask programmable selection each line of port 0 to be either quasi-bidirectional with a high impedance or true open-drain. The open drain configuration allows the line to sink current through the low impedance pull-down device or to float in the high output state. More importantly, the open drain configuration makes port 0 very easy to drive when it is used as inputs. The input circuitry for each line of port 0 includes a voltage comparator which amplifies the voltage difference between the input port line and the port 0 threshold reference pin (V_{TH}). The voltage gain of the comparator is sufficient to sense a 100 mV input differential within the range V_{SS} to $V_{CC}/2$.

If V_{TH} is allowed to float, it will bias itself to the digital switch point of the other ports, and port 0 behaves as a set of normal digital inputs. However, by biasing V_{TH}, the switch point can be both tightly controlled and adjusted. Common uses for

this would include high noise margin inputs ($V_{CC}/2$), unusual logic level inputs as from a diode isolated keyboard, analog channel expansion, and direct capacitive touchpanel interface. The comparator action is automatic and the port is read just as any other port.

2.16.2 High Current Outputs

Very high current drive is desirable for minimizing external parts required to do high power control. P10 and P11 have been designated high drive outputs capable of sinking 7mA at V_{SS} + 2.5 volts. (For clarity, this is 7mA to V_{SS} with a 2.5 volt drop across the buffer.) These pins may, of course, be paralleled for 14mA drive if the output logic states are always the same.

2.16.3 Expanded I/O

In addition to the 26 digital I/O lines contained onboard the 8022, a user can obtain additional I/O lines by utilizing the Intel® 8243 I/O expander chip or standard TTL. The 8243 interfaces to 4 port lines of the 8022 (lower half of port 2) and is strobed by the PROG line of the 8022.

The interface procedure is exactly the same as with the 8021—see Section 2.9.3.

2.17 Test and Interrupt Inputs

In addition to the 24 general purpose I/O lines which comprise ports 0, 1, and 2, the 8022 has two inputs which are testable via conditional jump instructions, T0 and T1. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. T0 and T1 have other functions as well.

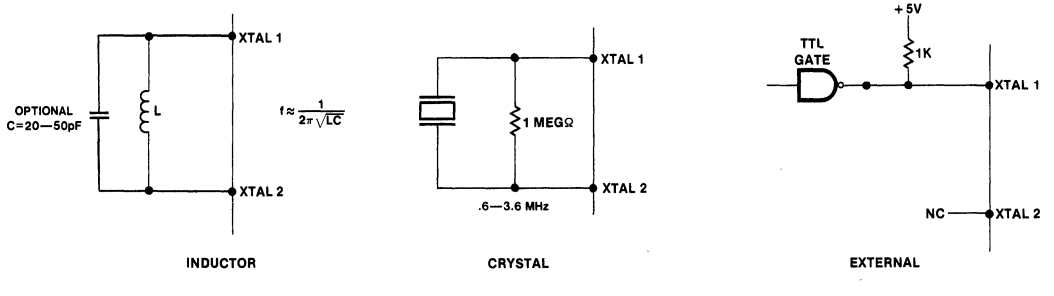
The Test 0 pin serves as an external interrupt input as well as a testable input. An interrupt sequence is initiated by applying a low "0" level input to the T0 pin when external interrupt is enabled. Interrupt is level triggered and active low to allow "WIRE ORING" of several interrupt sources at the input pin. When an interrupt is detected, it causes a "jump to subroutine" at location 3 in program memory as soon as all other cycles of the current instruction are complete. At this time, the program counter contents are saved in the program counter stack, but the remaining status of the processor is not. Unlike the 8048, the 8022 does not contain a program status word. Thus, when appropriate, the carry and auxiliary carry flags are saved in software, as is the accumulator. The routine shown below saves the accumulator and the carry flags in only four bytes.

Instructions	Bytes	Comments
MOV R6,A	1	;save accumulator
CLR A	1	;clear accumulator
DA A	1	;convert carry flags into sixes
MOV R7,A	1	;save status of carry flags

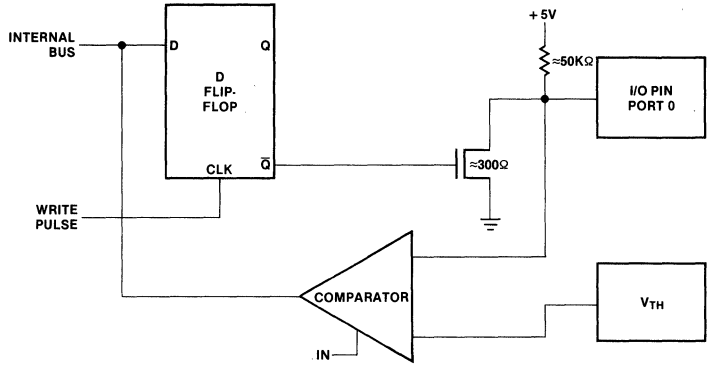
The end of an interrupt service subroutine is marked by the execution of a Return from Interrupt instruction (RETI). Prior to returning from the interrupt subroutine, however, the status of the accumulator and the carry flags are restored in software. The following routine restores the status of the accumulator and the carry flags, which was previously saved, in five bytes.

Instructions	Bytes	Comments
MOV A,R7	1	;restore carry flags status to
Add A,#0AAH	2	;accumulator and set/clear carry flags
MOV A,R6	1	;restore accumulator
RETI	1	;return

The interrupt system is single level in that once an interrupt is detected, all further interrupt requests are ignored until execution of a RETI re-enables

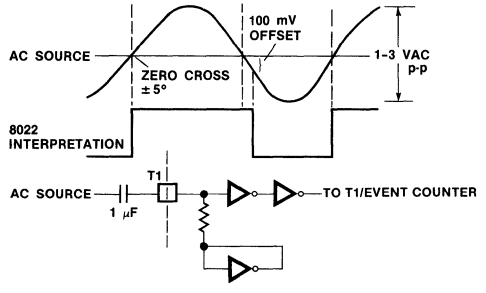


FREQUENCY REFERENCE OPTIONS



PORT 0 I/O STRUCTURE

the interrupt input logic. This sequence holds true also for an internal interrupt generated by timer overflow. If an external interrupt and an internal timer/counter generated interrupt are detected at the same time, the external source will be recognized. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the counter (one less than terminal count) and enabling the event counter mode. A low-to-high transition on the T1 input will then cause an interrupt vector to location 7.



ZERO-CROSS DETECTION

The Test 1 pin, in addition to being a testable input, serves two other important functions. It can be used as an input pin to the external event counter, as previously mentioned, and it can be used to detect the zero crossing point of slow moving AC signals. Execution of the STRT CNT instruction puts the T1 pin in the counter input mode by connecting T1 to the counter and enabling the counter. Subsequent low-to-high transitions on T1 will cause the counter to increment. Note that this operation differs from the rest of the MCS-48 devices, which increment the counter on high-to-low transitions. This change was made on the 8022 to take advantage of the accuracy of the rising edge detection on the zero cross circuitry. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 30 μs when using a 3 MHz crystal)—there is no minimum frequency.

In addition to serving as a testable input and as the counter input, the T1 pin has special circuitry to detect when an AC signal crosses its average DC level. When driven directly, this pin responds as a normal digital input. To utilize the zero cross detection mode, an AC signal of approximately 1–3 VAC p-p magnitude and a maximum frequency of 1 kHz is coupled through an external capacitor (1 μF) to the T1 pin.

The internal digital state is sensed as a zero until the rising edge crosses the DC average level, when it becomes a one. This is accomplished by the self-biasing high gain amplifier which is included in the T1 input. This circuit biases the T1 input exactly at its switching point, such that a small change will cause a digital transition to occur. This digital transition takes place within 5 degrees of the zero point. The digital value of T1 remains a one until the falling edge of the AC input drops approximately 100 mV below the switching point of the rising edge (100 mV below the zero point, if the digital transition occurred exactly at the zero point). The 100 mV offset is created by hysteresis and eliminates chattering of the internal signal caused by the external noise.

The zero cross detection capability allows the user to make the 60 Hz power signal the basis for this system timing. All timing routines, including time-of-day, can be implemented using the zero cross detection capability of T1 and its conditional jump instructions. In addition, the zero cross detection feature can be used in conjunction with the timer interrupt to interrupt processing at the zero voltage point. This enables the user to control voltage phase sensitive devices such as triacs and SCRs, and to use the 8022 in applications such as shaft speed and angle measurement.

2.18 Analog to Digital Converter

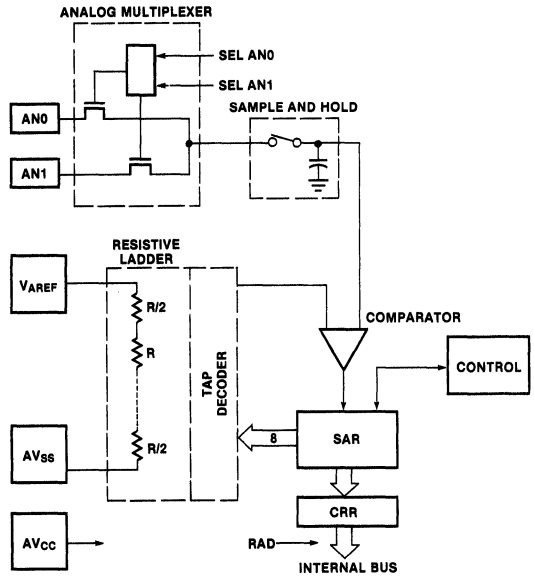
The 8022 contains on-chip a complete hardware implementation of an 8-bit analog to digital (A/D) converter with two multiplexed analog inputs. The A/D converter utilizes a successive approximation technique to provide an updated conversion once every four instruction cycles with a minimum of required software.

The A/D converter consists of four main parts, the input circuitry, a series string of resistors, a voltage comparator, and the successive approximation logic. The two analog inputs are multiplexed on-chip and selected via software by the SEL AN0 and SEL AN1 instructions. Besides selecting one of the analog inputs, these instructions restart the conversion sequence which operates continuously. Restarting a conversion sequence deletes the conversion in progress but does not affect the result of the previous conversion which is stored in the conversion result register. The continuous operation of the A/D converter saves program space and time by allowing the user obtain multiple readings from a given input with only one select instruction. To obtain a valid conversion reading, the user must provide the analog input signal no later than the beginning of the select instruction cycle. The analog input is then sampled by the A/D converter and maintained internally. This voltage becomes one input to the voltage comparator which amplifies the difference between the analog input and the voltage tap on the series resistor string.

The series resistor string is connected between the A/D reference pin (V_{AREF}) and ground (AV_{SS}). It is comprised of 256 identical resistors which divide the voltage between these two pins into 256 identical voltage steps. This configuration gives the converter its inherent monotonicity. The range of V_{AREF} in which full 8-bit resolution can be provided is between $V_{CC}/2$ and V_{CC}.

Thus, the user is given a minimum voltage range from ground to $V_{CC}/2$ and a maximum range from ground to V_{CC} over which 8-bit resolution is insured.

The voltage tap on the series resistor string is selected by the resistor ladder decoder. This decoder is driven by the 8-bit successive approximation register (SAR). Each bit of the SAR is set in succession, MSB to LSB, and a voltage comparison between the selected resistor ladder voltage and the analog input voltage is performed after the setting of each bit. The result of each comparison determines whether the particular bit will remain set or be reset. All comparisons are performed automatically by the on-chip A/D hardware. At the end of 8 comparisons the SAR contains a valid digital result which is then latched into the conversion result register (CRR). The RAD instruction (read A/D) loads the conversion result from the CRR to the accumulator of the 8022.



A/D CONVERTER BLOCK DIAGRAM

As mentioned previously, the software and time required to perform an A/D conversion is optimized by the 8022's on-chip A/D converter configuration. Typical software for reading two sequential A/D conversions and storing them in data memory is shown below:

```
First          SEL AN0       ;Starts conversion of AN0 input
Conversion     MOV R0,#24    ;Set up memory pointer
 50 μs         RAD           ;First conversion value to
 4 bytes                      accumulator

Second         MOV @R0,A     ;Store first conversion value
Conversion     INC R0        ;Increment memory location
 40 μs         RAD           ;Second conversion value to
 3 bytes                      accumulator
```

Note that the second conversion occurs without a second select instruction being used. Rather, the continuous operation of the A/D converter provides an updated digital value 4 instruction cycles after the first.

To insure maximum accuracy from the A/D converter, separate power supply pins (AV_{CC} and AV_{SS}) and a substrate pin (SUBST) have been provided. Supplying the power supply pins with a well filtered and regulated voltage supply minimizes the effect of power supply variance and system noise. The substrate pin should be bypassed to ground through a 500 pF to 0.001 μF capacitor.

2.19 CPU

The 8022 CPU has arithmetic and logical capability. There is a wide variety of arithmetic and logic instructions which affect the contents of the accumulator, and/or direct or indirect scratchpad locations. Provisions have been made for simplified BCD arithmetic capability using the DAA, SWAP A, and XCHD instructions. In addition, MOVP A,@A allows table lookup for display formating and constants. The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the user's program. Use the conditional jump instructions with the tests listed below to effect a change in the program execution sequence.

Test	Jump Condition		Jump Instructions
Accumulator	A=0	A≠0	JZ JNZ
Carry Flag	0	1	JNC, JC
Timer Overflow Flag	—	1	JTF
Test Input-T1	0	1	JNT1, JT1
Test Input-T0	0	1	JNT0, JT0

2.20 8022 Testing and Debug

To facilitate testing and debug, certain test modes may be activated in the 8022 by raising combinations of RESET, TEST 1 and PROG to 15 volts. Internal ROM is dumped out sequentially for verification. External memory operation is used for CPU checkout.

Reset	Prog	Test 1	Case	Function
5V	X	X		Power On Clear
0V	X	X		Normal Operation
15V	15V	15V	Mode 1a	On each cycle internal ROM is dumped to Port 0—sequentially after ALE leading edge.
15V	15V	⎍	Mode 1b	On every TEST 1 falling edge the program counter increments, dumps internal ROM to Port 0
0V	15V	X	Mode 2	Chip will operate from external memory (one page) via Port 0. ALE strobes Address out, memory in.
15V	X	X	Mode 3	Chip accepts op codes into Port 1. Allows Port 0 and 8243 testing.

NOTE
X = Normal mode - between 0V and V_{CC}

The Expanded MCS-48™ System

THE EXPANDED MCS-48™ SYSTEM

3.0 Summary

If the capabilities resident on the single-chip 8048/8049, 8748, or 8035/8039 are not sufficient for your system requirements, special on-board circuitry allows the addition of a wide variety external memory, I/O, or special peripherals you may require. The processors can be directly and simply expanded in the following areas:

- Program Memory to 4K words
- Data Memory to 320 words (384 words with 8049)
- I/O by unlimited amount
- Special Functions using 8080/8085 peripherals

By using bank switching techniques maximum capability is essentially unlimited. Bank switching is discussed later in the chapter. Expansion is accomplished in two ways:

1. Expander I/O—A special I/O Expander circuit the 8243 provides for the addition of four 4-bit Input/Output ports with the sacrifice of only the lower half (4 bits) of port 2 for inter-device communication. Multiple 8243's may be added to this 4-bit bus by generating the required "chip select" lines.

2. Standard 8085 Bus—One port of the 8048 is like the 8 bit bidirectional data bus of the 8085 microcomputer system allowing interface to the numerous standard memories and peripherals of the MCS-80/85 microcomputer family.

MCS-48 systems can be configured using either or both of these expansion features to optimize system capabilities to the application. Both expander devices and standard memories and peripherals can be added in virtually any number and combination required.

3.1 Expansion of Program Memory

Program Memory is expanded beyond the resident 1K or 2K words by using the 8085 BUS feature of the MCS-48. All program memory fetches from addresses less than 1024 (2048) occur internally with no external signals being generated (except ALE which is always present). At address 1024 the 8048 automatically initiates external program memory fetches.

3.1.1 Instruction Fetch Cycle (External)

For all instruction fetches from addresses of 1024 (2048) or greater the following will occur:

1. The contents of the 12 bit program counter will be output on BUS and the lower half of port 2.

2. Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.

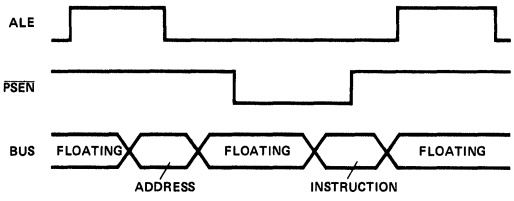
3. Program Store Enable ($\overline{\text{PSEN}}$) indicates that an external instruction fetch is in progress and serves to enable the external memory device.

4. BUS reverts to input (floating) mode and the processor accepts its 8 bit contents as an instruction word.

All instruction fetches including internal addresses can be forced to be external by activating the EA pin of the 8048/8049. The 8035/8039 processors without program memory always operate in the external program memory mode (EA=5V).

3.1.2 Extended Program Memory Addressing (Beyond 2K)

For programs of 2K words or less, the 8048/8049 addresses program memory in



INSTRUCTION FETCH FROM EXTERNAL PROGRAM MEMORY

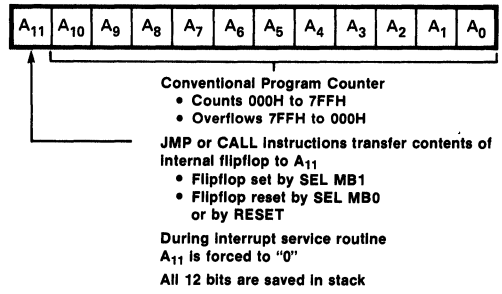
the conventional manner. Addresses beyond 2047 can be reached by executing a program memory bank switch instruction (SEL MB0, SEL MB1) followed by a branch instruction (JMP or CALL). The bank switch feature extends the range of branch instructions beyond their normal 2K range and at the same time prevents the user from inadvertently crossing the 2K boundary.

Program Memory Bank Switch

The switching of 2K program memory banks is accomplished by directly setting or resetting the most significant bit of the program counter (bit 11). Bit 11 is not altered by normal incrementing of the program counter but is loaded with the contents of a special flip-flop each time a JMP or CALL instruction is executed. This special flip-flop is set by executing an SEL MB1 instruction and reset by SEL MB0. Therefore, the SEL MB instruction may be executed at any time prior to the actual bank switch which occurs during the next branch instruction encountered. Since all twelve bits of the program counter including bit (11) are stored in the stack when a Call is executed, the user may jump to subroutines across the 2K boundary and the proper bank will be restored upon return. However, the bank switch flipflop will not be altered on return.

Interrupt Routines

Interrupts always vector the program counter to location 3 or 7 in the **first** 2K bank and bit 11 of the program counter is held at "0" during the interrupt service routine. The end of the service routine is signalled by the execution of an RETR instruction. Interrupt service routines should therefore be contained



PROGRAM COUNTER

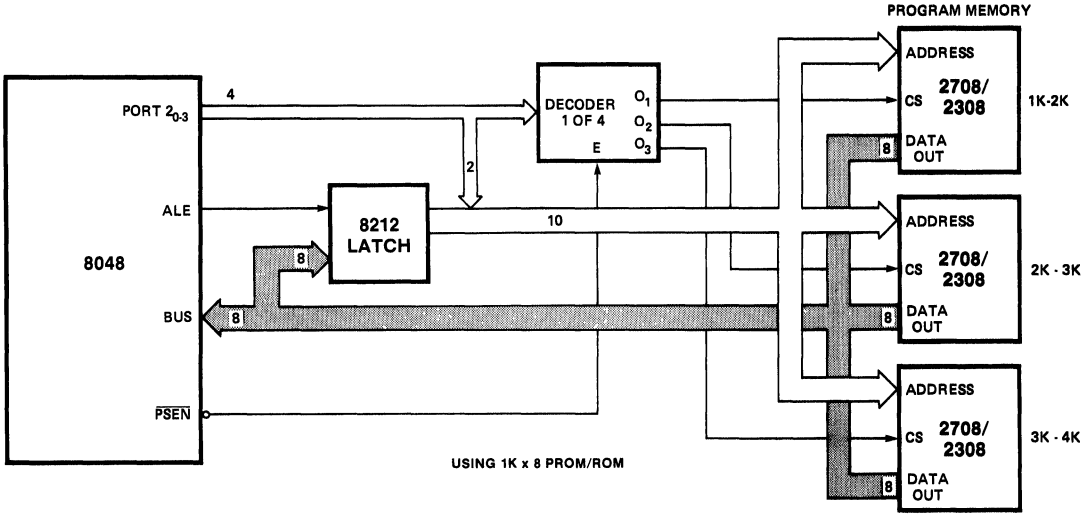
entirely in the lower 2K words of program memory. The execution of a SEL MB0 or SEL MB1 instruction within an interrupt routine is not recommended since it will not alter PC11 while in the routine, but will change the internal flip flop.

3.1.3 Restoring I/O Port Information

Although the lower half of Port 2 is used to output the four most significant bits of address during an external program memory fetch, the I/O information is still outputed during certain portions of each machine cycle. I/O information is always present on Port 2 lower at the rising edge of ALE and can be sampled or latched at this time.

3.1.4 Expansion Examples

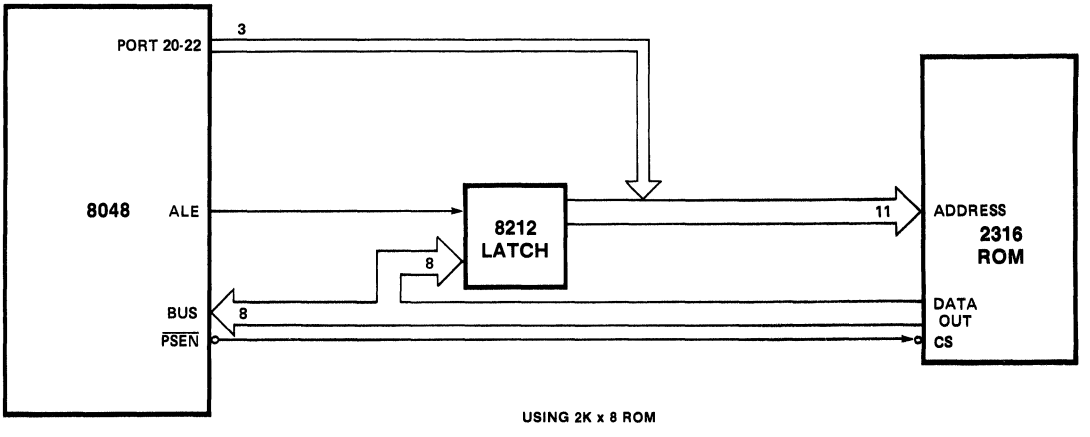
The accompanying figure shows the addition of three 2708 1K X 8 EPROMs or three 2308 pin-compatible ROM replacements for a total of 4K words of program memory. The BUS port of the 8048 is connected directly to the data output lines of the memories. The lower 8 bits of address are latched in an 8212 8-bit latch using ALE as the strobe. The lower half of Port 2 provides the upper 4 bits of address and since these address bits are stable for the duration of the program memory fetch, they do not have to be latched. Two of the upper address bits are connected directly to the address inputs of the memories while the two most significant bits are decoded to provide the three chip selects needed. The $\overline{PSEN}$ output of the 8048/8748 is used to enable the chip select lines and therefore the memories.



EXPANDING MCS-48™ PROGRAM MEMORY USING STANDARD MEMORY PRODUCTS

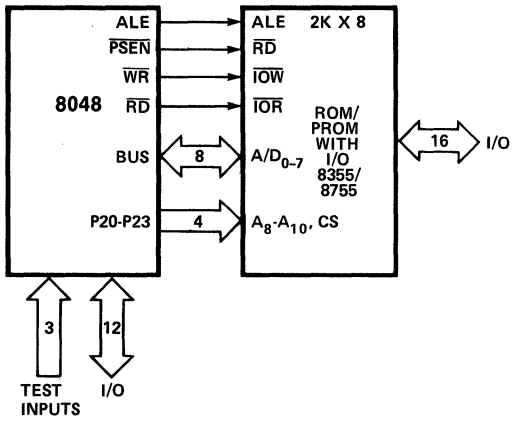
Also shown is the addition of 2K words of program memory using an 2316A 2K x 8 ROM to give a total of 3K words of program memory. In this case no chip select decoding is required and $\overline{PSEN}$ enables the memory directly through the chip select input. If the system requires only 2K of program the same configuration can be used with an 8035 substituted for the 8048. The 8049 would provide 4K with the same configuration.

The next figure shows how the new 8755/8355 EPROM/ROM with I/O interfaces directly to the 8048 without the need for an address latch. The 8755/8355 contains an internal 8-bit address latch eliminating the need for an 8212 latch. In addition to a 2K X 8 program memory the 8755/8355 also contains 16 I/O lines addressable as two 8-bit ports. These ports are addressed as external RAM; there-



EXPANDING MCS-48™ PROGRAM MEMORY USING STANDARD MEMORY PRODUCTS

fore, the $\overline{RD}$ and $\overline{WR}$ outputs of the 8048 are required. See the following section on data memory expansion for more detail. The subsequent section on I/O expansion explains the operation of the 16 I/O lines.



EXTERNAL PROGRAM MEMORY INTERFACE

3.2 Expansion of Data Memory

Data Memory is expanded beyond the resi dent 64 words by using the 8085 type bus feature of the MCS-48.

3.2.1 Read/Write Cycle

All address and data is transferred over the 8 lines of BUS. A read or write cycle occurs as follows:

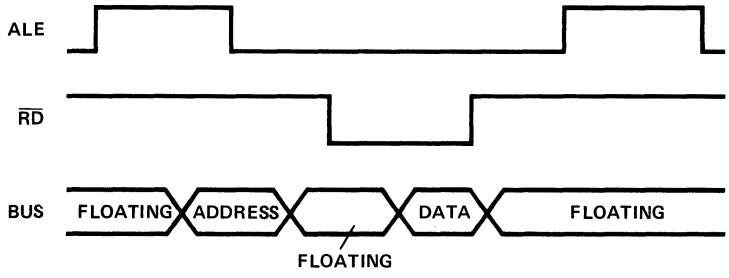
1. The contents of register R0 or R1 is outputed on BUS.

2. Address Latch Enable (ALE) indicates address is valid. The trailing edge of ALE is used to latch the address externally.

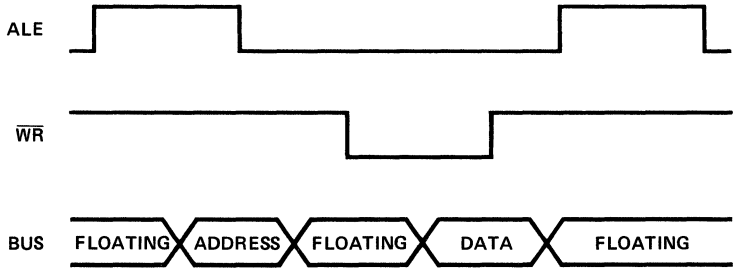
3. A read ($\overline{RD}$) or write ($\overline{WR}$) pulse on the corresponding output pins of the 8048 indicates the type of data memory access in progress. Output data is valid at the trailing edge of $\overline{WR}$ and input data must be valid at the trailing edge of $\overline{RD}$.

4. Data (8-bits) is transferred in or out over BUS.

READ FROM EXTERNAL DATA MEMORY



WRITE TO EXTERNAL DATA MEMORY



3.2.2 Addressing External Data Memory

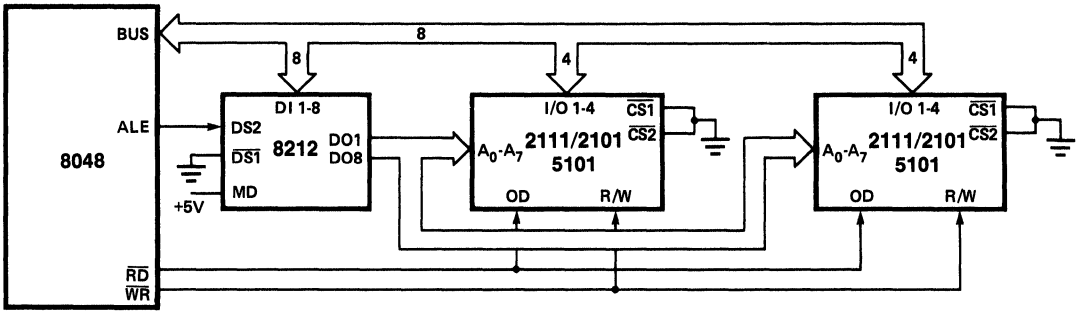
External Data Memory is accessed with its own two-cycle move instructions MOVX A, @R and MOVX @R, A which transfer 8 bits of data between the accumulator and the external memory location addressed by the contents of one of the RAM Pointer Registers R0 or R1. This allows 256 locations to be addressed in addition to the resident locations. Additional pages may be added by "bank switching" with extra output lines of the 8048.

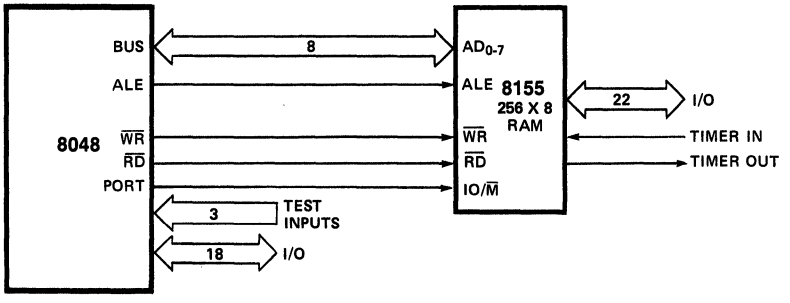
3.2.3 Examples of Data Memory Expansion

The accompanying figure shows how the 8048 can be expanded using standard 256 X 4 static RAMs such as the 2101-2 or its low power CMOS equivalent, the 5101. An 8212 serves as an address latch while each 4-bit half of BUS is connected directly to a bidirectional 4-bit data bus of the memories. The $\overline{WR}$ output of the processor controls the Read/Write input of the memories while the data bus output drivers of the memories are controlled by $\overline{RD}$. The chip select lines of the memories are continuously enabled unless additional pages of RAM are required. Also shown is the expansion of data memory using the 8155 memory and I/O expanding device. Since the 8155 has an internal 8-bit address latch it can interface directly to the 8048 without the use of an external 8212 latch. The 8155 provides an additional 256 words of static data memory and also includes 22 I/O lines and a 14 bit timer. See the following section on I/O expansion and the 8155 data sheet for more details on these additional features.

3.3 Expansion of Input/Output

There are four possible modes of I/O expansion with the 8048: one using a special low cost expander, the 8243;





8048 INTERFACE TO 256 X 8 STANDARD MEMORIES

another using standard MCS-80/85 I/O devices; and a third using the combination memory/I/O expander devices the 8155, 8355, and 8755. It is also possible to expand using standard TTL devices as shown in Chapter 5.

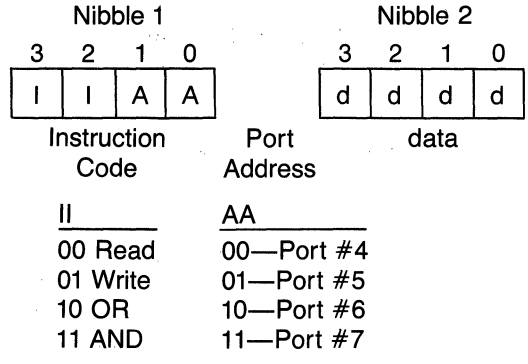
3.3.1 I/O Expander Device

The most efficient means of I/O expansion for small systems is the 8243 I/O Expander Device which requires only 4 port lines (lower half of Port 2) for communication with the 8048. The 8243 contains four 4-bit I/O ports which serve as extension of the on chip I/O and are addressed as ports #4-7. The following operations may be performed on these ports:

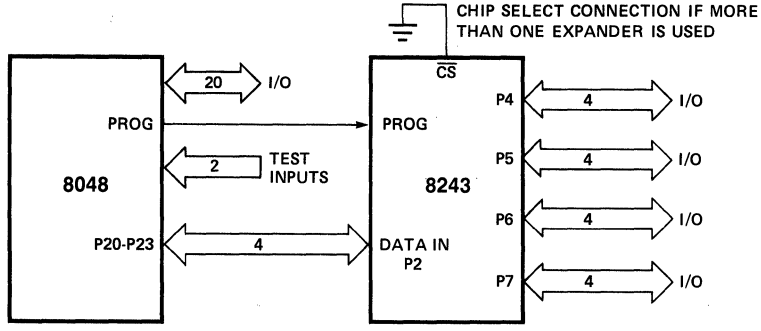
1. Transfer Accumulator to Port.
2. Transfer Port to Accumulator.
3. AND Accumulator to Port.
4. OR Accumulator to Port.

A 4-bit transfer from a port to the lower half of the Accumulator sets the most significant four bits to zero. All communication between the 8048 and the 8243 occurs over Port 2 lower (P20-P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

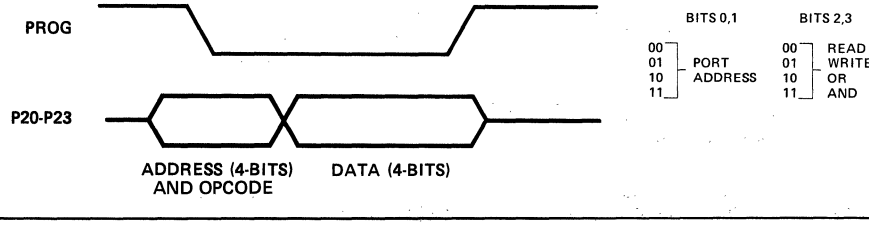
The first containing the "op code" and port address and the second containing the actual 4 bits of data.



EXPANDER INTERFACE



OUTPUT EXPANDER TIMING



A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the four bit bus and chip selected using additional output lines from the 8048/8748.

I/O Port Characteristics

Each of the four 4-bit ports of the 8243 can serve as either input or output and can provide high drive capability in both the high and low state.

3.3.2 I/O Expansion with Standard Peripherals

Standard MCS-80/85 type I/O devices may be added to the MCS-48 using the same bus and timing used for Data Memory expansion. I/O devices reside on the Data Memory bus and in the data memory address space and are accessed with the same MOVX instructions. See the previous section on data memory expansion for a description of timing. The following are a few of the Standard MCS-80 devices which are very useful in MCS-48 systems.

8214 Priority Interrupt Encoder
8251 Serial Communications Interface
8255 General Purpose Programmable I/O
8279 Keyboard/Display Interface
8253 Interval Timer

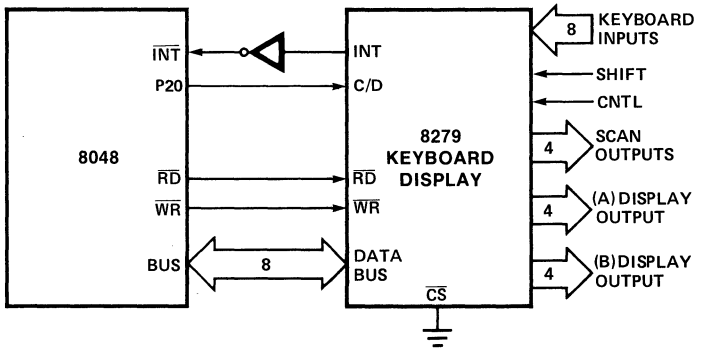
See Chapter 7 for detailed data sheets on these and other components.

3.3.3 Combination Memory and I/O Expanders

As mentioned in the sections on program and data memory expansion the 8355/8755 and 8155 expanders also contain I/O capability.

8355/8755: These two parts are ROM and EPROM equivalents and therefore contain the same I/O structure. I/O consists of two 8-bit ports which normally reside in the external data memory address space and are accessed with MOVX instructions. Associated with each port is an 8-bit Data Direction Register which defines each bit in the port as either an input or an output. The data direction registers are directly addressable thereby allowing the user to define under software control each individual bit of the ports as either input or output. All outputs are statically latched and double buffered. Inputs are not latched.

8155/8156: I/O on the 8155/8156 is configured as two 8-bit programmable I/O ports and one 6-bit programmable port. These three registers and a Control/Status register are accessible as external data memory with the MOVX instructions. The contents of the control register determines the mode of the three ports. The ports can be programmed as input or output with or without associated handshake communication lines. In the handshake mode, lines of the six-bit port become input and output strobes for the two 8-bit ports. See the

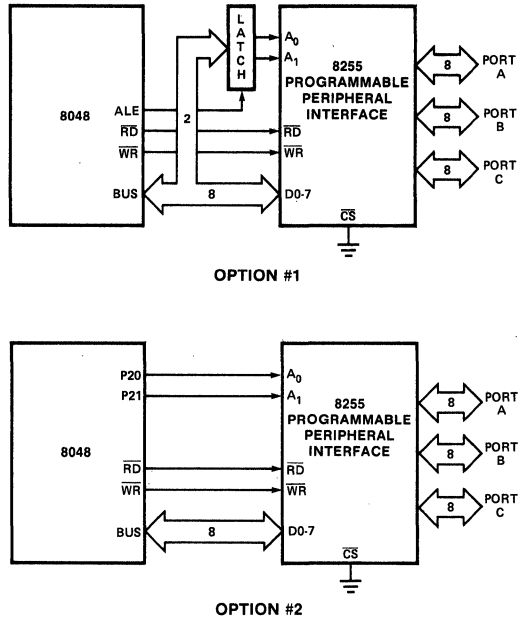


data sheet in Chapter 6 for details. Also included in the 8155 is a 14-bit programmable timer. The clock input to the timer and the timer overflow output are available on external pins. The timer can be programmed to stop on terminal count or to continuously reload itself. A square wave or pulse output on terminal count can also be specified.

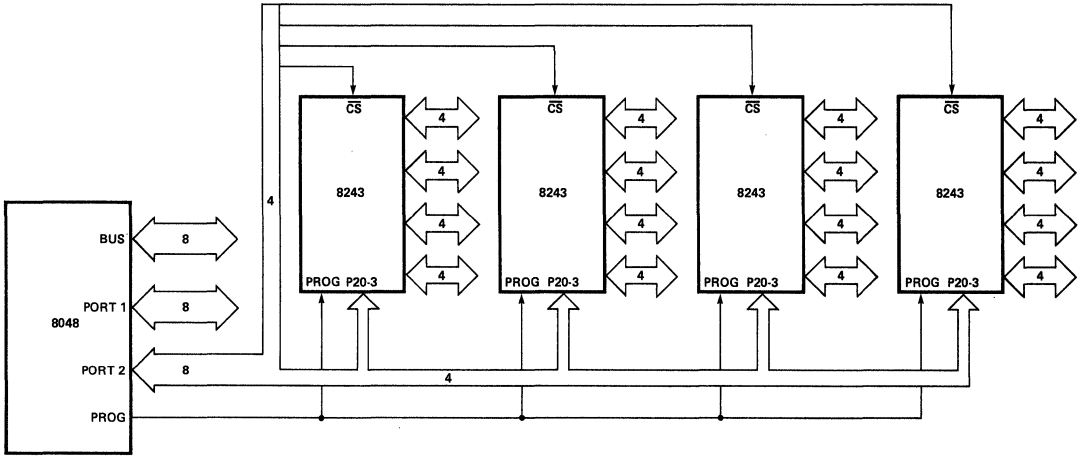
I/O Expansion Examples (See Also Chapter 5)

The accompanying figure shows the expansion of I/O using multiple 8243's. The only difference from a single 8243 system is the addition of chip selects provided by additional 8048 output lines. Two output lines and a decoder could also be used to address the four chips. Large numbers of 8243's would require a chip select decoder chip such as the 8205 to save I/O pins.

Also shown is the 8048 interface to a standard MCS-80 peripheral; in this case, the 8255 Programmable Peripheral Interface, a 40 pin part which provides three 8-bit programmable I/O ports. The 8255 bus interface is typical of programmable MCS-80 peripherals with an 8-bit bidirectional data bus, a $\overline{RD}$ and $\overline{WR}$ input for Read/Write control, a $\overline{CS}$ (chip select) input used to enable the Read/Write control logic and the address inputs used to select various internal registers.



INTERFACE TO MCS-80 PERIPHERALS



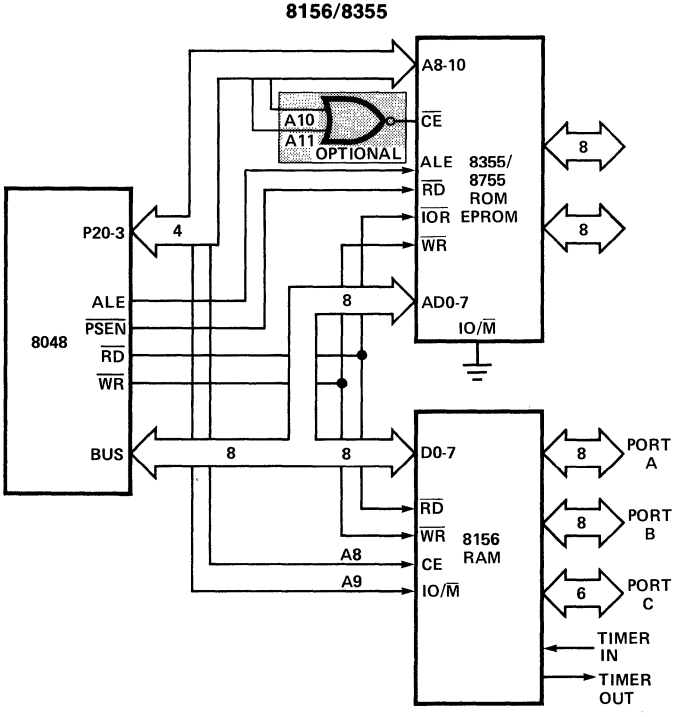
LOW COST I/O EXPANSION

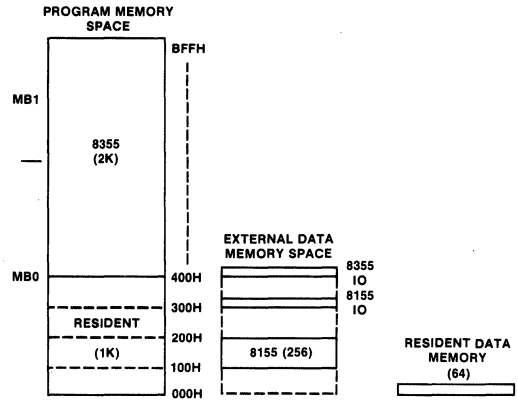
Interconnection to the 8048 is very straightforward with BUS, $\overline{RD}$, and $\overline{WR}$ connecting directly to the corresponding pins on the 8255. The only design consideration is the way in which the internal registers of the 8255 are to be addressed. If the registers are to be addressed as external data memory using the MOVX instructions, the appropriate number of address bits (in this case, 2) must be latched on BUS using ALE as described in the section on external data memories. If only a single device is connected to BUS, the 8255 may be continuously selected by grounding $\overline{CS}$. If multiple 8255's are used, additional address bits can be latched and used as chip selects.

A second addressing method eliminates external latches and chip select decoders by using output port lines as address and chip select lines directly. This method, of course, requires the setting of an output port with address information prior to executing a MOVX instruction.

3.4 Multi-Chip MCS-48 Systems

The accompanying figure shows the addition of two memory expanders to the 8048, one 8355/8755 ROM and one 8156 RAM. The main consideration in designing such a system is the addressing of the various memories and I/O ports. Note that in this configuration address lines A_{10} and A_{11} have been ORed to chip select the 8355. This ensures that the chip is active for all external program memory fetches in the 1K to 3K range and is disabled for all other addresses. This gating has been added to allow the I/O port of the 8355 to be used. If the chip was left selected all the time there would be conflict between these ports and the RAM and I/O of the 8156. The NOR gate could be eliminated and A_{11} connected directly to the CE (instead of $\overline{CE}$) input of the 8355; however, this would create a 1K word "hole" in the program memory by causing the 8355 to be active in the 2K to 4K range instead of the normal 1K to 3K range.





SECTION	ADDRESS	DESIGNATION
PROG. MEM	000—BFF	
DATA MEM	100—IFF	
8155 PORTS	300	CMD/STATUS
	301	PORT A
	302	PORT B
	303	PORT C
	304	TIMER LOW
	305	TIMER HI
8355 PORTS	400	PORT A
	401	PORT B
	402	DDR A
	403	DDR B

In this system the various locations are addressed as follows:

Data RAM—Addresses 0 to 255 when Port 2 Bit 0 has been previously set = 1 and Bit 1 set = 0

RAM I/O—Addresses 0 to 3 when Port 2 Bit 0 = 1 and Bit 1 = 1

ROM I/O—Addresses 0 to 3 when Port 2 Bit 2 or Bit 3 = 1

See the above memory map.

3.5 Memory Bank Switching

Certain systems may require more than the 4K words of program memory which are directly addressable by the program counter or more than the 256 data memory and I/O locations directly addressable by the pointer registers R0 and R1. These systems can be achieved using "bank switching" techniques. Bank switching is merely the selection of various blocks or "banks" of memory using dedicated output port lines from the processor. In the case of the 8048 program memory is selected in blocks of 4K words at a time while data memory and I/O are enabled 256 words at a time.

The most important consideration in implementing two or more banks is the software required to cross the bank boundaries. Each crossing of the boundary requires that the processor first write a control bit to an output port before accessing memory or I/O in the new bank. If program memory is being switched, programs should be organized to

keep boundary crossings to a minimum. Jumping to subroutines across the boundary should be avoided when possible since the programmer must keep track of which bank to return to after completion of the subroutine. If these subroutines are to be nested and accessed from either bank, a software "stack" should be implemented to save the bank switch bit just as if it were another bit of the program counter.

From a hardware standpoint bank switching is very straight-forward and involves only the connection of an I/O line or lines as bank enable signals. These enables are ANDed with normal memory and I/O chip select signals to activate the proper bank.

3.6 Control Signal Summary

The following table summarizes the instructions which activate the various control outputs of the MCS-48 processors.

CONTROL SIGNAL	WHEN ACTIVE
$\overline{RD}$	DURING MOVX A,@R OR INS BUS
$\overline{WR}$	DURING MOVX @R,A OR OUTL BUS
ALE	EVERY MACHINE CYCLE
$\overline{PSEN}$	DURING FETCH OF EXTERNAL PROGRAM MEMORY (INSTRUCTION OR IMMEDIATE DATA)
PROG	DURING MOVD A,P ANLD P,A MOVD P,A ORLD P,A

During all other instructions these outputs are driven to the inactive state.

3.7 Port Characteristics

BUS Port Operations

The BUS port can operate in three different modes: as a latched I/O port, as a bi-directional bus port, or as a program memory address output when external memory is used. The BUS port lines are either active high, active low, or high impedance (floating).

The latched mode (INS, OUTL) is intended for use in the single chip configuration where BUS is not being used as an expander port. OUTL and MOVX instructions can be mixed if necessary. However, a previously latched output will be destroyed by executing a MOVX instruction and BUS will be left in the high impedance state. INS does not put the BUS in a high impedance state. Therefore, the use of MOVX after OUTL to put the BUS in a high impedance state is necessary before an INS instruction intended to read an external word (as opposed to the previously latched value).

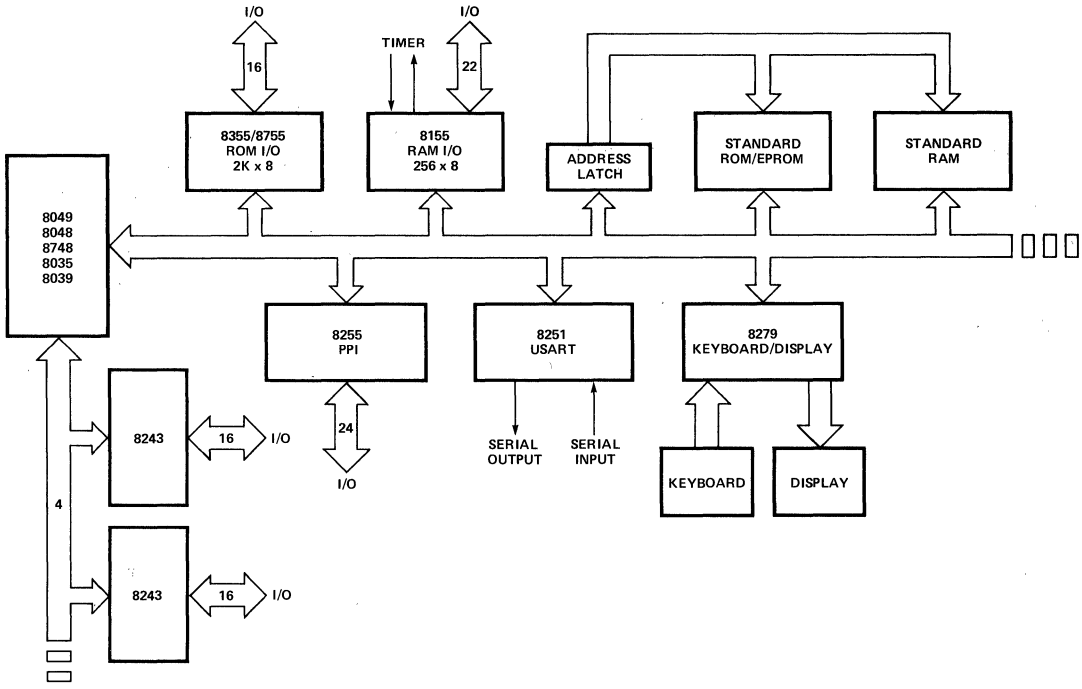
OUTL should never be used in a system with external program memory, since latching BUS can cause the next instruction, if external, to be fetched improperly.

Port 2 Operations

The lower half of Port 2 can be used in three different ways: as a quasi, bi-directional static port, as an 8243 expander port, and to address external program memory. In all cases outputs are driven low by an active device and driven high momentarily by an active device and held high by a 50KΩ resistor to +5V.

The port may contain latched I/O data prior to its use in another mode without affecting operation of either. If lower Port 2 (P20-3) is used to output address for an external program memory fetch the I/O information previously latched will be automatically removed temporarily while address is present then restored when the fetch is complete. However, if lower Port 2 is used to communicate with an 8243, previously latched I/O information will be removed and not restored. After an input from the 8243 P_{20-3} will be left in the input mode (floating). After an output to the 8243 P_{20-3} will contain the value written, ANDed, or ORed to the 8243 port.

MCS-48 EXPANSION CAPABILITY



Instruction Set

4

INSTRUCTION SET

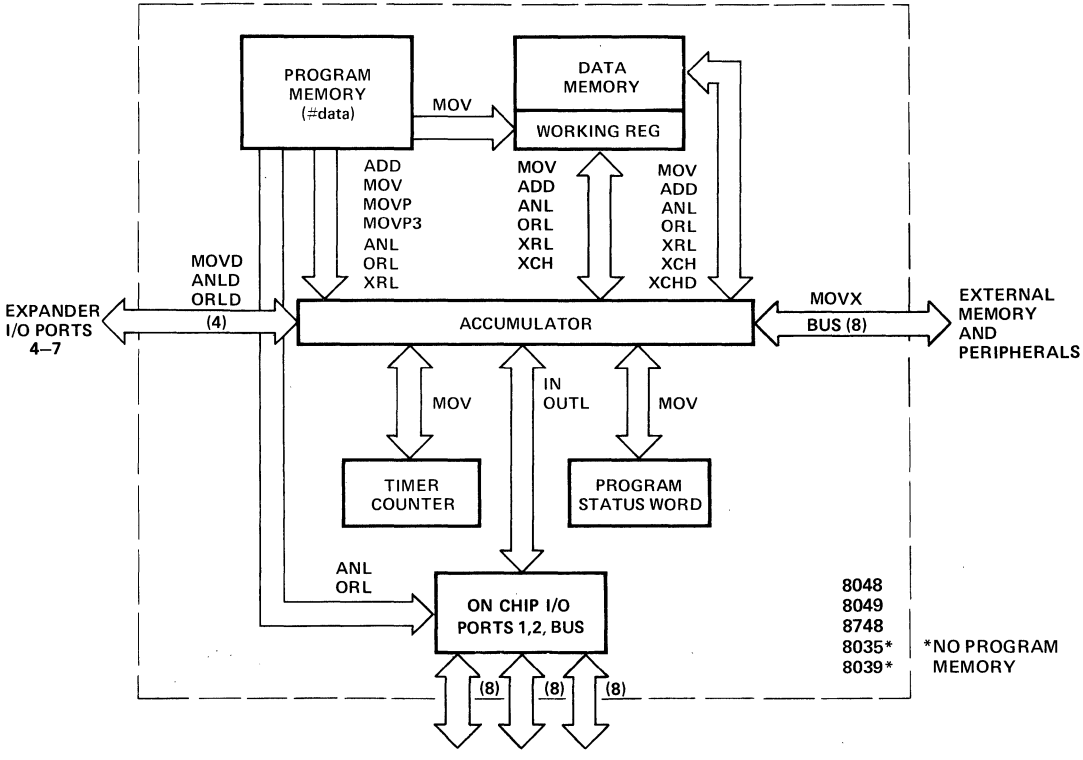
4.0 INTRODUCTION

The MCS-48 instruction set is extensive for a machine of its size and has been tailored to be straightforward and very efficient in its use of program memory. All instructions are either one or two bytes in length and over 70% are only one byte long. Also, all instructions execute in either one or two cycles (2.5μsec or 5.0μsec when using a 6 MHz XTAL) and over 50% of all instructions execute in a single cycle. Double cycle instructions include all immediate instructions, and all I/O instructions.

The MCS-48 microcomputers have been designed to efficiently handle arithmetic operations in both binary and BCD as well as to efficiently handle the single bit operations required in control applications. Special instructions have also been included to simplify loop counters, table lookup routines, and N-way branch routines.

Data Transfers

As can be seen in the accompanying diagram, the 8-bit accumulator is the central



point for all data transfers within the 8048. Data can be transferred between the 8 registers of each working register bank and the accumulator directly, i.e. the source or destination register is specified by the instruction. The remaining locations of the internal RAM array are referred to as Data Memory and are addressed indirectly via an address stored in either R0 or R1 of the active working register bank. R0 and R1 are also used to indirectly address external data memory when it is present. Transfers to and from internal RAM require one cycle while transfers to external RAM require two. Constants stored in Program Memory can be loaded directly to the accumulator and to the 8 working registers. Data can also be transfered directly between the accumulator and the on-board timer/counter or the accumulator and the Program Status word (PSW). Writing to the PSW alters machine status accordingly and provides a means of restoring status after an interrupt or of altering the stack pointer if necessary.

Accumulator Operations

Immediate data, data memory, or the working registers can be added with or without carry to the accumulator. These sources can also be ANDed, ORed, or Exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

In addition, the lower 4 bits of the accumulator can be exchanged with the lower 4-bits of any of the internal RAM locations. This instruction, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides for easy handling of 4-bit quantities, including BCD numbers. To facilitate BCD arithmetic, a Decimal Adjust instruction is included. This instruction is used to correct the result of the binary addition of two two-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the required BCD result.

Finally, the accumulator can be: incremented. decremented, cleared, or complemented and can be rotated left or right 1-bit at a time with or without carry.

Although there is no subtract instruction in the 8048, this operation can be easily implemented with three single-byte single-cycle instructions.

A value may be subtracted from the accumulator with the result in the accumulator by:

Complementing the accumulator
Adding the value to the accumulator
Complementing the accumulator.

Register Operations

The working registers can be accessed via the accumulator as explained above, or can be loaded immediate with constants from program memory. In addition, they can be incremented or decremented or used as loop counters using the decrement and jump, if not zero instruction, as explained under branch instructions.

All Data Memory including working registers can be accessed with indirect instructions via R0 and R1 and can be incremented.

Flags

There are four user accessible flags in the 8048: Carry, Auxillary Carry, F0, and F1. Carry indicates overflow of the accumulator, and Auxillary Carry is used to indicate overflow between BCD digits and is used during decimal adjust operation. Both Carry and Auxillary Carry are accessible as part of the program status word and are stored on the stack during subroutines. F0 and F1 are undedicated general purpose flags to be used as the programmer desires. Both flags can be cleared or complemented and tested by conditional jump instructions. F0 is also accessible via the Program Status word and is stored on the stack with the carry flags.

Branch Instructions

The unconditional jump instruction is two bytes and allows jumps anywhere in the first

2K words of program memory. Jumps to the second 2K of memory (4K words are directly addressible) are made by first executing a select memory bank instruction then executing the jump instruction. The 2K boundary can only be crossed via a jump or subroutine call instruction i.e. the bank switch does not occur until a jump is executed. Once a memory bank has been selected all subsequent jumps will be to the selected bank until another select memory bank instruction is executed. A subroutine in the opposite bank can be accessed by a select memory bank instruction followed by a call instruction. Upon completion of the subroutine execution will automatically return to the original bank; however, unless the original bank is reselected, the next jump instruction encountered will again transfer execution to the opposite bank.

Conditional jumps can test the following inputs and machine status:

T0 Input pin
T1 Input pin
$\overline{INT}$ Input pin
Accumulator Zero
Any bit of Accumulator
Carry Flag
F0 Flag
F1 Flag

Conditional jumps allow a branch to any address within the current page (256 words) of execution. The conditions tested are the instantaneous values at the time the conditional jump is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself not an intermediate zero flag.

The decrement register and jump if not zero instruction combines a decrement and a branch instruction to create an instruction very useful in implementing a loop counter. This instruction can designate any one of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A single byte indirect jump instruction allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator points to a location in program memory which contains the jump address. The 8-bit jump address refers to the current page of execution. This instruction could be used, for instance, to vector to any one of several routines based on an ASCII character which has been loaded in the accumulator. In this way ASCII key inputs can be used to initiate various routines.

Subroutines

Subroutines are entered by executing a call instruction. Calls can be made like unconditional jumps to any address in a 2K word bank and jumps across the 2K boundary are executed in the same manner. Two separate return instructions determine whether or not status (upper 4-bits of PSW) is restored upon return from the subroutine.

The return and restore status instruction also signals the end of an interrupt service routine if one has been in progress.

Timer Instructions

The 8-bit on board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting. The counter can be started as a timer with an internal clock source or as an event counter or timer with an external clock applied to the T1 input pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

Control Instructions

Two instructions allow the external interrupt source to be enabled or disabled. Interrupts are initially disabled and are automatically disabled while an interrupt service routine is in progress and re-enabled afterward.

There are four memory bank select instructions, two to designate the active working register bank and two to control program

memory banks. The operation of the program memory bank switch is explained in section 3.1.2. The working register bank switch instructions allow the programmer to immediately substitute a second 8 register working register bank for the one in use. This effectively provides 16 working registers or it can be used as a means of quickly saving the contents of the registers in response to an interrupt. The user has the option to switch or not to switch banks on interrupt. However, if the banks are switched, the original bank will be automatically restored upon execution of a return and restore status instruction at the end of the interrupt service routine.

A special instruction enables an internal clock, which is the XTAL frequency divided by three, to be output on pin T0. This clock can be used as a general purpose clock in the users system. This instruction should be used only to initialize the system since the clock output can be disabled only by application of system reset.

Input/Output Instructions

Ports 1 and 2 are 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs are not latched and must be read while inputs are present. In addition, immediate data from program memory can be ANDed or ORed directly to Port 1 and Port 2 with the result remaining on the port. This allows "masks" stored in program memory to selectively set or reset individual bits of the I/O ports. Ports 1 and 2 are configured to allow input on a given pin by first writing a "1" out to the pin.

An 8-bit port called BUS can also be accessed via the accumulator and can have statically latched outputs as well. It too can have immediate data ANDed or ORed directly to its outputs, however, unlike ports 1 and 2, all eight lines of BUS must be treated as either input or output at any one time. In addition to being a static port, BUS can be used as a true synchronous bi-directional port using the Move External instructions used to access external data memory. When these instructions are executed a corresponding $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ pulse is generated and data is valid only at that time. When data is not being transferred BUS is in a high impedance state.

The basic three on board I/O ports can be expanded via a 4-bit expander bus using half of port 2. I/O expander devices on this bus consist of four 4-bit ports which are addressed as ports 4 through 7. These ports have their own AND and OR instructions like the on board ports as well as move instructions to transfer data in or out. The expander AND and OR instructions, however, combine the contents of accumulator with the selected port rather than immediate data as is done with the on board ports.

I/O devices can also be added externally using the BUS port as the expansion bus. In this case the I/O ports become "memory mapped", i.e. they are addressed in the same way as external data memory and exist in the external data memory address space addressed by pointer register R0 or R1.

4.1 Instruction Set Description

The following pages describe the MCS-48 instruction set in detail. The instruction set is first summarized with instructions grouped functionally. This summary page is followed by a detailed description listed alphabetically by mnemonic opcode.

The alphabetical listing includes the following information:

Mnemonic
Machine Code
Verbal Description
Symbolic Description
Assembly Language Example

The machine code is represented with the most significant bit (7) to the left and two byte instructions are represented with the first byte on the left. The assembly language examples are formulated as follows:

```
Arbitrary
Label:  Mnemonic, Operand;   Descriptive Comment
```

See section 1.2.2 for a description and example of an assembly language program.

8048/8049
INSTRUCTION SET SUMMARY

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and jump	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag =1	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL addr	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A,@R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

8021
INSTRUCTION SET SUMMARY

	Mnemonic		Description	Bytes	Cycle
Accumulator	ADD	A,R	Add register to A	1	1
	ADD	A,@R	Add data memory to A	1	1
	ADD	A,#data	Add immediate to A	2	2
	ADDC	A,R	Add with carry	1	1
	ADDC	A,@R	Add with carry	1	1
	ADDC	A,#data	Add with carry	2	2
	ANL	A,R	And register to A	1	1
	ANL	A,@R	And data memory to A	1	1
	ANL	A,#data	And immediate to A	2	2
	ORL	A,R	Or register to A	1	1
	ORL	A,@R	Or data memory to A	1	1
	ORL	A,#data	Or immediate to A	2	2
	XRL	A,R	Exclusive Or register to A	1	1
	XRL	A,@R	Exclusive or data memory to A	1	1
	XRL	A,#data	Exclusive or immediate to A	2	2
	INC	A	Increment A	1	1
	DEC	A	Decrement A	1	1
	CLR	A	Clear A	1	1
	CPL	A	Complement A	1	1
	DA	A	Decimal Adjust A	1	1
	SWAP	A	Swap nibbles of A	1	1
	RL	A	Rotate A left	1	1
	RLC	A	Rotate A left through carry	1	1
	RR	A	Rotate A right	1	1
	RRC	A	Rotate A right through carry	1	1
Input/Output	IN	A,P	Input port to A	1	2
	OUTL	P,A	Output A to port	1	2
	MOVD	A,P	Input Expander port to A	1	2
	MOVD	P,A	Output A to Expander port	1	2
	ANLD	P,A	And A to Expander port	1	2
	ORLD	P,A	Or A to Expander port	1	2
Registers	INC	R	Increment register	1	1
	INC	@R	Increment data memory	1	1

	Mnemonic		Description	Bytes	Cycle
Branch	JMP	addr	Jump unconditional	2	2
	JMPP	@A	Jump indirect	1	2
	DJNZ	R,addr	Decrement register and Jump on R not zero	2	2
	JC	addr	Jump on Carry = 1	2	2
	JNC	addr	Jump on Carry = 0	2	2
	JZ	addr	Jump on A Zero	2	2
	JNZ	addr	Jump on A not Zero	2	2
	JT1	addr	Jump on T1 = 1	2	2
	JNT1	addr	Jump on T1 = 0	2	2
	JTF	addr	Jump on timer flag	2	2
Subroutine	CALL	addr	Jump to subroutine	2	2
	RET		Return	1	2
Flags	CLR	C	Clear Carry	1	1
	CPL	C	Complement Carry	1	1
Data Moves	MOV	A,R	Move register to A	1	1
	MOV	A,@R	Move data memory to A	1	1
	MOV	A,#data	Move immediate to A	2	2
	MOV	R,A	Move A to register	1	1
	MOV	@R,A	Move A to data memory	1	1
	MOV	R,#data	Move immediate to register	2	2
	MOV	@R,#data	Move immediate to data memory	2	2
	XCH	A,R	Exchange A and register	1	1
	XCH	A,@R	Exchange A and data memory	1	2
	XCHD	A,@R	Exchange nibble of A and register	1	1
	MOVP	A,@A	Move to A from current page	1	2
Timer/Counter	MOV	A,T	Read Timer/Counter	1	1
	MOV	T,A	Load Timer/Counter	1	1
	STRT	T	Start Timer	1	1
	STRT	CNT	Start Counter	1	1
	STOP	TCNT	Stop Timer/Counter	1	1
	NOP		No Operation	1	1

Instruction Set — The following instructions, which are found in the 8748, have been deleted from the 8021 instruction set.

Data Moves	Registers	Branch	Timer	Control	Input/Output
MOV A,PSW	DEC R	JT0 addr	EN TCNTI	EN I	ANL P,#data
MOV PSW,A		JNT0 addr	DIS TCNTI	DIS I	ORL P,#data
MOVX A,@R	Flags	JF0 addr		SEL RB0	INS A,BUS *
MOVX @R,A		JF1 addr	Subroutine	SEL RB1	OUTL BUS,A *
MOVP3 A,@A	CLR F0	JNI addr		SEL MB0	ANL BUS,#data
	CPL F0	JBb addr	RETR	SEL MB1	ORL BUS,#data
	CLR F1			ENT0 CLK	
	CPL F1				

*These Instructions have been replaced in the 8021 by IN A,PO and OUTL PO,A respectively.

INSTRUCTION SET SUMMARY

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
Accumulator	ADD A,R_r	Add register to A	1	1	68-6F
	ADD A,@ R	Add data memory to A	1	1	60-61
	ADD A,#data	Add immediate to A	2	2	03
	ADDC A,R_r	Add register with carry	1	1	78-7F
	ADDC A,@ R	Add data memory with carry	1	1	70-71
	ADDC A,#data	Add immediate with carry	2	2	13
	ANL A,R_r	And register to A	1	1	58-5F
	ANL A,@ R	And data memory to A	1	1	50-51
	ANL A,#data	And immediate to A	2	2	53
	ORL A,R_r	Or register to A	1	1	48-4F
	ORL A,@ R	Or data memory to A	1	1	40-41
	ORL A,#data	Or immediate to A	2	2	43
	XRL A,R_r	Exclusive Or register to A	1	1	D8-DF
	XRL A,@ R	Exclusive Or data memory to A	1	1	D0-D1
	XRL A,#data	Exclusive Or immediate to A	2	2	D3
	INC A	Increment A	1	1	17
	DEC A	Decrement A	1	1	07
	CLR A	Clear A	1	1	27
	CPL A	Complement A	1	1	37
	DA A	Decimal adjust A	1	1	57
	SWAP A	Swap nibbles of A	1	1	47
	RL A	Rotate A left	1	1	E7
	RLC A	Rotate A left through carry	1	1	F7
	RR A	Rotate A right	1	1	77
	RRC A	Rotate A right through carry	1	1	67
Input/Output	IN A, P_p	Input port to A	1	2	08,09,0A
	OUTL P_pA	Output A to port	1	2	90,39,3A
	MOVD A,P_p	Input expander port to A	1	2	0C-0F
	MOVD P_p,A	Output A to expander port	1	2	3C-3F
	ANLD P_p,A	And A to expander port	1	2	9C-9F
	ORLD P_p,A	Or A to expander port	1	2	8C-8F
Registers	INC R_r	Increment register	1	1	18-1F
	INC @ R	Increment data memory	1	1	10-11
Branch	JMP addr	Jump unconditional	2	2	04,24,44,64, 84,A4,C4,E4
	JMPP @ A	Jump indirect	1	2	B3
	DJNZ R,addr	Decrement register and jump on R not zero	2	2	E8-EF
	JC addr	Jump on carry=1	2	2	F6
	JNC addr	Jump on carry=0	2	2	E6
	JZ addr	Jump on A zero	2	2	C6
	JNZ addr	Jump on A not zero	2	2	96

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
	JT0	Jump on T0=1	2	2	36
	JNT0	Jump on T0=0	2	2	26
	JT1 addr	Jump on T1=1	2	2	56
	JNT1 addr	Jump on T1=0	2	2	46
	JTF addr	Jump on timer flag	2	2	16
Subroutine	CALL addr	Jump to subroutine	1	2	14,34,54,74 94,B4,D4,F4
	RET	Return	1	2	83
Flags	CLR C	Clear carry	1	1k	97
	CPL C	Complement carry	1	1	A7
Data Moves	MOV A,R_r	Move register to A	1	1	F8-FF
	MOV A,@ R	Move data memory to A	1	1	F0-F1
	MOV A,#data	Move immediate to A	2	2	23
	MOV R_r,A	Move A to register	1	1	A8-AF
	MOV @ R,A	Move A to data memory	1	1	A0-A1
	MOV R_r,#data	Move immediate to register	2	2	B8-BF
	MOV@R,#data	Move immediate to data memory	2	2	B0-B1
	XCH A, R_r	Exchange A and register	1	1	28-2F
	XCH A,@ R	Exchange A and data memory	1	1	20-21
	XCHD a,@ R	Exchange nibble of A and register	1	1	30-31
	MOVP A,@ A	Move to A from current page	1	2	A3
Timer/Counter	MOV A,T	Read timer/counter	1	1	42
	MOV T,A	Load timer/counter	1	1	62
	STRT T	Start timer	1	1	55
	STRT CNT	Start counter	1	1	45
	STOP TCNT	Stop timer/counter	1	1	65
A/D Converter	RAD	Move conversion result register to A	1	2	80
	SEL AN0	Select analog input zero	1	1	85
	SEL AN1	Select analog input one	1	1	95
Interrupts	EN I	Enable external interrupt	1	1	05
	DIS I	Disable external interrupt	1	1	15
	EN TCNTI	Enable timer/counter interrupt	1	1	25
	DIS TCNTI	Disable timer/counter interrupt	1	1	35
	RET I	Return from interrupt	1	2	93
	NOP	No operation	1	1	00

Instruction Set — The following instructions, which are found in the 8748, have been deleted from the 8022 instruction set.

Data Moves	Registers	Branch		Control	Input/Output
MOV A,PSW	DEC R				ANL P,#data
MOV PSW,A					ORL P,#data
MOVX A,@R	Flags	JF0 addr		SEL RB0	INS A,BUS *
MOVX @R,A		JF1 addr	Subroutine	SEL RB1	OUTL BUS,A *
MOVP3 A,@A	CLR F0	JNI addr	RETR	SEL MB0	ANL BUS,#data
	CPL F0	JBb addr		SEL MB1	ORL BUS,#data
	CLR F1			ENTO CLK	
	CPL F1				

*These Instructions have been replaced in the 8022 by IN A,PO and OUTL PO,A respectively.

MCS-48™ INSTRUCTION SET

SYMBOLS AND ABBREVIATIONS USED

A	Accumulator
AC	Auxillary Carry
addr	12-Bit Program Memory Address
Bb	Bit Designator (b=0-7)
BS	Bank Switch
BUS	BUS Port
C	Carry
CLK	Clock
CNT	Event Counter
CRR	Conversion Result Register
D	Mnemonic for 4-Bit Digit (Nibble)
data	8-Bit Number or Expression
DBF	Memory Bank Flip-Flop
F0, F1	Flag 0, Flag 1
I	Interrupt
P	Mnemonic for "in-page" Operation
PC	Program Counter
Pp	Port Designator (p=1, 2 or 4-7)
PSW	Program Status Word
Rr	Register Designator (r=0, 1 or 0-7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T0, T1	Test 0, Test 1
X	Mnemonic for External RAM
#	Immediate Data Prefix
@	Indirect Address Prefix
$	Current Value of Program Counter
(X)	Contents of X
((X))	Contents of Location Addressed by X
←	Is Replaced by

ADD A,R_r Add Register Contents to Accumulator

0 1 1 0	1 r r r

The contents of register 'r' are added to the accumulator. Carry is affected.

(A) ← (A) + (Rr) r=0-7

Example:
```
ADDREG: ADD A,R6    ;ADD REG 6 CONTENTS
                    ;TO ACC
```

ADD A,@R_r Add Data Memory Contents to Accumulator

0 1 1 0	0 0 0 r

The contents of the resident data memory location addressed by register 'r' bits 0-5*are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) r=0-1

Example:
```
ADDM: MOV R0, #01FH  ;MOVE '1F' HEX TO REG 0
      ADD A, @R0     ;ADD VALUE OF LOCATION
                     ;31 TO ACC
```

ADD A,#data Add Immediate Data to Accumulator

0 0 0 0	0 0 1 1	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected

(A) ← (A) + data

Example:
```
ADDID: ADD A,#ADDER: ;ADD VALUE OF SYMBOL
                     ;'ADDER' TO ACC
```

ADDC A,R_r Add Carry and Register Contents to Accumulator

0 1 1 1	1 r r r

The content of the carry bit is added to accumulator location 0 and the carry bit cleared. The contents of register 'r' are then added to the accumulator. Carry is affected.

(A) ← (A)+(Rr)+(C) r=0-7

Example:
```
ADDRGC: ADDC A,R4   ;ADD CARRY AND REG 4
                    ;CONTENTS TO ACC
```


*0-6 for 8039/8049

ADDC A,@R_r Add Carry and Data Memory Contents to Accumulator

0 1 1 1	0 0 0 r

The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the contents of the resident data memory location addressed by register 'r' bits 0-5*are added to the accumulator. Carry is affected.

(A) ← (A)+((Rr))+(C) r=0-1

Example:
```
ADDMC: MOV R1,#40     ;MOVE '40' DEC TO REG 1
       ADDC A,@R1     ;ADD CARRY AND LOCATION 40
                      ;CONTENTS TO ACC
```

ADDC A,#data Add Carry and Immediate Data to Accumulator

0 0 0 1	0 0 1 1	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the specified data is added to the accumulator. Carry is affected.

(A) ← (A)+data+(C)

Example:
```
       ADDC A,#225    ;ADD CARRY AND '225' DEC
                      ;TO ACC
```

ANL A,R_r Logical AND Accumulator With Register Mask

0 1 0 1	1 r r r

Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

(A) ← (A) AND (Rr) r=0-7

Example:
```
ANDREG: ANL A,R3      ;'AND' ACC CONTENTS WITH MASK
                      ;IN REG 3
```

ANL A,@R_r Logical AND Accumulator With Memory Mask

0 1 0 1	0 0 0 r

Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'r', bits 0-5*.

(A) ← (A) AND ((Rr)) r=0-1

Example:
```
ANDDM: MOV R0,#03FH   ;MOVE '3F' HEX TO REG 0
       ANL A, @R0     ;'AND' ACC CONTENTS WITH
                      ;MASK IN LOCATION 63
```


*0-6 for 8039/8049

ANL A,#data Logical AND Accumulator With Immediate Mask

0 1 0 1	0 0 1 1	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.

(A) ← (A) AND data

Examples:
```
ANDID: ANL A,#0AFH    ;'AND' ACC CONTENTS
                      ;WITH MASK 10101111
       ANL A,#3+X/Y   ;'AND' ACC CONTENTS
                      ;WITH VALUE OF EXP
                      ;'3+X/Y'
```

ANL BUS,#data Logical AND BUS With Immediate Mask (Not in 8021, 8022)

1 0 0 1	1 0 0 0	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on the BUS port is logically ANDed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS, A' instruction.

(BUS) ← (BUS) AND data

Example:
```
ANDBUS: ANL BUS, #MASK ;'AND' BUS CONTENTS
                       ;WITH MASK EQUAL VALUE
                       ;OF SYMBOL 'MASK'
```

ANL Pp,#data Logical AND Port 1-2 With Immediate Mask (Not in 8021, 8022)

1 0 0 1	1 0 p p	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

(Pp) ← (Pp) AND data p=1-2

Example:
```
ANDP2: ANL P2,#0F0H   ;'AND' PORT 2 CONTENTS
                      ;WITH MASK 'F0' HEX
                      ;(CLEAR P20-23)
```

ANLD Pp,A Logical AND Port 4-7 With Accumulator Mask

1 0 0 1	1 1 p p

This is a 2-cycle instruction. Data on port 'p' is logically ANDed with the digit mask contained in accumulator bits 0-3.

(Pp) ← (Pp) AND (A0-3) p=4-7

Note: The mapping of port 'p' to opcode bits 0-1 is as follows:

1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:
```
ANDP4: ANLD P4,A        ;'AND' PORT 4 CONTENTS
                        ;WITH ACC BITS 0-3
```

CALL address Subroutine Call

a_{10} a_9 a_8 1	0 1 0 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. The program counter and PSW bits 4-7 are saved in the stack. The stack pointer (PSW bits 0-2) is updated. Program control is then passed to the location specified by 'address'. PC bit 11 is determined by the most recent SEL MB instruction.

A CALL cannot begin in locations 2046-2047 or 4094-4095. Execution continues at the instruction following the CALL upon return from the subroutine.

$((SP)) \leftarrow (PC), (PSW_{4-7})$
$(SP) \leftarrow (SP)+1$
$(PC_{8-10}) \leftarrow (addr_{8-10})$
$(PC_{0-7}) \leftarrow addr_{0-7}$
$(PC_{11}) \leftarrow DBF$

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
          MOV R0,#50    ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
BEGADD:   MOV A,R1      ;MOVE CONTENTS OF REG 1
                        ;TO ACC
          ADD A,R2      ;ADD REG 2 TO ACC
          CALL SUBTOT   ;CALL SUBROUTINE 'SUBTOT'
          ADD A R3      ;ADD REG 3 TO ACC
          ADD A,R4      ;ADD REG 4 TO ACC
          CALL SUBTOT   ;CALL SUBROUTINE 'SUBTOT'
          ADD A,R5      ;ADD REG 5 TO ACC
          ADD A,R6      ;ADD REG 6 TO ACC
          CALL SUBTOT   ;CALL SUBROUTINE 'SUBTOT'

SUBTOT:   MOV @R0,A     ;MOVE CONTENTS OF ACC TO
                        ;LOCATION ADDRESSED BY
                        ;REG 0
          INC R0        ;INCREMENT REG 0
          RET           ;RETURN TO MAIN PROGRAM
```

CLR A Clear Accumulator

0 0 1 0	0 1 1 1

The contents of the accumulator are cleared to zero.

A ← 0

CLR C Clear Carry Bit

1 0 0 1	0 1 1 1

During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPL C, RRC, and DAA instructions. This instruction resets the carry bit to zero.

C ← 0

CLR F1 Clear Flag 1 (Not in 8021, 8022)

1 0 1 0	0 1 0 1

Flag 1 is cleared to zero.

(F1) ← 0

CLR F0 Clear Flag 0 (Not in 8021, 8022)

1 0 0 0	0 1 0 1

Flag 0 is cleared to zero.

(F0) ← 0

CPL A Complement Accumulator

0 0 1 1	0 1 1 1

The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.

(A) ← NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A          ;ACC CONTENTS ARE COMPLE-
                     ;MENTED TO 10010101
```

CPL C Complement Carry Bit

1 0 1 0	0 1 1 1

The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.

(C) ← NOT (C)

Example: Set C to one; current setting is unknown.

```
CTO1: CLR C          ;C IS CLEARED TO ZERO
      CPL C          ;C IS SET TO ONE
```

CPL F0 Complement Flag 0 (Not in 8021, 8022)

1 0 0 1	0 1 0 1

The setting of flag 0 is complemented; one is changed to zero, and zero is changed to one.

F0← NOT (F0)

CPL F1 Complement Flag 1 (Not in 8021, 8022)

1 0 1 1	0 1 0 1

The setting of flag 1 is complemented; one is changed to zero, and zero is changed to one.

(F1)← NOT (F1)

DA A Decimal Adjust Accumulator

0 1 0 1	0 1 1 1

The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0-3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4-7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example: Assume accumulator contains 10011011.

```
DA A                        ;ACC ADJUSTED TO 00000001
                            ;WITH C SET
C  AC  7    4  3    0
0   0  1 0 0 1  1 0 1 1
                   0 1 1 0      ADD SIX TO BITS 0-7
0   0  1 0 1 0  0 0 0 1
       0 1 1 0                  ADD SIX TO BITS 4-7
1   0  0 0 0 0  0 0 0 1          OVERFLOW TO C
```

DEC A Decrement Accumulator

0 0 0 0	0 1 1 1

The contents of the accumulator are decremented by one.

(A)← (A)-1

Example: Decrement contents of external data memory location 63.

```
MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
MOVX A,@R0       ;MOVE CONTENTS OF LOCATION 63
                 ;TO ACC
DEC A            ;DECREMENT ACC
MOVX @R0,A       ;MOVE CONTENTS OF ACC TO
                 ;LOCATION 63 IN EXPANDED
                 ;MEMORY
```

DEC R_r Decrement Register (Not in 8021, 8022)

1 1 0 0	1 r r r

The contents of working register 'r' are decremented by one.

(Rr)← (Rr)-1 r=0-7

Example: DECR1: DEC R1 ;DECREMENT CONTENTS OF REG 1

DIS I Disable External Interrupt (Not in 8021)

0 0 0 1	0 1 0 1

External interrupts are disabled. A low signal on the interrupt input pin has no effect.

DIS TCNTI Disable Timer/Counter Interrupt (Not in 8021)

0 0 1 1	0 1 0 1

Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ R_r, address Decrement Register and Test

1 1 1 0	1 r r r	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Register 'r' is decremented and tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits, that is, the jump must be to a location within the current 256-location page.

(Rr)← (Rr)-1 r=0-7
If Rr not 0
(PC_{0-7})← addr

Note: A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example: Increment values in data memory locations 50-54.

```
        MOV R0,#50      ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
        MOV R3,#5       ;MOVE '5' DEC TO COUNTER
                        ;REG 3
INCRT:  INC @R0         ;INCREMENT CONTENTS OF
                        ;LOCATION ADDRESSED BY
                        ;REG 0
        INC R0          ;INCREMENT ADDRESS IN REG 0
        DJNZ R3, INCRT  ;DECREMENT REG 3 — JUMP TO
                        ;'INCRT' IF REG 3 NONZERO
        NEXT —          ;'NEXT' ROUTINE EXECUTED
                        ;IF R3 IS ZERO
```

EN I Enable External Interrupt (Not in 8021)

0 0 0 0	0 1 0 1

External interrupts are enabled. A low signal on the interrupt input pin initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt (Not in 8021)

0 0 1 0	0 1 0 1

Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

ENT0 CLK Enable Clock Output (Not in 8021, 8022)

0 1 1 1	0 1 0 1

The test 0 pin is enabled to act as the clock output. This function is disabled by a system reset.

Example:

```
EMTST0: ENT0 CLK        ;ENABLE T0 AS CLOCK OUTPUT
```

IN A,Pp Input Port or Data to Accumulator

0 0 0 0	1 0 p p

This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator. In the 8021 IN A,P2 inputs P_{20}-P_{23} to A_0-A_3 while A_4-A_7 is set to zero.

(A) ← (Pp) p=1-2

```
Example: INP12: IN A,P1          ;INPUT PORT 1 CONTENTS
                                 ;TO ACC
         MOV R6,A                ;MOVE ACC CONTENTS TO
                                 ;REG 6
         IN A,P2                 ;INPUT PORT 2 CONTENTS
                                 ;TO ACC
         MOV R7,A                ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

0 0 0 1	0 1 1 1

The contents of the accumulator are incremented by one.

(A) ← (A)+1

Example: Increment contents of location 100 in external data memory.

```
INCA: MOV R0,#100        ;MOVE '100' DEC TO ADDRESS
                         ;REG 0
      MOVX A,@R0         ;MOVE CONTENTS OF LOCATION
                         ;100 TO ACC
      INC A              ;INCREMENT A
      MOVX @R0,A         ;MOVE ACC CONTENTS TO
                         ;LOCATION 100
```

INC R_r Increment Register

0 0 0 1	1 r r r

The contents of working register 'r' are incremented by one.

(Rr) ← (Rr)+1 r=0-7

```
Example: INCR0: INC R0           ;INCREMENT ADDRESS REG 0
```

INC @R_r Increment Data Memory Location

0 0 0 1	0 0 0 r

The contents of the resident data memory location addressed by register 'r' bits 0-5*are incremented by one.

((Rr)) ← ((Rr))+1 r=0-1

```
Example: INCDM: MOV R1,#03FH     ;MOVE ONES TO REG 1
         INC @R1                 ;INCREMENT LOCATION 63
```


*0-6 for 8039/8049

IN A,P0 Input of Port 0 Data to Accumulator (8021, 8022 Only)

Same as INS A,BUS except no $\overline{RD}$ pulse generated.

INS A,BUS Strobed Input of BUS Data to Accumulator

0 0 0 0	1 0 0 0

This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the $\overline{RD}$ pulse is dropped. (Refer to section on programming memory expansion for details).

(A)← (BUS)

Example:
```
INPBUS: INS A,BUS      ;INPUT BUS CONTENTS
                       ;TO ACC
```

JBb address Jump If Accumulator Bit is Set (Not in 8021, 8022)

$b_2\ b_1\ b_0$ 1	0 0 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

b=0-7

(PC_{0-7})← addr — If Bb=1
(PC) = (PC)+2 — If Bb=0

Example:
```
JB4IS1: JB4 NEXT       ;JUMP TO 'NEXT' ROUTINE
                       ;IF ACC BIT 4=1
```

JC address Jump If Carry Is Set

1 1 1 1	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

(PC_{0-7})← addr — If C=1
(PC) = (PC)+2 — If C=0

Example:
```
JC1: JC OVFLOW         ;JUMP TO 'OVFLOW' ROUTINE
                       ;IF C=1
```

JF0 address Jump If Flag 0 Is Set (Not in 8021, 8022)

1 0 1 1	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

(PC_{0-7})← addr — If F0=1
(PC) = (PC)+2 — If F0=0

Example:
```
JF0IS1: JF0 TOTAL      ;JUMP TO 'TOTAL' ROUTINE
                       ;IF F0=1
```

JF1 address Jump If Flag 1 Is Set (Not in 8021, 8022)

0 1 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

$(PC_{0-7}) \leftarrow$ addr	If F1=1
(PC) = (PC)+2	IF F1=0

Example:
```
JF1IS1: JF1 FILBUF      ;JUMP TO 'FILBUF'
                        ;ROUTINE IF F1=1
```

JMP address Direct Jump Within 2K Block

a_{10} a_9 a_8 0	0 1 0 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Bits 0-10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 is determined by the most recent SELECT MB instruction.

$(PC_{8-10}) \leftarrow$ addr 8-10
$(PC_{0-7}) \leftarrow$ addr 0-7
$(PC_{11}) \leftarrow$ DBF

Example:
```
JMP SUBTOT              ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6                 ;JUMP TO INSTRUCTION SIX LOCATIONS
                        ;BEFORE CURRENT LOCATION
JMP 2FH                 ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump Within Page

1 0 1 1	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0-7).

$(PC_{0-7}) \leftarrow ((A))$

Example: Assume accumulator contains 0FH.
```
JMPPAG: JMPP @A         ;JUMP TO ADDRESS STORED IN
                        ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

1 1 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

	$(PC_{0-7}) \leftarrow$ addr	If C=0
	(PC) = (PC)+2	IF C=1
Example:	JC0: JNC NOVFLO	;JUMP TO 'NOVFLO' ROUTINE ;If C=0

JNI address Jump If Interrupt Input is Low (Not in 8021, 8022)

1 0 0 0 | 0 1 1 0	$a_7\ a_6\ a_5\ a_4$ | $a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if the interrupt input signal is low (=0), that is, an external interrupt has been signaled. (This signal initiates an interrupt service sequence if the external interrupt is enabled.)

	$(PC_{0-7}) \leftarrow$ addr	If I=0
	(PC) = (PC)+2	If I=1
Example:	LOC 3: JNI EXTINT	;JUMP TO 'EXTINT' ROUTINE ;If I=0

JNT0 address Jump If Test 0 Is Low (Not in 8021)

0 0 1 0 | 0 1 1 0	$a_7\ a_6\ a_5\ a_4$ | $a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address, if the test 0 signal is low

	$(PC_{0-7}) \leftarrow$ addr	If T0=0
	(PC) = (PC)+2	If T0=1
Example:	JT0LOW: JNT0 60	;JUMP TO LOCATION 60 DEC ;IF T0=0

JNT1 address Jump If Test 1 Is Low

0 1 0 0 | 0 1 1 0	$a_7\ a_6\ a_5\ a_4$ | $a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address, if the test 1 signal is low.

	$(PC_{0-7}) \leftarrow$ addr	If T1=0
	(PC) = (PC)+2	If T1=1

JNZ address Jump If Accumulator Is Not Zero

1 0 0 1 | 0 1 1 0	$a_7\ a_6\ a_5\ a_4$ | $a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control pases to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

	$(PC_{0-7}) \leftarrow$ addr	If A≠0
	(PC) = (PC)+2	If A=0
Example:	JACCN0: JNZ 0ABH	;JUMP TO LOCATION 'AB' HEX ;IF ACC VALUE IS NONZERO

JTF address Jump If Timer Flag Is Set

0 0 0 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register has overflowed. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

$(PC_{0-7}) \leftarrow$ addr	If TF=1
(PC) = (PC)+2	If TF=0

```
Example: JTF1: JTF TIMER      ;JUMP TO 'TIMER' ROUTINE
                              ;IF TF=1
```

JT0 address Jump If Test 0 Is High (Not in 8021)

0 0 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the test 0 signal is high (=1).

$(PC_{0-7}) \leftarrow$ addr	If T0=1
(PC) = (PC)+2	If T0=0

```
Example: JT0HI: JT0 53        ;JUMP TO LOCATION 53 DEC
                              ;IF T0=1
```

JT1 address Jump If Test 1 Is High

0 1 0 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the test 1 signal is high (=1).

$(PC_{0-7}) \leftarrow$ addr	If T1=1
(PC) = (PC)+2	If T1=0

```
Example: JT1HI: JT1 COUNT     ;JUMP TO 'COUNT' ROUTINE
                              ;IF T1=1
```

JZ address Jump If Accumulator Is Zero

1 1 0 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr	If A=0
(PC) = (PC)+2	If A≠0

```
Example: JACCO: JZ 0A3H       ;JUMP TO LOCATION 'A3' HEX
                              ;IF ACC VALUE IS ZERO
```

MOV A,#data Move Immediate Data to Accumulator

0 0 1 0	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator.

(A)← data

Example:
```
MOV A,#0A3H          ;MOVE 'A3' HEX TO ACC
```

MOV A,PSW Move PSW Contents to Accumulator (Not in 8021, 8022)

1 1 0 0	0 1 1 1

The contents of the program status word are moved to the accumulator.

(A)← (PSW)

Example: Jump to 'RB1SET' routine if PSW bank switch, bit 4, is set.

```
BSCHK: MOV A,PSW     ;MOVE PSW CONTENTS TO ACC
       JB4 RB1SET    ;JUMP TO 'RB1SET' IF ACC
                     ;BIT 4=1
```

MOV A,R_r Move Register Contents to Accumulator

1 1 1 1	1 r r r

8-bits of data are moved from working register 'r' into the accumulator.

(A)← (Rr) r=0-7

Example:
```
MAR: MOV A,R3        ;MOVE CONTENTS OF REG 3
                     ;TO ACC
```

MOV A,@R_r Move Data Memory Contents to Accumulator

1 1 1 1	0 0 0 r

The contents of the resident data memory location addressed by bits 0-5*of register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

(A)← ((Rr)) r=0-1

Example: Assume R1 contains 00110110.

```
MADM: MOV A,@R1      ;MOVE CONTENTS OF DATA MEM
                     ;LOCATION 54 TO ACC
```


*0-6 for 8039/8049

MOV A,T Move Timer/Counter Contents to Accumulator

0 1 0 0	0 0 1 0

The contents of the timer/event-counter register are moved to the accumulator.

(A)← (T)

Example: Jump to "EXIT" routine when timer reaches '64', that is, when bit 6 set — assuming initialization 64,

```
TIMCHK: MOV A,T     ;MOVE TIMER CONTENTS TO
                    ;ACC
        JB6 EXIT    ;JUMP TO 'EXIT' IF ACC BIT
                    ;6=1
```

MOV PSW,A Move Accumulator Contents to PSW (Not in 8021, 8022)

1 1 0 1	0 1 1 1

The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.

(PSW)← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.

```
INCPTR: MOV A,PSW   ;MOVE PSW CONTENTS TO ACC
        INC A       ;INCREMENT ACC BY ONE
        MOV PSW,A   ;MOVE ACC CONTENTS TO PSW
```

MOV R_r,A Move Accumulator Contents to Register

1 0 1 0	1 r r r

The contents of the accumulator are moved to register 'r'.

(Rr)← (A) r=0-7

Example:

```
MRA: MOV R0,A       ;MOVE CONTENTS OF ACC TO
                    ;REG 0
```

MOV R_r,#data Move Immediate Data to Register

1 0 1 1	1 r_2 r_1 r_0	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

(Rr)← data r=0-7

Examples:
```
MIR4: MOV R4,#HEXTEN ;THE VALUE OF THE SYMBOL
                     ;'HEXTEN' IS MOVED INTO
                     ;REG 4
MIR 5: MOV R5,#PI*(R*R) ;THE VALUE OF THE
                        ;EXPRESSION 'PI*(R*R)
                        ;IS MOVED INTO REG 5
MIR 6: MOV R6, #0ADH ;'AD' HEX IS MOVED INTO
                     ;REG 6
```

MOV @R_r,A Move Accumulator Contents to Data Memory

1 0 1 0	0 0 0 r

This is a 2-cycle instruction. The contents of the accumulator are moved to the resident data memory location whose address is specified by bits 0-5* of register 'r'. Register 'r' contents are unaffected.

((Rr)) ← (A) r=0-1

Example: Assume R0 contains 00000111.

```
MDMA: MOV @R0,A    ;MOVE CONTENTS OF ACC TO
                   ;LOCATION 7 (REG 7)
```

MOV @R_r,#data Move Immediate Data to Data Memory

1 0 1 1	0 0 0 r	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register 'r', bits 0-5*

((Rr)) ← data r=0-1

Examples: Move the hexadecimal value AC3F to locations 62-63.

```
MIDM: MOV R0,#62      ;MOVE '62' DEC TO ADDR REG 0
      MOV @R0,#0ACH   ;MOVE 'AC' HEX TO LOCATION 62
      INC R0          ;INCREMENT REG 0 TO '63'
      MOV @R0,#3FH    ;MOVE '3F' HEX TO LOCATION 63
```

MOV T,A Move Accumulator Contents to Timer/Counter

0 1 1 0	0 0 1 0

The contents of the accumulator are moved to the timer/event-counter register.

(T) ← (A)

Example: Initialize and start event counter.

```
INITEC: CLR A      ;CLEAR ACC TO ZEROS
        MOV T,A    ;MOVE ZEROS TO EVENT COUNTER
        STRT CNT   ;START COUNTER
```


*0-6 for 8039/8049

MOVD A,Pp Move Port 4-7 Data to Accumulator

0 0 0 0	1 1 p p

This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are zeroed.

(0-3) ← (Pp) p=4-7
(4-7) ← 0

Note: Bits 0-1 of the opcode are used to represent ports 4-7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits 1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:
```
INPPT5: MOVD A,P5      ;MOVE PORT 5 DATA TO ACC
                       ;BITS 0-3, ZERO ACC BITS 4-7
```

MOVD Pp,A Move Accumulator Data to Port 4-7

0 0 1 1	1 1 p p

This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See NOTE above regarding port mapping.)

(Pp) ← (A_{0-3}) p=4-7

Example: Move data in accumulator to ports 4 and 5.
```
OUTP45: MOVD P4,A      ;MOVE ACC BITS 0-3 TO PORT 4
        SWAP A         ;EXCHANGE ACC BITS 0-3 AND 4-7
        MOVD P5,A      ;MOVE ACC BITS 0-3 TO PORT 5
```

MOVP A,@A Move Current Page Data to Accumulator

1 0 1 0	0 0 1 1

The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0-7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored following this operation

(PC_{0-7}) ← (A)
(A) ← ((PC))

Note: This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the following page.

Example:
```
MOV128: MOV A,#128    ;MOVE '128' DEC TO ACC
        MOVP A,@A     ;CONTENTS OF 129th LOCATION
                      ;IN CURRENT PAGE ARE MOVED TO
                      ;ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator (Not in 8021, 8022)

1 1 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

$(PC_{0-7}) \leftarrow (A)$
$(PC_{8-11}) \leftarrow 0011$
$(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#0B8H ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH  ;LOGICAL AND ACC TO MASK BIT
                    ;7 (00111000)
        MOVP3 A,@A  ;MOVE CONTENTS OF LOCATION
                    ;'38' HEX IN PAGE 3 TO ACC
                    ;(ASCII '8')
```

Access contents of location in page 3 labelled TAB1. Assume current program location is not in page 3.

```
TABSCH: MOV A,#LOW TAB1 ;ISOLATE BITS 0-7 OF LABEL
                       ;ADDRESS VALUE
        MOVP3 A,@A     ;MOVE CONTENTS OF PAGE 3
                       ;LOCATION LABELED 'TAB1'
                       ;TO ACC
```

MOVX A,@R_r Move External-Data-Memory Contents to Accumulator

1 0 0 0	0 0 0 r

(Not in 8021, 8022)

This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

$(A) \leftarrow ((Rr))$ r=0-1

Example: Assume R1 contains 01110110.

```
MAXDM: MOVX A,@R1   ;MOVE CONTENTS OF LOCATION
                    ;118 TO ACC
```

MOVX @R_r,A Move Accumulator Contents to External Data Memory

1 0 0 1	0 0 0 r

(Not in 8021, 8022)

This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'r'. Register 'r' contents are unaffected.

((Rr)) ← A r=0-1

Example: Assume R0 contains 11000111.

```
MXDMA: MOVX @R0,A      ;MOVE CONTENTS OF ACC TO
                       ;LOCATION 199 IN EXPANDED
                       ;DATA MEMORY
```

NOP The NOP Instruction

0 0 0 0	0 0 0 0

No operation is performed. Execution continues with the following instruction.

ORL A,R_r Logical OR Accumulator With Register Mask

0 1 0 0	1 r r r

Data in the accumulator is logically ORed with the mask contained in working register 'r'.

(A) ← (A) OR (Rr) r=0-7

Example:

```
ORREG: ORL A,R4        ;'OR' ACC CONTENTS WITH
                       ;MASK IN REG 4
```

ORL A,@R_r Logical OR Accumulator With Memory Mask

0 1 0 0	0 0 0 r

Data in the accumulator is logically ORed with the mask contained in the resident data memory location referenced by register 'r', bits 0-5*

(A) ← (A) OR ((Rr)) r=0-1

Example:

```
ORDM: MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
      ORL A,@R0        ;'OR' ACC CONTENTS WITH MASK
                       ;IN LOCATION 63
```

ORL A,#data Logical OR Accumulator With Immediate Mask

0 1 0 0	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.

(A) ← (A) OR data

Example:

```
ORID: ORL A,#'X'       ;'OR' ACC CONTENTS WITH MASK
                       ;01011000 (ASCII VALUE OF 'X')
```


*0-6 for 8039/8049

ORL BUS,#data Logical OR BUS With Immediate Mask (Not in 8021, 8022)

1 0 0 0 | 1 0 0 0	$d_7\ d_6\ d_5\ d_4$ | $d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS,A' instruction.

(BUS) ← (BUS) OR data

```
Example: ORBUS: ORL BUS,#HEXMSK ;'OR' BUS CONTENTS WITH
                                ;MASK EQUAL VALUE OF SYMBOL
                                ;'HEXMSK'
```

ORL Pp, #data Logical OR Port 1 or 2 With Immediate Mask (Not in 8021, 8022)

1 0 0 0 | 1 0 p p	$d_7\ d_6\ d_5\ d_4$ | $d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.

(Pp) ← (Pp) OR data p=1-2

```
Example: ORP1: ORL P1, #0FFH  ;'OR' PORT 1 CONTENTS WITH
                              ;MASK 'FF' HEX ( SET PORT 1
                              ;TO ALL ONES)
```

ORLD Pp,A Logical OR Port 4-7 With Accumulator Mask

1 0 0 0 | 1 1 p p

This is a 2-cycle instruction. Data on port 'p' is logically ORed with the digit mask contained in accumulator bits 0-3.

(Pp) ← (Pp) OR (A_{0-3}) p=4-7

```
Example: ORP7: ORLD P7,A  ;'OR' PORT 7 CONTENTS
                          ;WITH ACC BITS 0-3
```

OUTL P0,A Output Accumulator Data to Port 0 (8021, 8022 Only)

1 0 0 1 | 0 0 0 0

OUTL BUS,A Output Accumulator Data to BUS (Not in 8021, 8022)

0 0 0 0 | 0 0 1 0

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to the BUS port and latched. The latched data remains valid until altered by another OUTL instruction. Any other instruction requiring use of the BUS port (except INS) destroys the contents of the BUS latch. This includes expanded memory operations (such as the MOVX instruction). Logical operations on BUS data (AND, OR) assume the OUTL BUS,A instruction has been issued previously.

Does not apply for OUTL P0,A of 8021, 8022

(BUS) ← (A)

```
Example: OUTLBP: OUTL BUS,A  ;OUTPUT ACC CONTENTS TO BUS
```

OUTL Pp,A Output Accumulator Data to Port 1 or 2

0 0 1 1	1 0 p p

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.

(Pp) ← (A) p=1-2

Example:
```
OUTLP: MOV A,R7     ;MOVE REG 7 CONTENTS TO ACC
       OUTL P2,A    ;OUTPUT ACC CONTENTS TO PORT 2
       MOV A,R6     ;MOVE REG 6 CONTENTS TO ACC
       OUTL P1,A    ;OUTPUT ACC CONTENTS TO PORT 1
```

RAD Move Conversion Result Register to Accumlator (8022 Only)

1 0 0 0	0 0 0 0

This is a two cycle instruction. The contents of the A/D conversion result register are moved to the accumulator.

(A) ← (CRR)

RET Return Without PSW Restore

1 0 0 0	0 0 1 1

This is a 2-cycle instruction. The stack pointer (PSW bits 0-2) is decremented. The program counter is then restored from the stack. PSW bits 4-7 are not restored.

(SP) ← (SP)-1
(PC) ← ((SP))

RETI Return From Interrupt (8022 Only)

1 0 0 1	0 0 1 1

This is a two cycle instruction. The stack pointer is decremented and the program counter is restored from the stack. Interrupt input logic is re-enabled.

(SP) ← (SP)-1
(PC) ← ((SP)) +1

RETR Return With PSW Restore (Not in 8021, 8022)

1 0 0 1	0 0 1 1

This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4-7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine by resetting the Interrupt in Progress flipflop.

(SP)← (SP)-1
(PC)← ((SP))
(PSW 4-7)← ((SP))

RL A Rotate Left Without Carry

1 1 1 0	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

(An+1)← (An)
(A0)← (A7) n=0-6

Example: Assume accumulator contains 10110001.

```
RLNC: RL A          ;NEW ACC CONTENTS ARE 01100011.
```

RLC A Rotate Left Through Carry

1 1 1 1	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

(An+1)← (An)
n=0-6
(A0)← (C)
(C)← (A7)

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C         ;CLEAR CARRY TO ZERO
      RLC A         ;ROTATE ACC LEFT, SIGN
                    ;BIT (7) IS PLACED IN CARRY
      RR A          ;ROTATE ACC RIGHT — VALUE
                    (BITS 0-6) IS RESTORED,
                    ;CARRY UNCHANGED, BIT 7
                    ;IS ZERO
```

RR A Rotate Right Without Carry

0 1 1 1	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position

(An) ← (An +1) n=0-6
(A7) ← (A0)

Example: Assume accumulator contains 10110001.

```
RRNC: RR A          ;NEW ACC CONTENTS ARE 11011000
```

RRC A Rotate Right Through Carry

0 1 1 0	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

(An) ← (An+1) n=0-6
(A7) ← (C)
(C) ← (A_0)

Example: Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRC A         ;CARRY IS SET AND ACC
                    ;CONTAINS 01011000
```

SEL AN0 Select Analog Input Zero (8022 Only)

1 0 0 1	0 1 0 1

SEL AN1 Select Analog Input One (8022 Only)

1 0 0 0	0 1 0 1

One of the two analog inputs to the A/D converter is selected. The conversion process is started. Restarting a sequence deletes the sequence in progress.

SEL MBO Select Memory Bank 0 (Not in 8021, 8022)

1 1 1 0	0 1 0 1

PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0-2047.

(DBF) ← 0

Example: Assume program counter contains 834 Hex.

```
SEL MBO          ;SELECT MEMORY BANK 0
JMP $+20         ;JUMP TO LOCATION
                 ;48 HEX
```

SEL MB1 Select Memory Bank 1 (Not in 8021, 8022)

1 1 1 1	0 1 0 1

PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 2048-4095.

(DBF) ← 1

SEL RB0 Select Register Bank 0 (Not in 8021, 8022)

1 1 0 0	0 1 0 1

PSW bit 4 is set to zero. References to working registers 0-7 address data memory locations 0-7. This is the recommended setting for normal program execution.

(BS) ← 0

SEL RB1 Select Register Bank 1 (Not in 8021, 8022)

1 1 0 1	0 1 0 1

PSW bit 4 is set to one. References to working registers 0-7 address data memory locations 24-31. This is the recommended setting for interrupt service routines, since locations 0-7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

(BS) ← 1

Example: Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

```
LOC3: JNI INIT    ;JUMP TO ROUTINE 'INIT' IF
                  ;INTERRUPT INPUT IS ZERO
```

```
INIT: MOV R7,A          ;MOVE ACC CONTENTS TO
                        ;LOCATION 7
      SEL RB1           ;SELECT REG BANK 1
      MOV R7,#0FAH      ;MOVE 'FA' HEX TO LOCATION 31
      .
      .
      .
      SEL RB0           ;SELECT REG BANK 0
      MOV A,R7          ;RESTORE ACC FROM LOCATION 7
      RETR              ;RETURN — RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event-Counter

0 1 1 0	0 1 0 1

This instruction is used to stop both time accumulation and event counting.

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START:  DIS TCNTI       ;DISABLE TIMER INTERRUPT
        CLR A           ;CLEAR ACC TO ZEROS
        MOV T,A         ;MOVE ZEROS TO TIMER
        MOV R7,A        ;MOVE ZEROS TO REG 7
        STRT T          ;START TIMER
MAIN:   JTF COUNT       ;JUMP TO ROUTINE 'COUNT'
                        ;IF TF=1 AND CLEAR TIMER FLAG
        JMP MAIN        ;CLOSE LOOP
COUNT:  INC R7          ;INCREMENT REG 7
        MOV A,R7        ;MOVE REG 7 CONTENTS TO ACC
        JB3 INT         ;JUMP TO ROUTINE 'INT' IF ACC
                        ;BIT 3 IS SET (REG 7=8)
        JMP MAIN        ;OTHERWISE RETURN TO ROUTINE
                        ;MAIN
        .
        .
        .
INT:    STOP TCNT       ;STOP TIMER
         JMP 7H         ;JUMP TO LOCATION 7 (TIMER)
                        ;INTERRUPT ROUTINE
```

STRT CNT Start Event Counter

0 1 0 0	0 1 0 1

The test 1 (T1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high-to-low transition on the T1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T1 input.

```
STARTC: EN TCNTI      ;ENABLE COUNTER INTERRUPT
        MOV A,#0FFH   ;MOVE 'FF' HEX (ONES) TO
                      ;ACC
        MOV T,A       ;MOVE ONES TO COUNTER
        STRT CNT      ;ENABLE T1 AS COUNTER
                      ;INPUT AND START
```

STRT T Start Timer

0 1 0 1	0 1 0 1

Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: CLR A         ;CLEAR ACC TO ZEROS
        MOV T,A       ;MOVE ZEROS TO TIMER
        EN TCNTI      ;ENABLE TIMER INTERRUPT
        STRT T        ;START TIMER
```

SWAP A Swap Nibbles Within Accumulator

0 1 0 0	0 1 1 1

Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

$(A_{4-7}) \rightleftarrows (A_{0-3})$

Example: Pack bits 0-3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0, #50   ;MOVE '50' DEC TO REG 0
        MOV R1, #51   ;MOVE '51' DEC TO REG 1
        XCHD A,@R0    ;EXCHANGE BITS 0-3 OF ACC
                      ;AND LOCATION 50
        SWAP A        ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@R1    ;EXCHANGE BITS 0-3 OF ACC AND
                      ;LOCATION 51
        MOV @R0,A     ;MOVE CONTENTS OF ACC TO
                      ;LOCATION 50
```

XCH A,R_r Exchange Accumulator-Register Contents

0 0 1 0	1 r r r

The contents of the accumulator and the contents of working register 'r' are exchanged.

(A) ⇄ (Rr) r=0-7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7     ;EXCHANGE CONTENTS OF REG 7
                     ;AND ACC
        MOV A, PSW   ;MOVE PSW CONTENTS TO ACC
        XCH A,R7     ;EXCHANGE CONTENTS OF REG 7
                     ;AND ACC AGAIN
```

XCH A,@R_r Exchange Accumulator and Data Memory Contents

0 0 1 0	0 0 0 r

The contents of the accumulator and the contents of the resident data memory location addressed by bits 0-5*of register 'r' are exchanged. Register 'r' contents are unaffected.

(A) ⇄ ((Rr)) r=0-1

Example: Decrement contents of location 52.

```
DEC52: MOV R0,#52    ;MOVE '52' DEC TO ADDRESS
                     ;REG 0
       XCH A,@R0     ;EXCHANGE CONTENTS OF ACC
                     ;AND LOCATION 52
       DEC A         ;DECREMENT ACC CONTENTS
       XCH A,@R0     ;EXCHANGE CONTENTS OF ACC
                     ;AND LOCATION 52 AGAIN
```

XCHD A,@R_r Exchange Accumulator and Data Memory 4-Bit Data

0 0 1 1	0 0 0 r

This instruction exchanges bits 0-3 of the accumulator with bits 0-3 of the data memory location addressed by bits 0-5*of register 'r'. Bits 4-7 of the accumulator, bits 4-7 of the data memory location, and the contents of register 'r' are unaffected.

(A_{0-3}) ⇄ ((Rr0-3)) r=0-1

*0-6 for 8039/8049

Example: Assume program counter contents have been stacked in locations 22-23.

```
XCHNIB: MOV R0,#23    ;MOVE '23' DEC TO REG 0
        CLR A         ;CLEAR ACC TO ZEROS
        XCHD A,@R0    ;EXCHANGE BITS 0-3 OF ACC
                      ;AND LOCATION 23 (BITS 8-11
                      ;OF PC ARE ZEROED, ADDRESS
                      ;REFERS TO PAGE 0)
```

XRL A,R_r Logical XOR Accumulator With Register Mask

1 1 0 1	1 r r r

Data in the accumulator in EXCLUSIVE ORed with the mask contained in working register 'r'.

(A) ← (A) XOR (Rr) r=0-7

Example:
```
XORREG: XRL A,R5    ;'XOR' ACC CONTENTS WITH
                    ;MASK IN REG 5
```

XRL A,@R_r Logical XOR Accumulator With Memory Mask

1 1 0 1	0 0 0 r

Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register 'r', bits 0-5*.

(A) ← (A) XOR ((Rr)) r=0-1

Example:
```
XORDM: MOV R1, #20H  ;MOVE '20' HEX TO REG 1
       XRL A,@R1     ;'XOR' ACC CONTENTS WITH MASK
                     ;IN LOCATION 32
```

XRL A,#data Logical XOR Accumulator With Immediate Mask

1 1 0 1	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

(A) ← (A) XOR data

Example:
```
XORID: XOR A,#HEXTEN ;XOR CONTENTS OF ACC WITH
                     ;MASK EQUAL VALUE OF SYMBOL
                     ;'HEXTEN'
```


*0-6 for 8039/8049

Application Examples

5

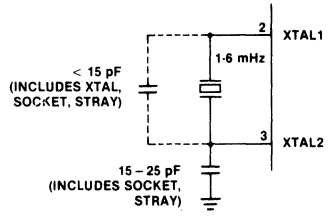
APPLICATION EXAMPLES

5.0 Introduction

The following chapter is organized in two sections, Hardware and Software. The hardware section gives examples of some typical configurations of MCS-48 components while software section gives assembly language listings of some common applications routines.

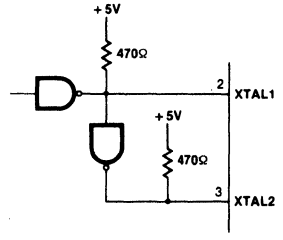
5.1 Hardware Examples

CRYSTAL OSCILLATOR MODE



CRYSTAL SERIES RESISTANCE SHOULD BE <75Ω AT 6 MHz; <180Ω AT 3.6 MHz.

DRIVING FROM EXTERNAL SOURCE



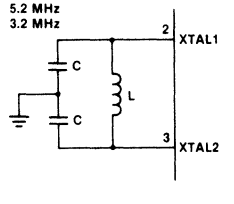
BOTH XTAL1 AND XTAL2 SHOULD BE DRIVEN. RESISTORS TO V_{CC} ARE NEEDED TO ENSURE V_{IH} = 3.8V IF TTL CIRCUITRY IS USED.

NOTE:
A STANDARD SERIES RESONANT XTAL SUCH AS CTS KNIGHTS MP060 OR CRYSTEK CY6B WILL PROVIDE BETTER THAN .1% FREQUENCY ACCURACY.

EACH C SHOULD BE APPROXIMATELY 20pF, INCLUDING STRAY CAPACITANCE. FOR MORE INFORMATION ON XTALS SEE INTEL APPLICATION NOTE AP-35.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

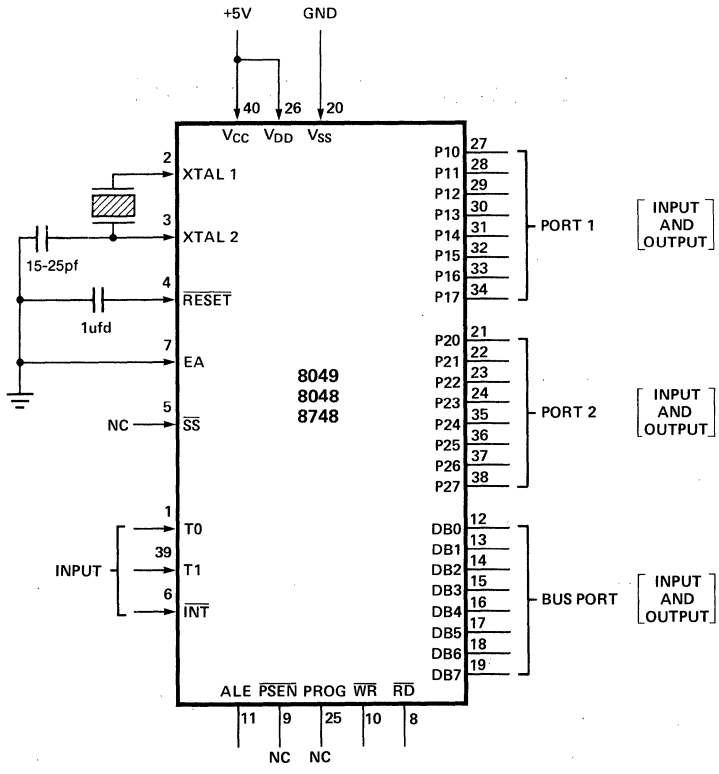
$$C' = \frac{C + 3C_{PP}}{2}$$

$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

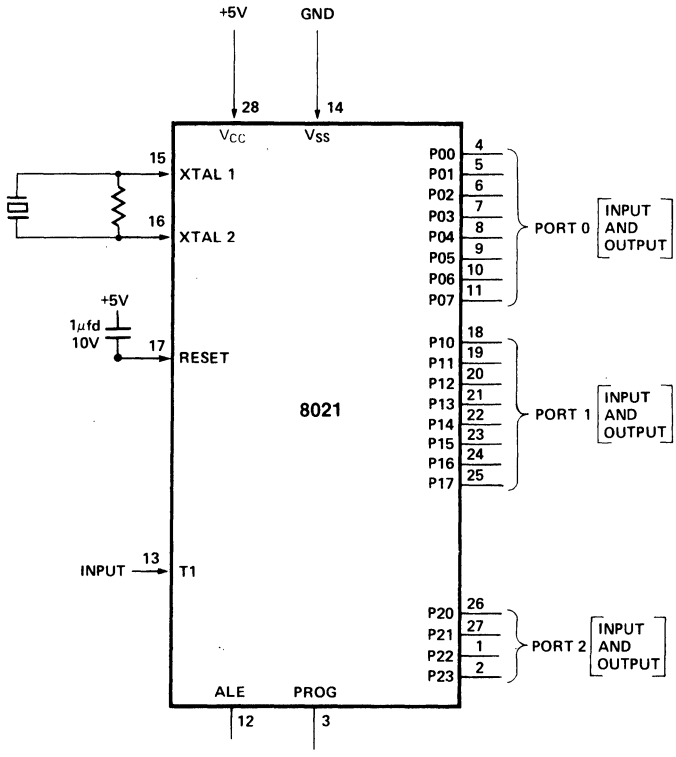
NOTE: SEE PAGE 5-3 FOR 8021 FREQUENCY REFERENCES.

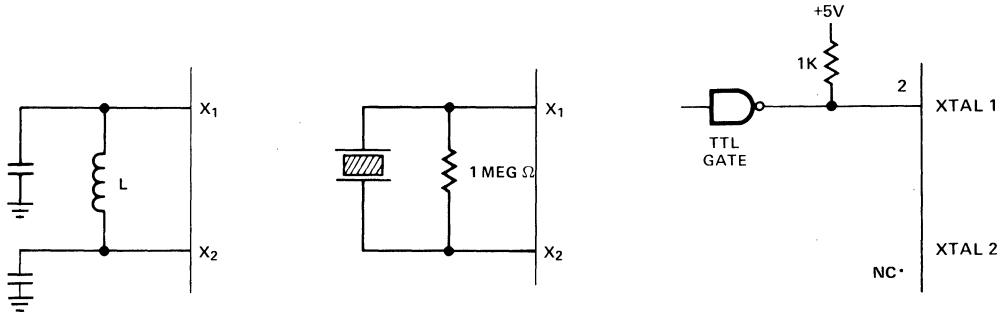
FREQUENCY REFERENCE OPTIONS



- **All inputs and outputs (except $\overline{\text{RESET}}$, X1, X2) standard TTL compatible**
- **P1 and P2 outputs drive 5V CMOS directly others require 10 to 50KΩ pullup for CMOS compatibility**

XTAL:	**Series Resonant**	**CTS Knights MP060**
	1 to 6 MHz	**Crystek CY6B**
	or	**or equivalent**
	Parallel Resonant	**or standard 3.58 MHz**
	for higher	**TV Color Burst XTAL**
	accuracy	

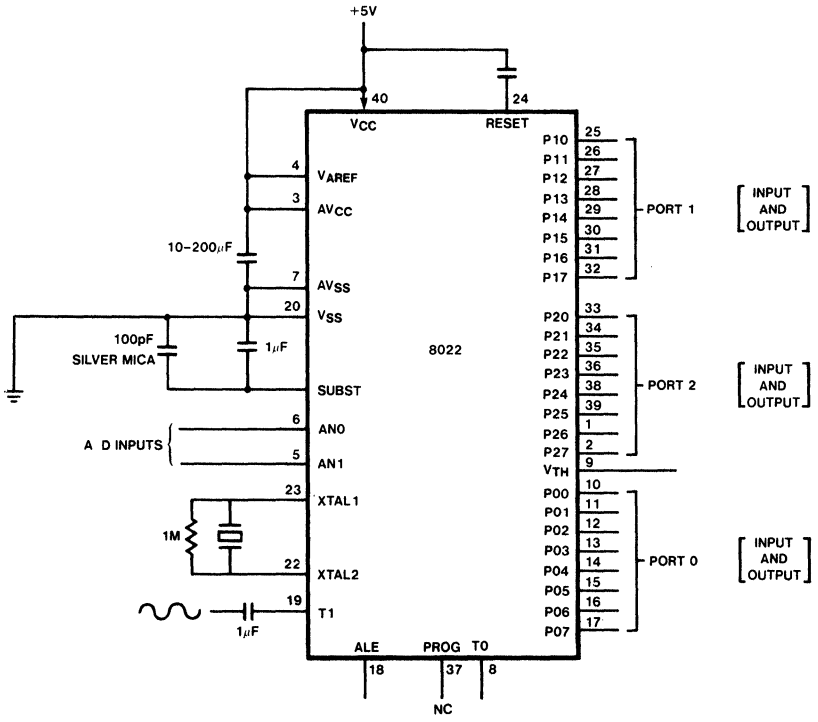


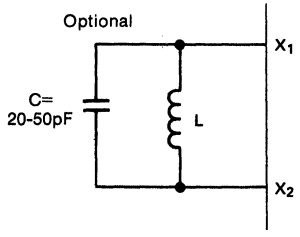


LC **CRYSTAL** **EXTERNAL**

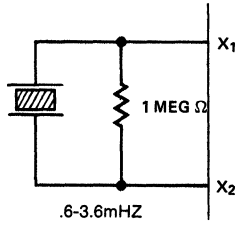
ALTERNATE FREQUENCY REFERENCE OPTIONS
(COMPONENT VALUES TO BE DETERMINED)

THE STAND ALONE 8021

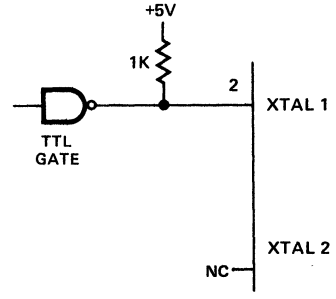




L



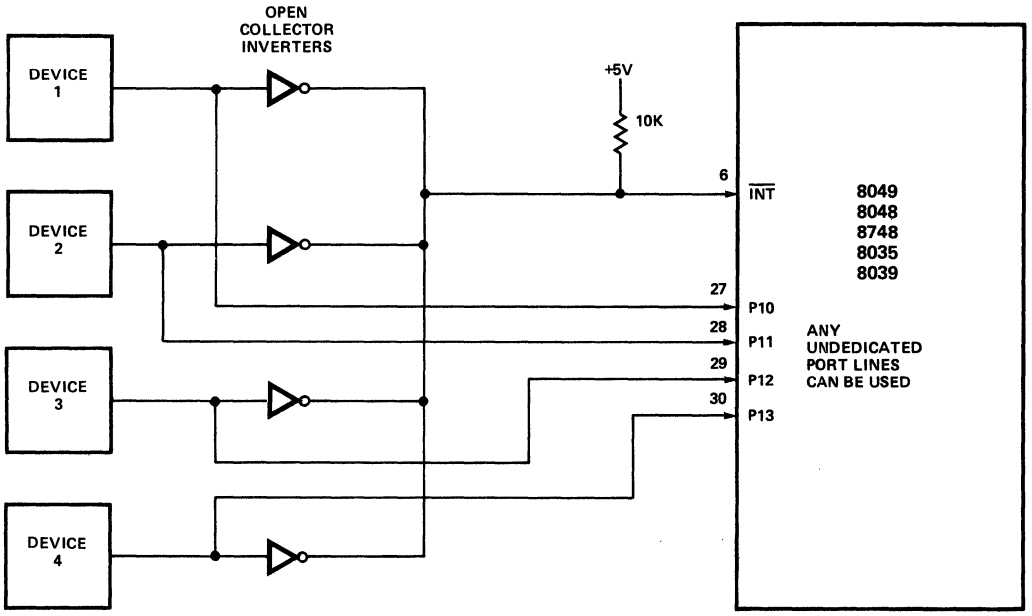
CRYSTAL



EXTERNAL

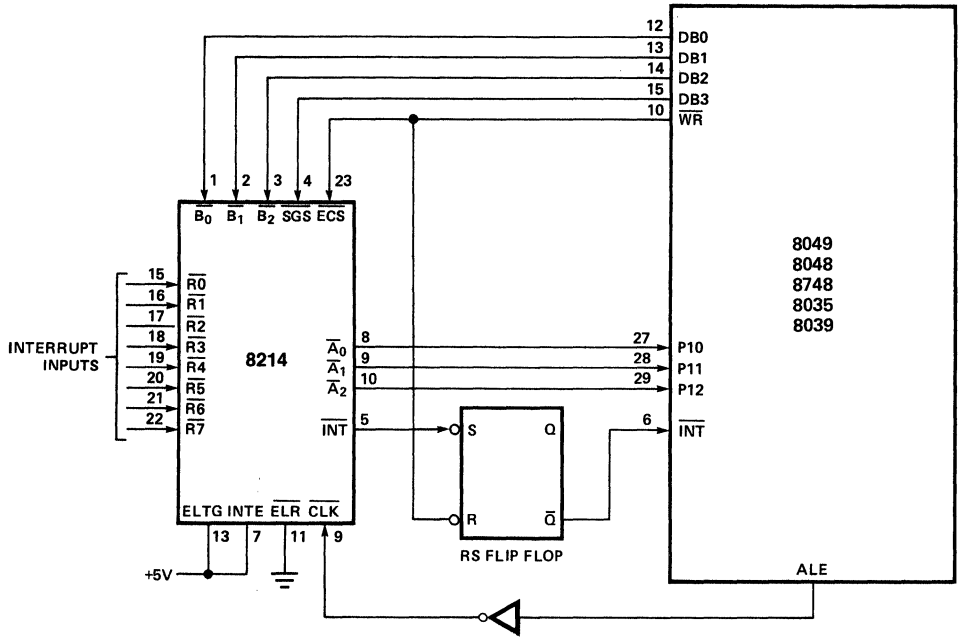
ALTERNATE FREQUENCY REFERENCE OPTIONS
(COMPONENT VALUES TO BE DETERMINED)

THE STAND ALONE 8022



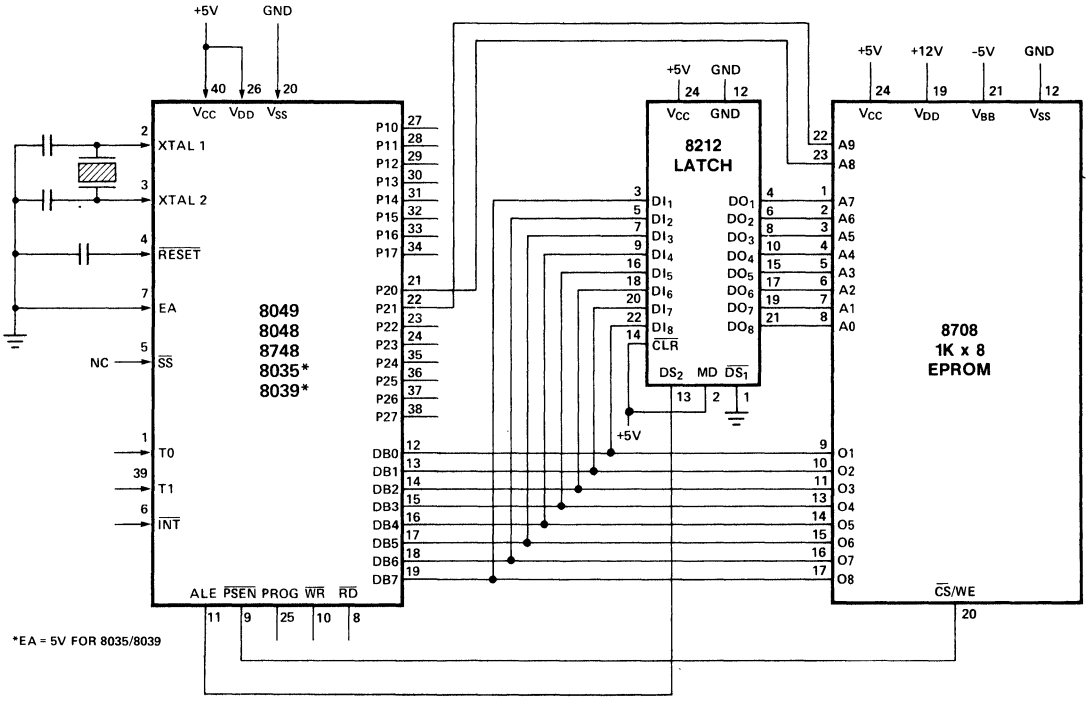
- **All devices equal priority**
- **Processor polls Port 1 to determine interrupting device**

MULTIPLE INTERRUPT SOURCES



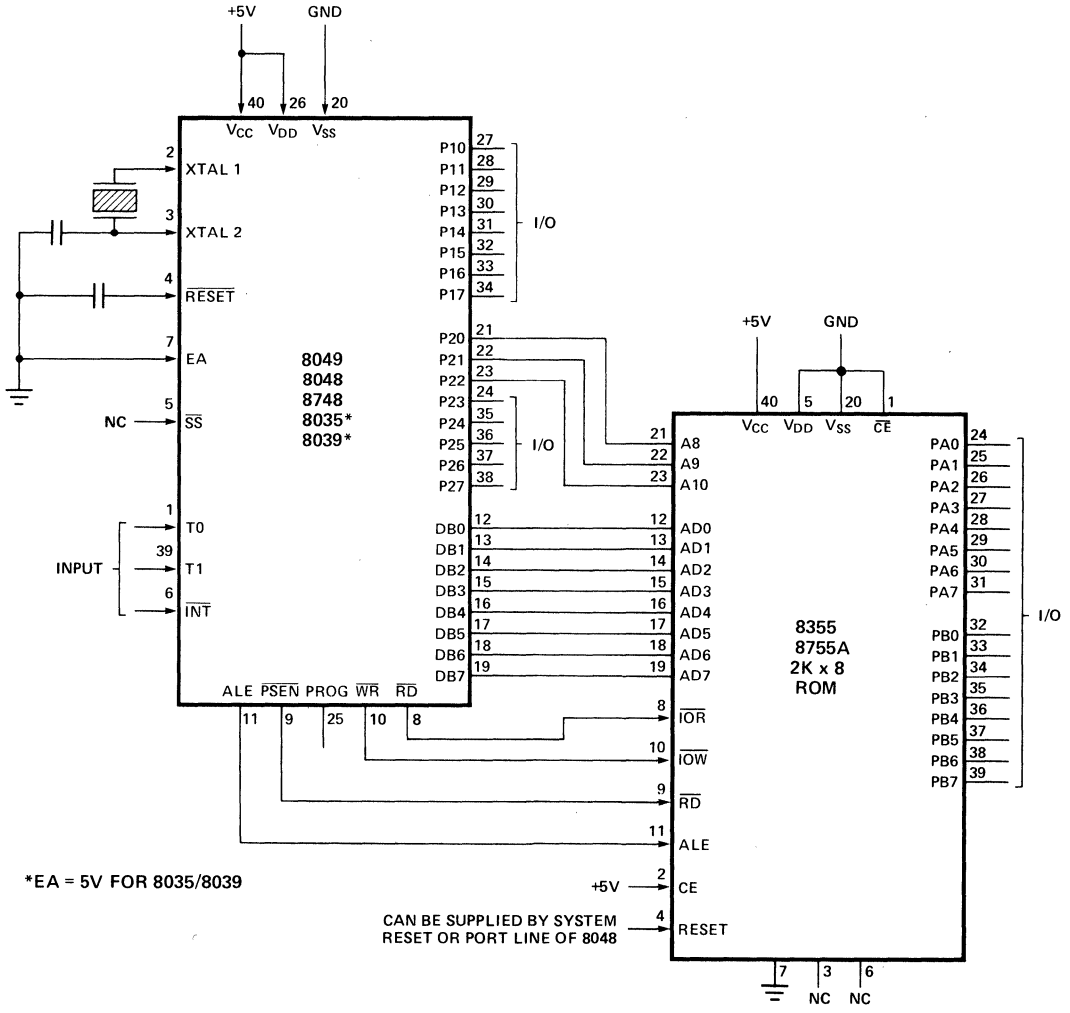
- **Processor polls Port 1 to determine interrupting device**
- **Processor sets priority level by writing 4-bits to 8214**

MULTIPLE INTERRUPTS WITH PRIORITY LEVELS



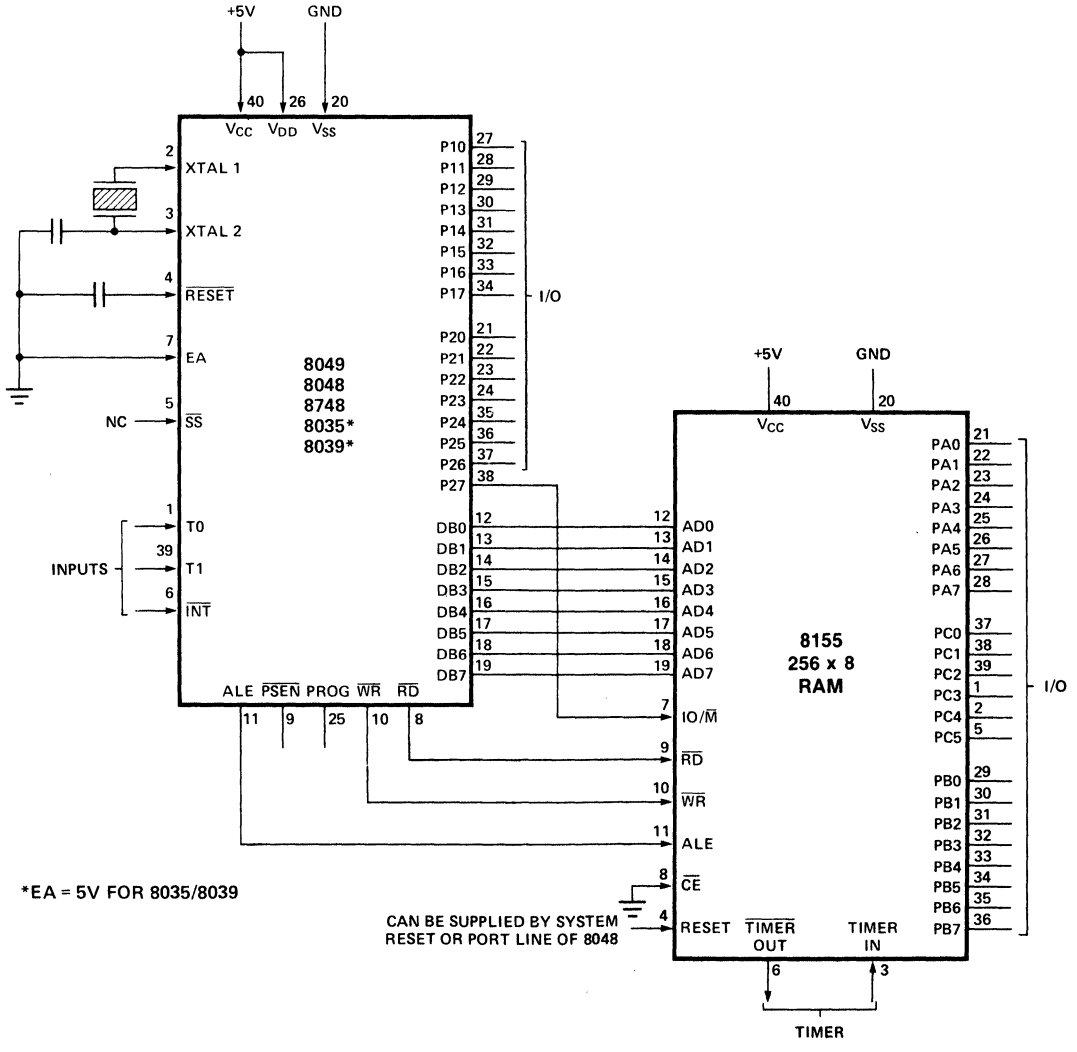
- 8212 serves as address latch
- Address is valid while ALE is high and is latched when ALE goes low

EXTERNAL PROGRAM MEMORY



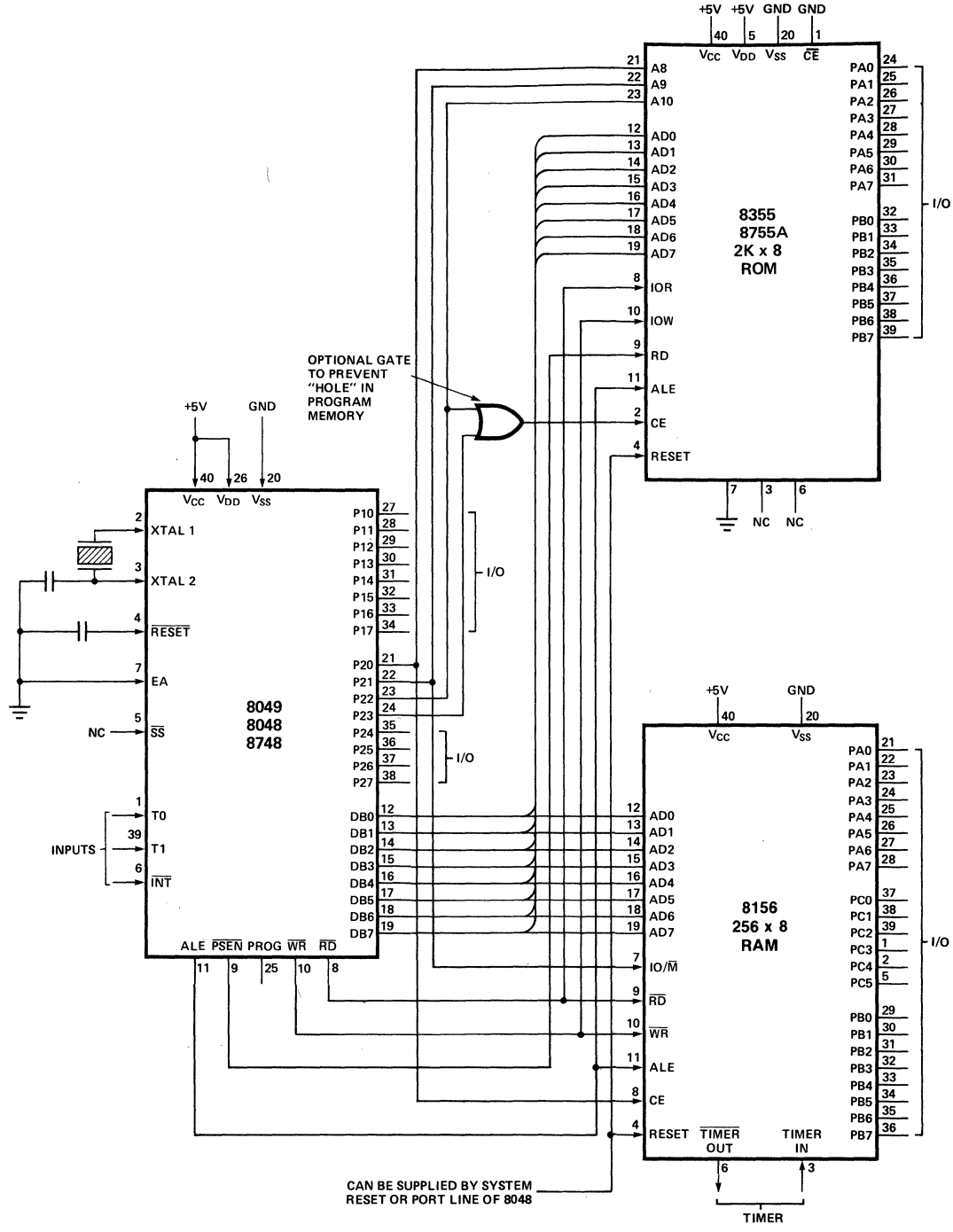
- **External I/O ports are addressed as data memory PA=00 PB=01**
- **If the 8048's internal Program Memory is used this configuration will result in the upper 1K of external memory being addressed before the lower 1K. Inverting A10 will correct this if necessary. This inversion is not necessary if the 8049 is used.**

ADDING A PROGRAM MEMORY AND I/O EXPANDER



- **Both I/O and RAM are addressed as data memory**
- **Writing a bit to P27 determines whether RAM or I/O is to be accessed**

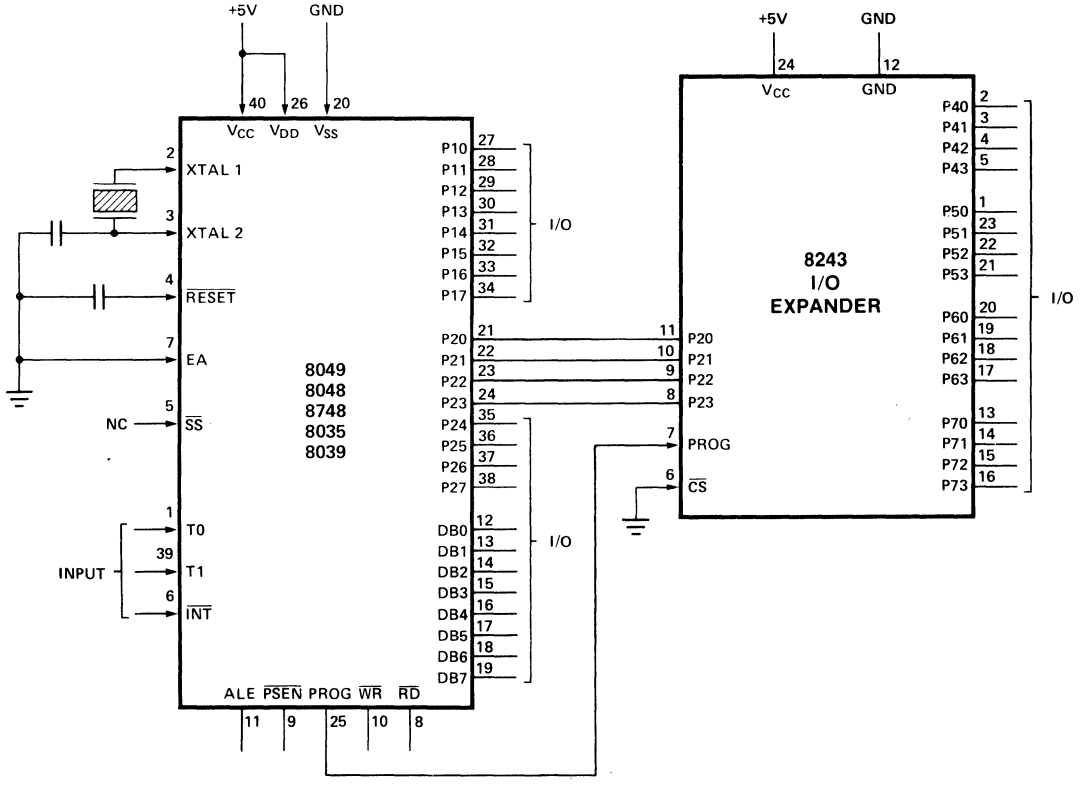
ADDING A DATA MEMORY AND I/O EXPANDER



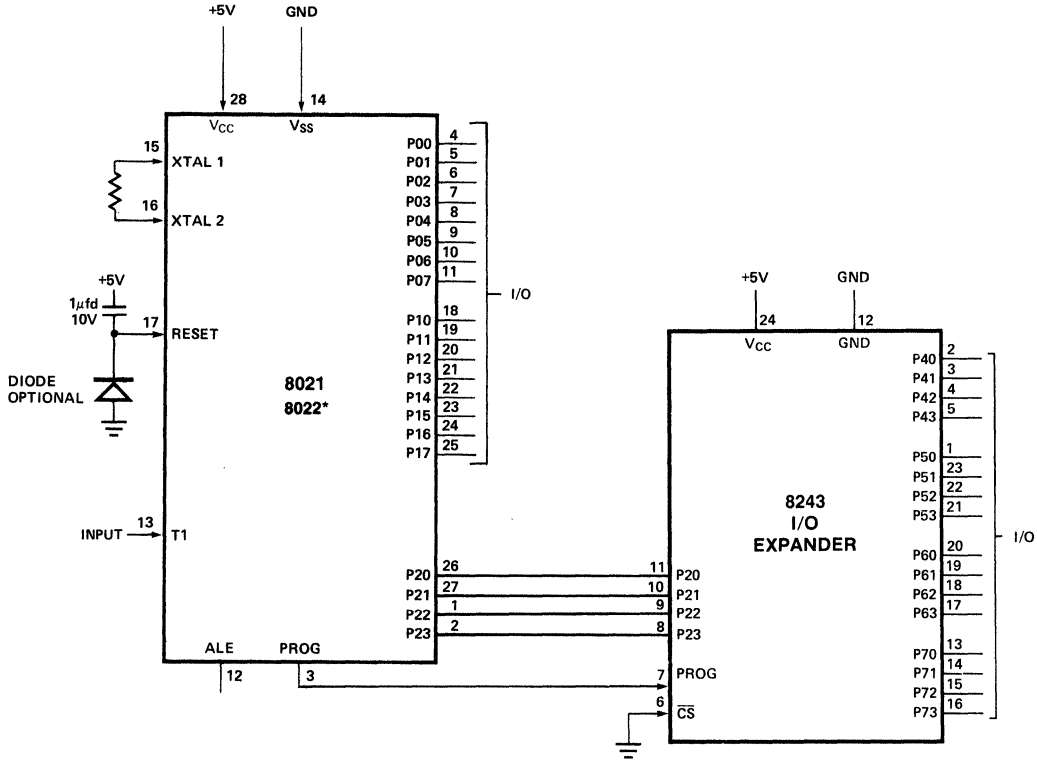
- **This configuration is explained in section 3.4**

I/O Expansion Techniques

The following are several examples of how the basic I/O capability of the MCS-48™ microcomputers can be easily expanded externally using either the 8243 I/O expander device or standard logic circuits. These techniques can be used whenever the combination memory/IO expanders illustrated on the preceeding pages are not required.

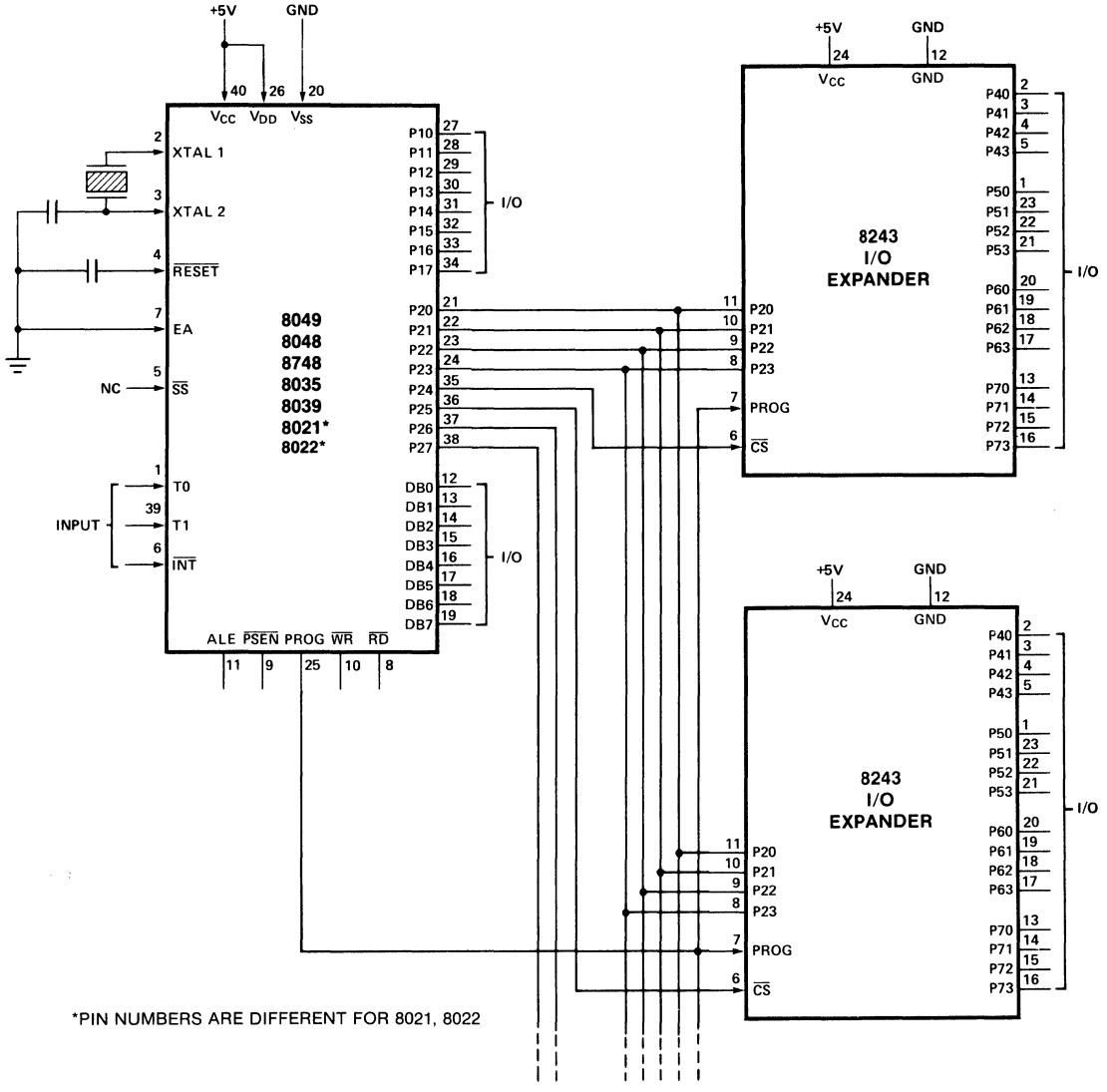


ADDING AN I/O EXPANDER



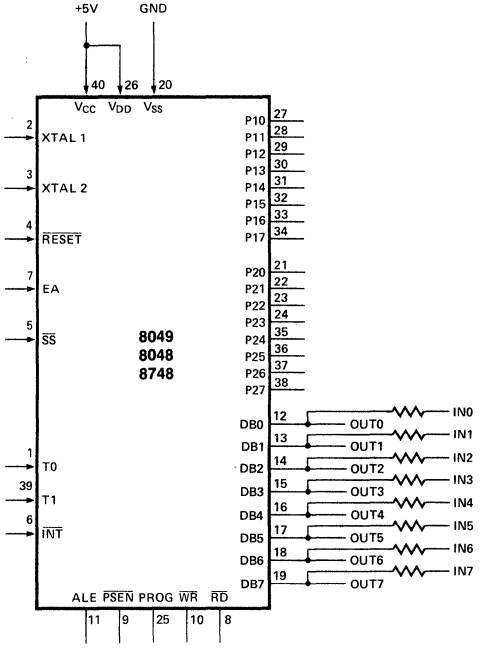
*PIN NUMBERS ARE DIFFERENT FOR 8022

ADDING AN I/O EXPANDER TO THE 8021, 8022



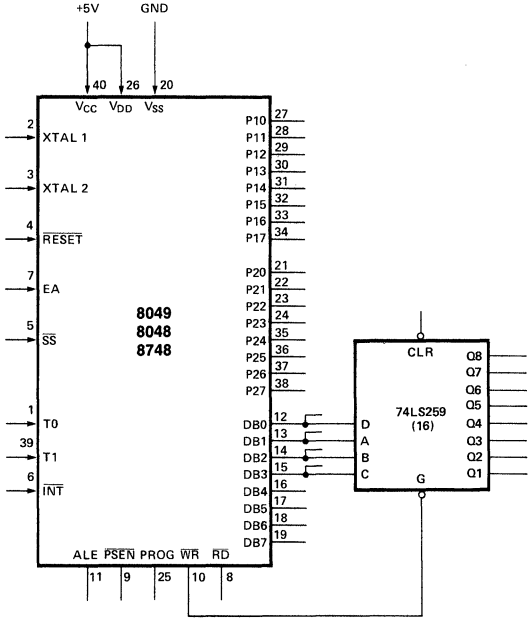
*PIN NUMBERS ARE DIFFERENT FOR 8021, 8022

ADDING MULTIPLE I/O EXPANDERS



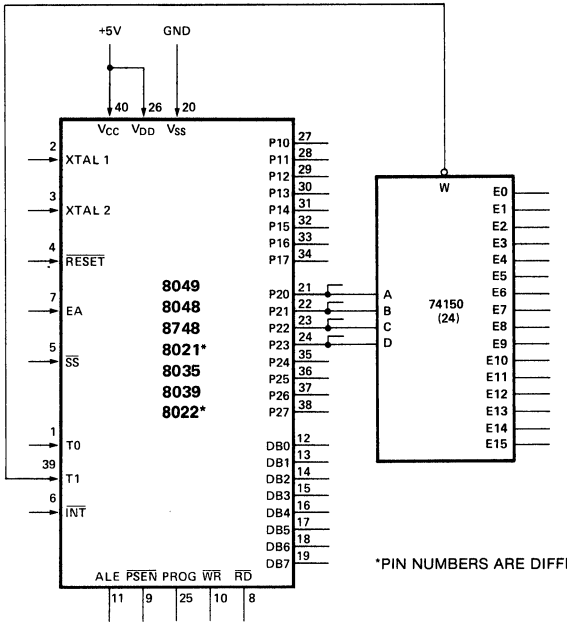
The bus is normally used as an output port. To use it as an input port the bus is put in the high impedance state using the MOVX instruction and then read using the INS instruction. The resistor value chosen is a function of the output loading and the characteristics of the input signals.

ADDING 8 INPUT LINES



Individual bits of the 74LS259 eight-bit addressable latch can be set or reset using the OUTL instruction. During the OUTL operation bit zero of the accumulator is written into the bit of the latch specified by bits 1 through 3 of the accumulator. In this configuration DB_0-DB_7 will be momentarily disturbed while the external latch is loaded.

ADDING 8 OUTPUT LINES

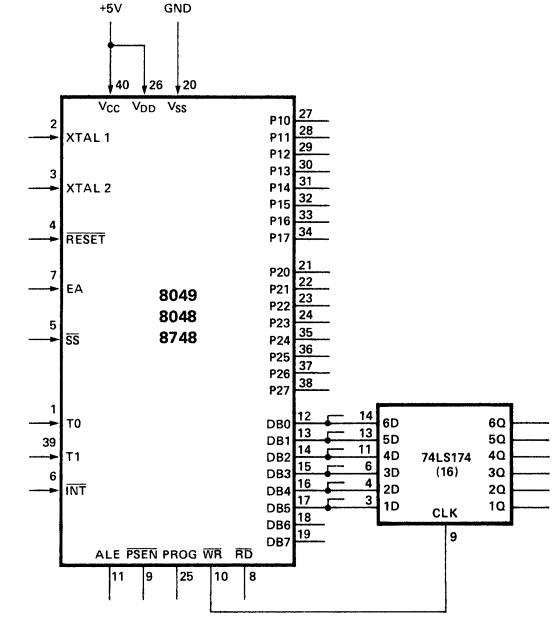


*PIN NUMBERS ARE DIFFERENT FOR 8021, 8022

Normal I/O port is used to select an address for the 16-to-1 multiplexer. The output of the multiplexer is brought into a test input. Eight inputs could be added with a 74LS151 8-to-1 multiplexer using the same structure.

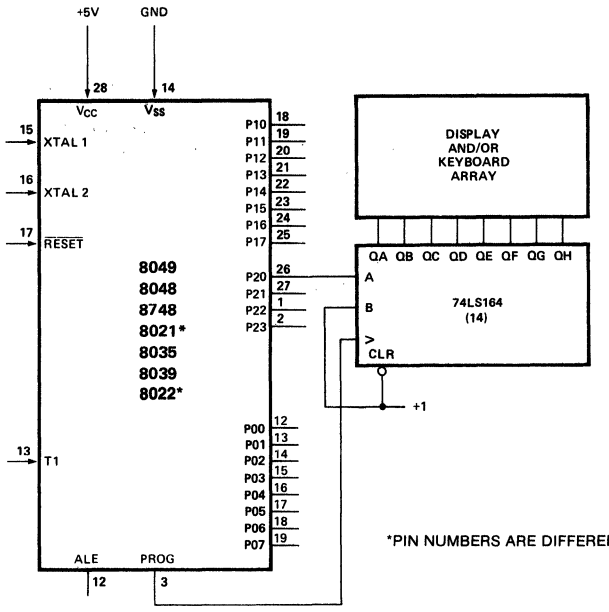
Note: If external program memory is being addressed, use P24-P27 instead of P20-P23.

ADDING 16 INPUT LINES



The latch can be loaded with the OUTL instruction. After the latch is loaded the BUS output state can be modified with the ANL BUS, # DATA and the ORL BUS, # DATA. The OUTL generates a WR strobe; ANL and ORL do not. In this configuration DB_0-DB_7 will be momentarily disturbed while the external latch is loaded.

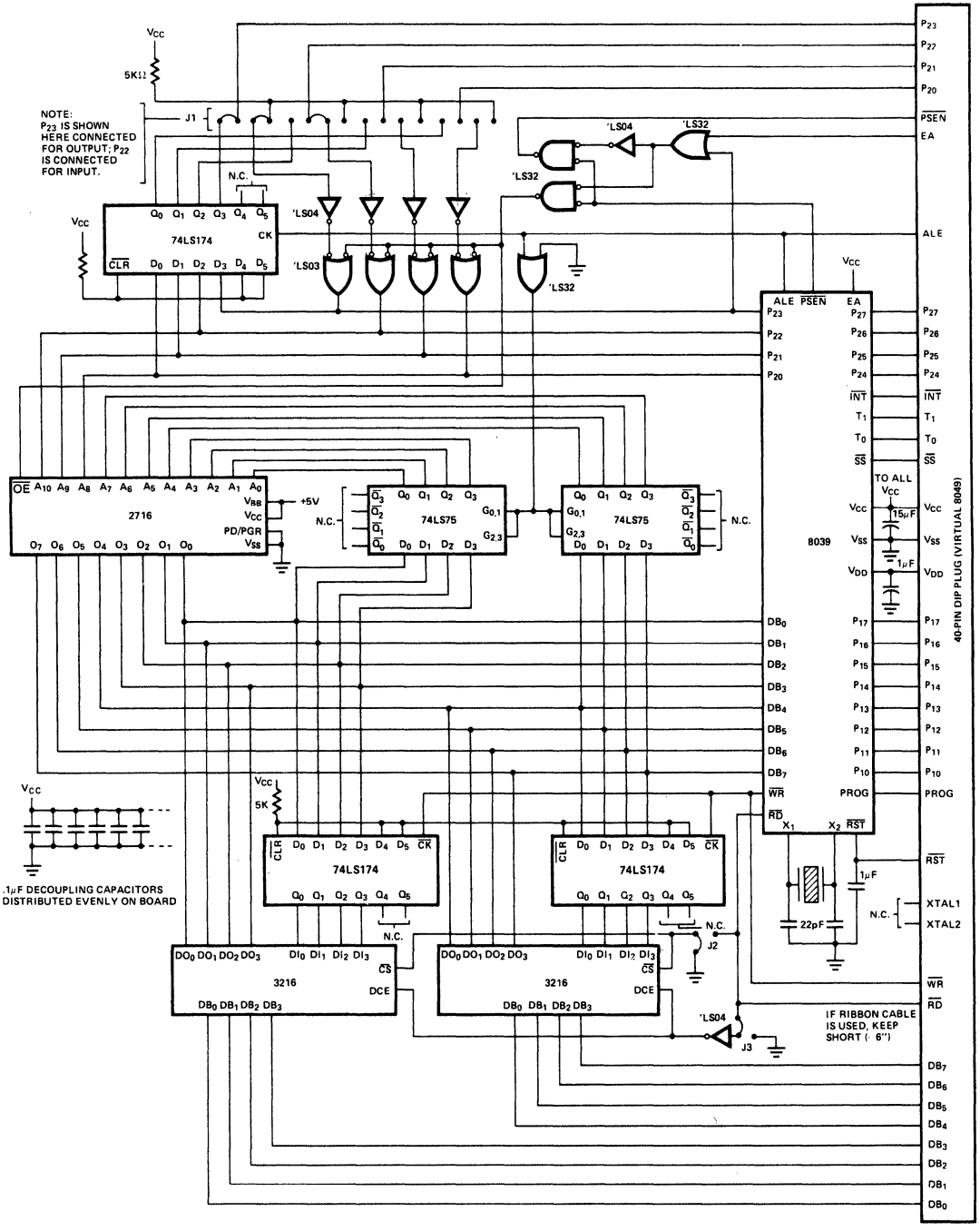
ADDING 6 OUTPUT LINES



*PIN NUMBERS ARE DIFFERENT FOR 8021, 8022

This is most useful when the outputs of the 74LS164 eight-bit shift register will be used to scan a display and/or keyboard. In this case an ANLD P7, A with A = 0FFH can be used to load the initial "1" and an ANLD P6, A with A = 0FEH can be used to move it down the shift register.

ADDING OUTPUT FOR KEYBOARD/DISPLAY SCANNING



8049 EMULATOR CIRCUIT

8049 EMULATOR CIRCUIT DESCRIPTION-6 MHZ

The following is an explanation of a circuit which emulates the operation of an Intel® 8049 using a standard EPROM for program storage.

With the 8049, software may be developed by running external program memory, but doing so requires the use of the bus and P_{23}-P_{20} to access this memory.

The circuit shown may be used to restore the normal functioning of these twelve I/O pins. The circuit consists of an 8039 CPU, 2716 EPROM, two 8216 bi-directional bus drivers, and eight other 7400 Series Low-Power Schottky TTL packages. The whole assembly can be built on a 2-3/4" x 4" board.

A cable coming off the board can be terminated by a forty-pin plug which may be inserted directly into the CPU socket intended for the 8049 in a system undergoing design or testing. Alternatively, a pattern of forty pins extending below the board can be used to plug the board directly into the system undergoing testing, "piggy-back" fashion. The emulator board may be configured in various ways so that the 40 pin plug is the logical equivalent of an 8049 in every legal operating mode. (In the following explanation of the operation of the circuit, an asterisk appearing before a signal or pin number — as in *$\overline{PSEN}$ — refers to that pin on the "virtual 8049" represented by the forty-pin plug).

Since the CPU is identical with the 8049 in all respects other than its lack of program memory, most of the pins of the 8039 are simply connected directly to the corresponding pins of the forty-pin plug. These include all of Port 1, the high order bits of Port 2, the test pins, etc. Signals which are emulated with additional logic include the rest of Port 2, DB_7-DB_0, *$\overline{PSEN}$, etc. $\overline{RD}$, $\overline{WR}$, ALE, and $\overline{PSEN}$ are obtained from the 8039, but are also used by the emulation circuitry.

The EA input of the 8039 is hard-wired high so all instruction fetches are made from the 2716. Two 74LS75 four-bit latches gated by the buffered ALE signal are used to hold the lower eight bits of address from the time-multiplexed data bus. Since the Bus is being used for fetching instructions, data latched to the Bus will be lost on the next instruction fetch. Two 74LS174 latches are used to retain the output data when a bus write is executed. These latches are triggered by the trailing edge of the $\overline{WR}$ pulse, so their outputs are glitch free. Since logical operations to the bus do not generate a WR strobe, the "ANL BUS,#" and "ORL BUS,#" instructions may not be used, though they do function properly with the other ports.

The two 8216 bi-directional bus drivers normally buffer the latched bus contents to the DB pins of the virtual 8049. When an "INS A,BUS" instruction is executed, they buffer the input signals on to the emulator data bus. Thus, the circuit is designed to use the Bus for both latched output and strobed input. If $DB_7$9DB_0 of the 8049 are to be used solely for input data, J2 and J3 may be changed from what is shown in the Figure, so that DB_7-DB_0 act as high impedance inputs and the 8216s are enabled only when the read operation is performed. If the bus is to be used only for latched output, the 8216s can be omitted entirely.

Bi-directional data transfers which require the transfer of address information as well as data, such as to and from external data memory, require removal of the 8216s and replacement with 16-pin jumper blocks on which the DB_x pins are connected with the respective DO_x pins.

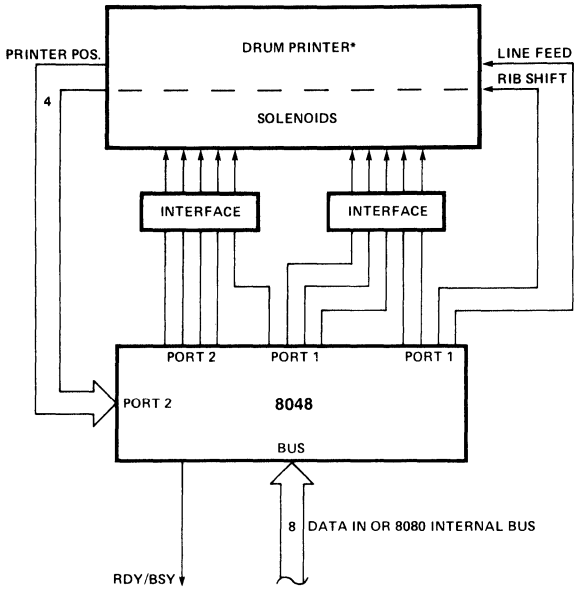
The lower four bits of Port 2 are also used in fetching instructions from the 2716, in addition to their use as input or output pins in the user's system. In configuring the emulator for a particular application, the user must dedicate each of these as either an input or output pin and connect jumper set J1 accordingly. Any mix of input and output pins is allowed. At the beginning of each instruction fetch, the last data written to P2 will be present on P_{23}-P_{20} at the rising edge of ALE and will be latched by a 74LS174. The latched data may be connected through the jumpers to those pins which will be used as outputs on the 8049. Emulator pins used as inputs should be pulled above 2.0V for a logic "one". If this is not the case, i.e., if switches to Ground are to be read, 50K pullup resistors should be added to the circuit on each input. They were omitted from the diagram to minimize input loading.

Pins which will be used as inputs may be connected to the input of an OR gate formed of inverters and open-collector NAND gates. The input signals will be relayed directly to the 8039 and will be read by an "IN A,P2" instruction. But when $\overline{PSEN}$ is low, the NAND outputs are forced off, allowing the 8039 pins to be used for high-order program adressing. Open-collector outputs are needed to prevent line contention when $\overline{PSEN}$ is not low.

If 8243s will be be used in the final system, the low order pins of Port 2 must be connected directly to the plug. This may be done by replacing the Port 2 latch with four jumpers connecting the inputs to the outputs. The NANDs should be removed or disabled by grounding the common NAND inputs.

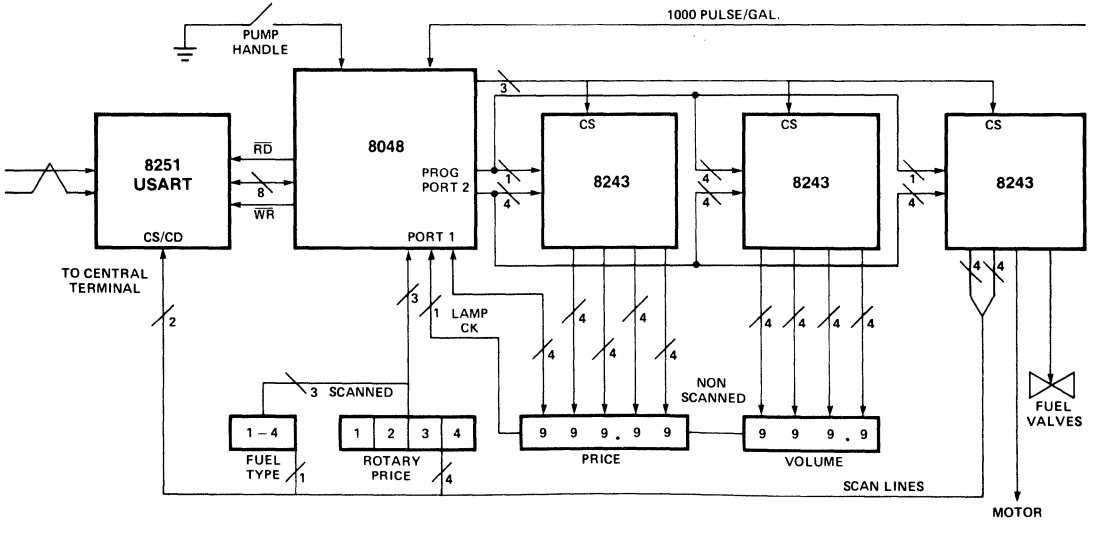
The cluster of three OR gates is used to enable the on-board 2716 and generate the *$\overline{PSEN}$ signal, each of which is a function of $\overline{PSEN}$, *EA, and the high order bit of the program counter. Thus *$\overline{PSEN}$ is generated, forcing an off-board read, only when a jump has been made to Memory Bank 1 or when *EA is brought high. If this feature is to be used to address off-board memory, DB_7-DB_0 may not be used for normal I/O. The 8216s and 74LS174 must be replaced with jumper blocks and the open collector NAND gates disabled, as explained above. The same changes are required to operate the board in single step mode.

NOTE: FOR EMULATION AT 11 MHZ:
1. Substitute a 2716-1;
2. Delete 74LS03 package (leave lines open). Elimination of 74LS03 precludes use of P20-P23 as inputs.

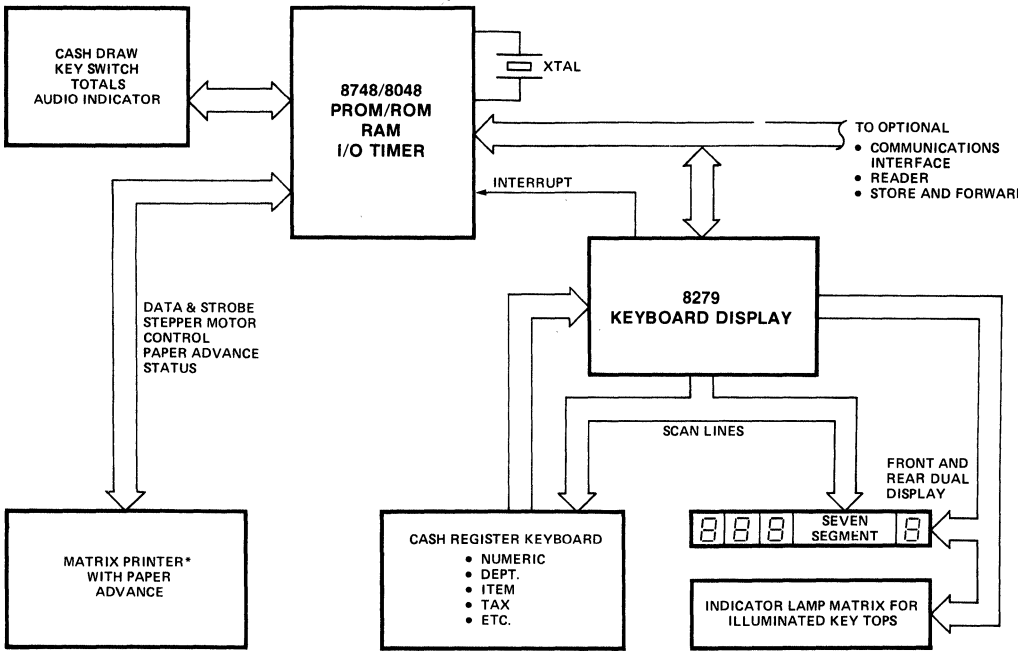


*SEIKO #101

8048 INTERFACE TO DRUM PRINTER

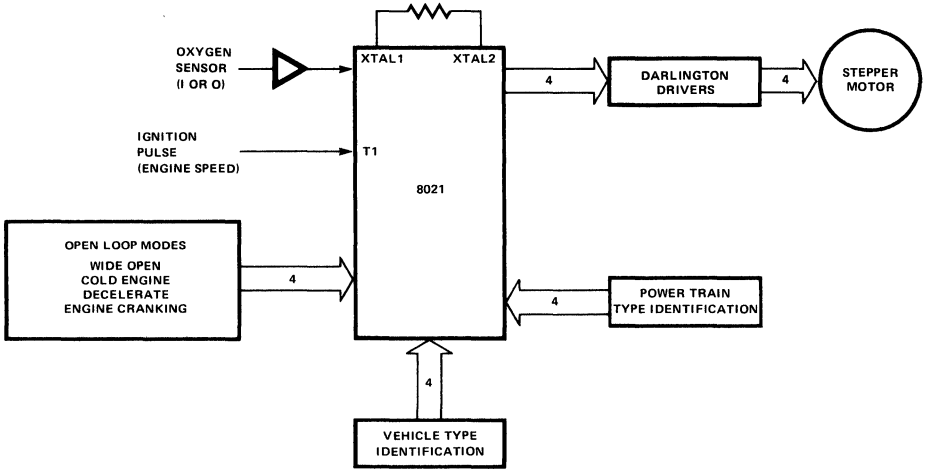


MCS-48™ GAS PUMP

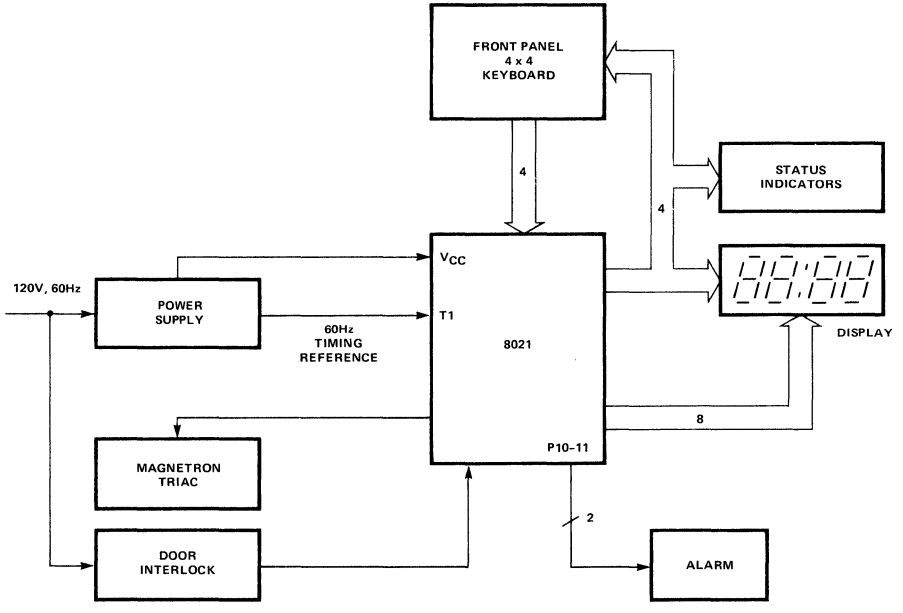


*DRUM PRINTER MAY BE USED. DRUM PRINTER REQUIRES MORE OUTPUTS WHICH CAN BE OBTAINED FROM AN EXPANDER DEVICE.

LOW COST POINT OF SALE TERMINAL



SIMPLE FEEDBACK CARBURETOR CONTROLLER



MICROWAVE OVEN CONTROLLER

5.2 Software Examples

The following routines are written as subroutines. R0 and R1 are used as data pointers, R2 is used as an extension of the accumulator and R3 is used as a loop counter.

```
                RX0 = R0
                AEX = R2
```

DOUBLE ADD

```
DADD:   DEC     RX0         ;GET LOW BYTE AND ADD TO A
        ADD     A,@RX0
        INC     RX0         ;GET HI BYTE AND ADD TO AEX
        XCH     A,AEX
        ADDC    A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE SUBTRACT

```
DMIN:   DEC     RX0         ;GET LOW BYTE AND SUB FROM A
        CPL     A
        ADD     A,@RX0
        CPL     A
        INC     RX0         ;GET HI BYTE AND SUB FROM AEX
        XCH     A,AEX
        CPL     A
        ADDC    A,@RX0
        CPL     A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE LOAD

```
DLD:    DEC     RX0         ;GET LOW BYTE AND PLACE IN A
        MOV     A,@RX0
        INC     RX0         ;GET HI BYTE AND PLACE IN AEX
        XCH     A,AEX
        MOV     A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE STORE

```
DST:    DEC     RX0         ;MOVE A INTO LOW BYTE
        MOV     @RX0,A
        INC     RX0         ;MOVE AEX INTO HIGH BYTE
        XCH     A,AEX
        MOV     @RX0,A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE EXCHANGE

```
DEX:    DEC     RX0         ;EXCHANGE A AND LOW BYTE
        XCH     A,@RX0
        INC     RX0         ;EXCHANGE AEX AND HIGH BYTE
        XCH     A,AEX
        XCH     A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE LEFT LOGICAL SHIFT

```
LLSH:   RLC     A           ;SHIFT A
        XCH     A,AEX       ;SHIFT AEX
        RLC     A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE RIGHT LOGICAL SHIFT

```
RLSH:   XCH     A,AEX       ;SHIFT AEX
        RRC     A
        XCH     A,AEX
        RRC     A           ;SHIFT A
        RET                 ;RETURN
```

DOUBLE RIGHT ARITHMETIC SHIFT

```
RASH:   CLR     C           ;SET CARRY
        CPL     C

        XCH     A,AEX       ;IF AEX[7]<>1 THEN
        JB7     $+3
        CLR     C           ;CLEAR CARRY
        RRC     A           ;SHIFT C INTO AEX
        XCH     A,AEX
        RRC     A           ;SHIFT A
        RET                 ;RETURN
```

SINGLE PRECISION BINARY MULTIPLY

This routine assumes a one-byte multiplier and a one-byte multiplicand. The product, therefore, is two-bytes long.

The algorithm follows these steps:

1. The registers are arranged as follows:

   ACC — 0
   R1 — Multiplier
   R2 — Multiplicand
   R3 — Loop Counter (=8)

   The Accumulator and register R1 are treated as a register pair when they are shifted right (see Step 2)

2. The Accumulator and R1 are shifted right one place, thus the LSB of the multiplier goes into the carry.
3. The multiplicand is added to the accumulator if the carry bit is a 'one'. No action if the carry is a 'zero'.
4. Decrement the loop counter and loop (return to Step 2) until it reaches zero.
5. Shift the result right one last time just before exiting the routine

*The result will be found in the Accumulator (MS Byte) and R1 (LS Byte).

BINARY MULTIPLY

```
BMPY:   MOV   R3,#08H  ;SET COUNTER TO 8
        CLR   A        ;CLEAR A
        CLR   C        ;CLEAR CARRY BIT
BMPI:   RRC   A        ;DOUBLE SHIFT RIGHT ACC & R1
        XCH   A,R1     ;INTO CARRY
        RRC   A
        XCH   A,R1
        JNC   BMP3     ;IF CARRY=1 ADD, OTHERWISE DON'T
        ADD   A,R2     ;ADD MULTIPLICAND TO ACCUMULATOR
BMP3:   DJNZ  R3,BMPI  ;DECREMENT COUNTER AND LOOP IF 0
        RRC   A        ;DO A FINAL RIGHT SHIFT AT THE
        XCH   A,R1     ;END OF THE ROUTINE
        RRC   A
        XCH   A,R1
```

INTERRUPT HANDLING

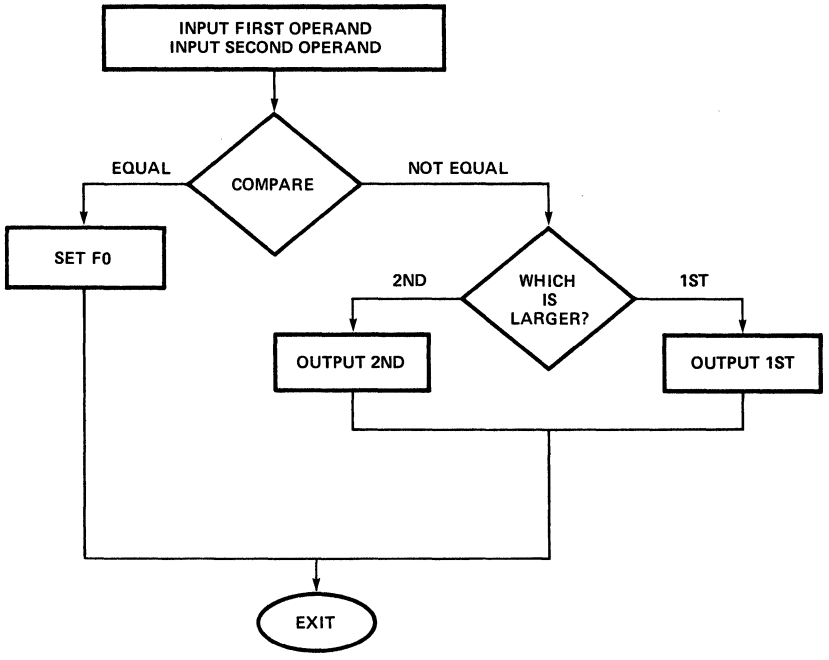
This interrupt routine assumes single level interrupt. The purpose is to store the status of the machine at the time the interrupt occurs by storing contents of all registers, accumulator, and the status word. At the end of the interrupt the state of the machine is restored and interrupts are enabled again.

```
INTRPT: SEL   RB1      ;SAVE WORKING REGISTERS
        MOV   @R0,A    ;R0 IN ALTERNATE REGISTER
                       ;BANK CONTAINS SACC
                       ;POINTER FOR SAVING
                       ;ACCUMULATOR
          |     |      {INTERRUPT SERVICE
          |     |      {ROUTINE
        MOV   R0,SACC  ;RESTORE SACC
        MOV   A,@R0    ;RESTORE ACCUMULATOR
        RETR           ;RESTORE WORKING REGISTERS
                       ;RESTORE PSW AND
                       ;RE-ENABLE INTERRUPTS
```

2 BYTE PROCESSING SYSTEM

A suggested model of a processing routine takes two single byte inputs from different ports, compares them, and performs the following, depending on the result of the comparison:

(If Equal) Sets Flag and Exits
(If Not Equal) and Outputs the Larger to a Third Port




```
PROCESS: CLR     F0       ;CLEAR F0 BIT (INITIALIZE)
         IN      A,P1     ;READ FIRST INPUT, STORE IN R0
         MOV     R0,A
         IN      A,P2     ;READ SECOND INPUT, STORE IN R1
         MOV     R1,A
         CPL     A        ;SUBTRACT SECOND FROM FIRST
         INC     A        ;(2's COMPLEMENT AND ADD)
         ADD     A,R0
         JZ      EQUL     ;BRANCH IF THEY ARE EQUAL
         JNC     SECOND   ;IF NEGATIVE, SECOND WAS LARGER
         MOV     A,R0     ;ELSE, OUTPUT FIRST
         OUTL    BUS,A
         JMP     DONE     ;EXIT

SECOND:  MOV     A,R1     ;OUTPUT SECOND
         OUTL    BUS,A
         JMP     DONE     ;EXIT
EQUL:    CPL     F0       ;SET F0
         JMP     DONE     ;EXIT
```

8 x 8 MULTIPLY–ASSEMBLED BY MCS-48 MACRO ASSEMBLER SEE AP-49

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0

  LOC  OBJ          LINE         SOURCE STATEMENT

                     = 111 $INCLUDE(:F1:MPY8)
                    1= 112 $INCLUDE(:F1:MPY8.PDL)
                    2= 113 ;*******************************************************************************
                    2= 114 ;*                                                                             *
                    2= 115 ;*       MPY8X8                                                                *
                    2= 116 ;*                                                                             *
                    2= 117 ;*=============================================================================*
                    2= 118 ;*                                                                             *
                    2= 119 ;*       THIS UTILITY PROVIDES AN 8 BY 8 UNSIGNED MULTIPLY                     *
                    2= 120 ;*       AT ENTRY:                                                             *
                    2= 121 ;*         A = LOWER EIGHT BITS OF DESTINATION OPERAND                          *
                    2= 122 ;*         XA= DON'T CARE                                                      *
                    2= 123 ;*         R1= POINTER TO SOURCE OPERAND (MULTIPLIER) IN INTERNAL MEMEORY       *
                    2= 124 ;*                                                                             *
                    2= 125 ;*       AT EXIT:                                                              *
                    2= 126 ;*         A = LOWER EIGHT BITS OF RESULT                                      *
                    2= 127 ;*         XA= UPPER EIGHT BITS OF RESULT                                      *
                    2= 128 ;*         C = SET IF OVERFLOW ELSE CLEARED                                    *
                    2= 129 ;*                                                                             *
                    2= 130 ;*******************************************************************************
                    2= 131 ;
                    2= 132 ;
                    2= 133 ;1 MPY8X8:
                    2= 134 ;1 MULTIPLICAND[15-8]:=0
                    2= 135 ;1 COUNT:=8
                    2= 136 ;1 REPEAT
                    2= 137 ;2     IF MULTIPLICAND[0]=0 THEN BEGIN
                    2= 138 ;3        MULTIPLICAND:=MULTIPLICAND/2
                    2= 139 ;2     ELSE
                    2= 140 ;3        MULTIPLICAND[15-8]:=MULTIPLICAND[15-8]+MULTIPLIER
                    2= 141 ;3        MULTIPLICAND:=MULTIPLICAND/2
                    2= 142 ;2     ENDIF
                    2= 143 ;2     COUNT:=COUNT-1
                    2= 144 ;1 UNTIL COUNT=0
                    2= 145 ;1 END MPY8X8
                    1= 146 ;
                    1= 147 ;
                    1= 148 $EJECT
```

8 x 8 MULTIPLY-ASSEMBLED BY MCS-48 MACRO ASSEMBLER SEE AP-49

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0

  LOC  OBJ          LINE        SOURCE STATEMENT

                  1= 149 ;1 MPY8X8:
                  1= 150 MPY8X8:
                  1= 151 ;1 MULTIPLICAND[15-8]:=0
  0035 BA00       1= 152          MOV     XA,#00
                  1= 153 ;1 COUNT:=8
  0037 B808       1= 154          MOV     COUNT,#8
                  1= 155 ;1 REPEAT
                  1= 156 MPY8LP:
                  1= 157 ;2     IF MULTIPLICAND[0]=0 THEN BEGIN
  0039 1243       1= 158          JB0     MPY8A
                  1= 159 ;3       MULTIPLICAND:=MULTIPLICAND/2
  003B 2A         1= 160          XCH     A,XA
  003C 97         1= 161          CLR     C
  003D 67         1= 162          RRC     A
  003E 2A         1= 163          XCH     A,XA
  003F 67         1= 164          RRC     A
  0040 EB39       1= 165          DJNZ    COUNT,MPY8LP
  0042 83         1= 166          RET
                  1= 167 ;2     ELSE
                  1= 168 MPY8A:
                  1= 169 ;3       MULTIPLICAND[15-8]:=MULTIPLICAND[15-8]+MULTIPLIER
  0043 2A         1= 170          XCH     A,XA
  0044 61         1= 171          ADD     A,@R1
  0045 67         1= 172          RRC     A
  0046 2A         1= 173          XCH     A,XA
  0047 67         1= 174          RRC     A
  0048 EB39       1= 175          DJNZ    COUNT,MPY8LP
  004A 83         1= 176          RET
                  1= 177 ;3       MULTIPLICAND:=MULTIPLICAND/2
                  1= 178 ;2     ENDIF
                  1= 179 ;2     COUNT:=COUNT-1
                  1= 180 ;1 UNTIL COUNT=0
                  1= 181 ;1 END MPY8X8
                  1= 182 $EJECT
```

16 x 8 DIVIDE–(ASSEMBLED BY MCS-48 MACRO ASSEMBLER SEE AP-49)

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0

 LOC  OBJ          LINE        SOURCE STATEMENT

                    = 183 $INCLUDE(:F1:DIV16)
                   1= 184 ;*******************************************************************************
                   1= 185 ;*                                                                             *
                   1= 186 ;*       DIV16                                                                 *
                   1= 187 ;*                                                                             *
                   1= 188 ;*******************************************************************************
                   1= 189 ;*                                                                             *
                   1= 190 ;*       THIS UTILITY PROVIDES AN 16 BY 8 UNSIGNED DIVIDE                      *
                   1= 191 ;*       AT ENTRY:                                                             *
                   1= 192 ;*         A = LOWER EIGHT BITS OF DESTINATION OPERAND                          *
                   1= 193 ;*         XA= UPPER EIGHT BITS OF DIVIDEND                                    *
                   1= 194 ;*         R1= POINTER TO DIVISOR IN INTERNAL MEMORY                           *
                   1= 195 ;*                                                                             *
                   1= 196 ;*       AT EXIT:                                                              *
                   1= 197 ;*         A = LOWER EIGHT BITS OF RESULT                                      *
                   1= 198 ;*         XA= REMAINDER                                                       *
                   1= 199 ;*         C = SET IF OVERFLOW ELSE CLEARED                                    *
                   1= 200 ;*                                                                             *
                   1= 201 ;*******************************************************************************
                   1= 202 ;
                   1= 203 ;
                   1= 204 ;1 DIV16:
 0048 2A           1= 205 DIV16:  XCH     A,XA     ; ROUTINE WORKS MOSTLY WITH BITS 15-8
                   1= 206 ;1 COUNT:=8
 004C B808         1= 207         MOV     COUNT,#8
                   1= 208 ;1 DIVIDEND[15-8]:=DIVIDEND[15-8]-DIVISOR
 004E 37           1= 209         CPL     A
 004F 61           1= 210         ADD     A,@R1
 0050 37           1= 211         CPL     A
                   1= 212 ;1 IF BORROW=0 THEN /* IT FITS*/
 0051 F656         1= 213         JC      DIVIA
                   1= 214 ;2    SET OVERFLOW FLAG
 0053 A7           1= 215         CPL     C
 0054 046F         1= 216         JMP     DIVIB
                   1= 217 ;1 ELSE
                   1= 218 DIVIA:
                   1= 219 ;2    RESTORE DIVIDEND
 0056 61           1= 220         ADD     A,@R1
                   1= 221 ;2    REPEAT
                   1= 222 DIVILP:
                   1= 223 ;4          DIVIDEND:=DIVIDEND*2
                   1= 224 ;4          QUOTIENT:=QUOTIENT*2
 0057 97           1= 225         CLR     C
 0058 2A           1= 226         XCH     A,XA
 0059 F7           1= 227         RLC     A
 005A 2A           1= 228         XCH     A,XA
 005B F7           1= 229         RLC     A
 005C E663         1= 230         JNC     DIVIE
 005E 37           1= 231         CPL     A
 005F 61           1= 232         ADD     A,@R1
 0060 37           1= 233         CPL     A
 0061 0468         1= 234         JMP     DIVIC
                   1= 235 ;4          DIVIDEND[15-8]:=DIVIDEND[15-8]-DIVISOR
 0063 37           1= 236 DIVIE:  CPL     A
 0064 61           1= 237         ADD     A,@R1
```

16 x 8 DIVIDE-(ASSEMBLED BY MCS-48 MACRO ASSEMBLER SEE AP-49)

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V3.0

  LOC  OBJ          LINE         SOURCE STATEMENT

  0065 37         1= 238          CPL      A
                  1= 239 ;            IF BORROW=1 THEN
  0066 E668       1= 240          JNC      DIVIC
                  1= 241 ;                RESTORE DIVIDEND
  0068 61         1= 242          ADD      A,@R1
  0069 046C       1= 243          JMP      DIVID
                  1= 244 ;            ELSE
                  1= 245 DIVIC:
                  1= 246 ;                QUOTIENT[0]:=1
  006B 1A         1= 247          INC      XA
                  1= 248 ;            ENDIF
                  1= 249 ;            COUNT:=COUNT-1
                  1= 250 ;          UNTIL COUNT=0
  006C E657       1= 251 DIVID:   DJNZ     COUNT,DIVILP
                  1= 252 ;          CLEAR OVERFLOW FLAG
  006E 97         1= 253          CLR      C
                  1= 254 ;        ENDIF
                  1= 255 ;      ENDDIVIDE
  006F 2A         1= 256 DIVIB:   XCH      A,XA
  0070 83         1= 257          RET
```

MCS-48™ Component Specifications

6

PRELIMINARY

8048H/8048H-1/8035HL/8035HL-1
HMOS SINGLE COMPONENT 8-BIT MICROCOMPUTER

- **8048H/8048H-1 Mask Programmable ROM**
- **8035HL/8035HL-1 CPU Only with Power Down Mode**

- **8-BIT CPU, ROM, RAM, I/O in Single Package**
- **High Performance HMOS**
- **Reduced Power Consumption**
- **1.4 usec and 1.9 usec Cycle Versions All Instructions 1 or 2 Cycles.**
- **Over 90 Instructions: 70% Single Byte**
- **1K x 8 ROM 64 x 8 RAM 27 I/O Lines**
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with 8080/8085 Series Peripherals**
- **Two Single Level Interrupts**

The Intel® 8048H/8048H-1/8035HL/8035HL-1 are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's advanced N-channel silicon gate HMOS process.

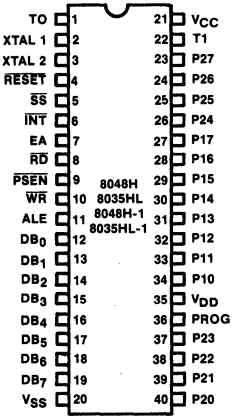
The 8048H contains a 1K X 8 program memory, a 64 X 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability the 8048H can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8035HL is the equivalent of the 8048H without program memory and can be used with external ROM AND RAM.

To reduce development problems to a minimum and provide maximum flexibility, a logically and functionally pin compatible version of the 8048H with UV-erasable user-programmable EPROM program memory is available. The 8748 will emulate the 8048H up to 6 MHz clock frequency with minor differences.

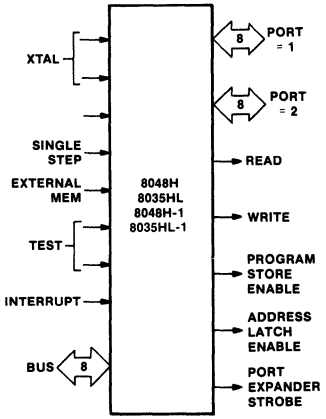
The 8048H is fully compatible with the 8048 when operated at 6 MHz.

These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single bit instructions and no instructions over 2 bytes in length.

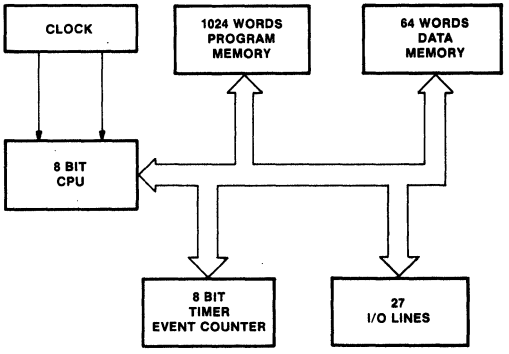
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin =	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Low power standby pin
V_{CC}	40	Main power supply; +5V during operation.
PROG	25	Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-27 Port 2	21-24	8-bit quasi-bidirectional port.
	35-38	P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB0-DB7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched.
		Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
TO	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device.
		Used as a read strobe to external data memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a bus write. (Active low)
		Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output.
		The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

Accumulator

Mnemonic	Description	Bytes	Cycles
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, # data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, # data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, # data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A @R	Or data memory to A	1	1
ORL A, # data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL, A, # data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1

Input/Output

Mnemonic	Description	Bytes	Cycles
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, # data	And immediate to port	2	2
ORL P, # data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, # data	And immediate to BUS	2	2
ORL BUS, # data	Or immediate to BUS	2	2
MOVD A,P	Input expander port to A	1	2
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2

Registers

Mnemonic	Description	Bytes	Cycles
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1

Branch

Mnemonic	Description	Bytes	Cycles
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JTO addr	Jump on TO = 1	2	2
JNTO addr	Jump on TO = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JN1 addr	Jump on INT = 0	2	2
JBb addr	Jump on accumulator bit	2	2

Subroutine

Mnemonic	Description	Bytes	Cycles
CALL addr	Jump to subroutine	2	2
RETR	Return	1	2
RETR	Return and restore status	1	2

Flags

Mnemonic	Description	Bytes	Cycles
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	CLear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1

Data Moves

Mnemonic	Description	Bytes	Cycles
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, #data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @	Move to A from page 3	1	2

Timer/Counter

Mnemonic	Description	Bytes	Cycles
MOV A, T	Read timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNT1	Enable timer/counter interrupt	1	1
DIS TCNT1	Disable timer/counter interrupt	1	1

Control

Mnemonic	Description	Bytes	Cycles
EN 1	Enable external interrupt	1	1
DIS 1	Disable external interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT 0 CLK	Enable clock output on T0	1	1

Mnemonic	Description	Bytes	Cycles
NOP	No operation	1	1

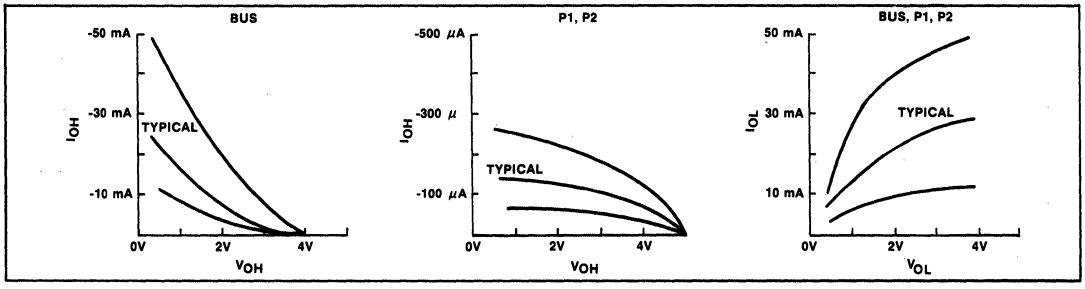
ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -65°C to + 125°C
Voltage On Any Pin With Respect to Ground ..... -0.5V to +7V
Power Dissipation ..... 1.5 Watt

• COMMENT Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. AND OPERATING CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = V_{DD} = 5V ± 10%, V_{SS} = 0V

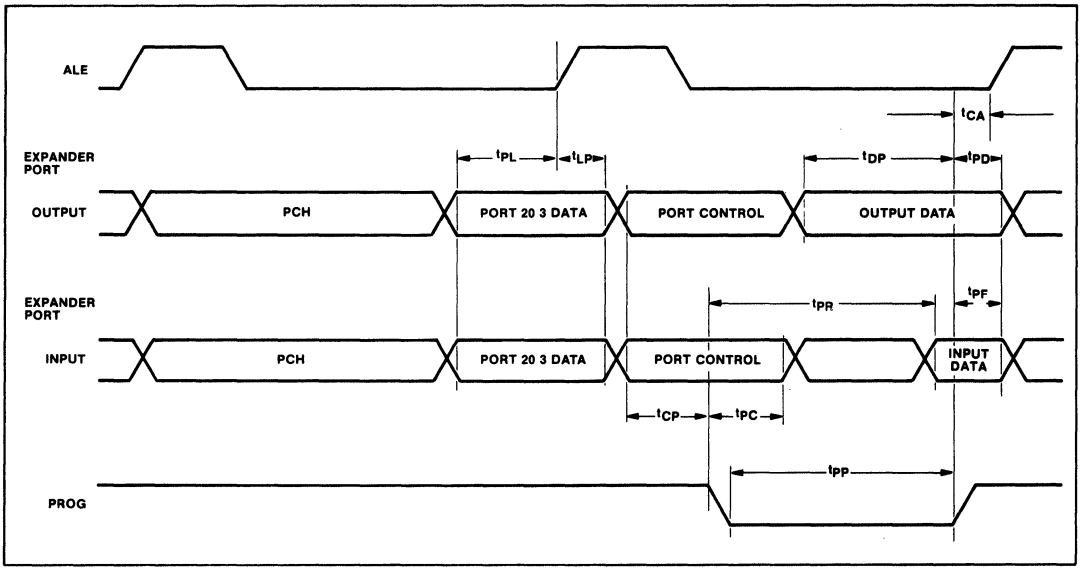
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All Except $\overline{RESET}$, X1, X2)	-.5		.8	V	
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	-.5		.6	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage (X1, X2, $\overline{RESET}$)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS)			.45	V	V_{OL} = 2.0 mA
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			.45	V	I_{OL} = 1.8 mA
V_{OL2}	Output Low Voltage (PROG)			.45	V	I_{OL} = 1.0 mA
V_{OL3}	Output Low Voltage (All Other Outputs)			.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage (BUS)	2.4			V	I_{OH} = -400 μA
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = -100 μA
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = -40 μA
I_{L1}	Input Leakage Current (T1, $\overline{INT}$)			± 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{LI1}	Input Leakage Current (P10-P17, P20-P27, EA, $\overline{SS}$)			-500	μA	V_{SS} + .45 ≤ V_{IN} ≤ V_{CC}
I_{LO}	Output Leakage Current (BUS, TO) (High Impedance State)			± 10	μA	V_{SS} + .45 ≤ V_{IN} ≤ V_{CC}
I_{DD}	V_{DD} Supply Current		4	8	mA	
I_{DD} + I_{CC}	Total Supply Current		40	80	mA	



A.C. CHARACTERISTICS (PORT 2 TIMING) TA = 0°C to 70°C, VCC = 5V ± 10%, VSS = 0V

Symbol	Parameter	8048H 8035HL				8048H-1 8035HL-1		Unit
		6 MHz		8 MHz		11 MHz		
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CP}	Port control Setup Before Falling Edge of PROG.	110		105				ns
t_{PC}	Port Control Hold After Falling Edge of PROG.	100		90				ns
t_{PR}	PROG to Time P2 Input Must Be Valid		810		700		650	ns
t_{PF}	Input Data Hold Time	0	150	0	150	0	150	ns
t_{DP}	Output Data Setup Time	250		210		200		ns
t_{PD}	Output Data Hold Time	65		35		20		ns
t_{PP}	PROG Pulse Width	1200		970		700		ns
t_{PL}	Port 2 I/O Data Setup	350		300		250		ns
t_{LP}	Port 2 I/O Data Hold	150		65		20		ns

PORT 2 TIMING



BUS TIMING AS A FUNCTION OF TCY *

SYMBOL	FUNCTION OF TCY		
T_{LL}	7/30	T_{CY}	MIN
T_{AL}	1/10	T_{CY}	MIN
T_{LA}	1/15	T_{CY}	MIN
T_{CC} (1)	1/2	T_{CY}	MIN
T_{CC} (2)	2/5	T_{CY}	MIN
T_{DW}	2/15	T_{CY}	MIN
T_{WD}	1/15	T_{CY}	MIN
T_{DR}	0		MIN

T_{CC} (1) : $\overline{RD}/\overline{WR}$
T_{CC} (2) : $\overline{PSEN}$

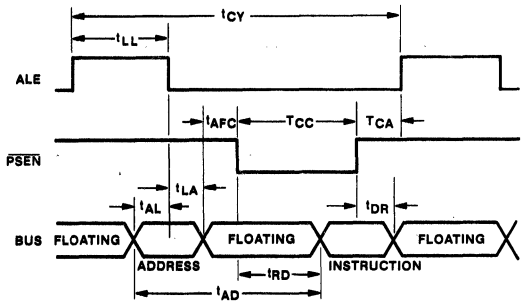
SYMBOL	FUNCTION OF TCY		
T_{RD} (1)	11/30	T_{CY}	MAX
T_{RD} (2)	3/10	T_{CY}	MAX
T_{AW}	3/10	T_{CY}	MIN
T_{AD} (1)	1/2	T_{CY}	MAX
T_{AD} (2)	1/3	T_{CY}	MAX
T_{AFC}	1/30	T_{CY}	MIN
T_{CA}	1/15	T_{CY}	MIN

T_{RD} (1) : $\overline{RD}$
T_{RD} (2) : $\overline{PSEN}$

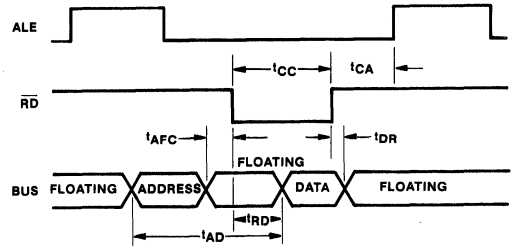
T_{AD} (1) : $\overline{RD}$
T_{AD} (2) : $\overline{PSEN}$

* APPROXIMATE VALUES NOT INCLUDING GATE DELAYS.

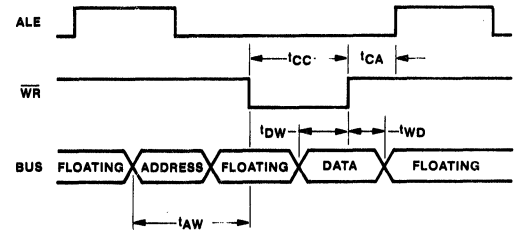
WAVEFORMS



Instruction Fetch From External Program Memory



Read From External Data Memory



Write to External Data Memory

Input and Output for A.C.Tests

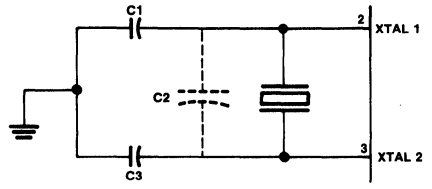
A.C. CHARACTERISTICS TA = 0°C to 70°C VCC = VDD = 5V ± 10%, VSS = 0V

Symbol	Parameter	8048H 8035HL				8048H-1 8035HL-1		Unit	Conditions (Note 1)
		6 MHz		8 MHz		11 MHz			
		Min.	Max.	Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	400		270		150		ns	
t_{AL}	Address Setup to ALE	75		75		70		ns	
t_{LA}	Address Hold from ALE	65		65		50		ns	
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)	700		490		300		ns	
t_{DW}	Data Setup before $\overline{WR}$	370		370		280		ns	
t_{WD}	Data Hold after $\overline{WR}$	80		80		40		ns	CL = 20pF (NOTE 2)
t_{CY}	Cycle Time	2.5		1.875		1.36		μs	
t_{DR}	Data Hold	0	200	0	150	0	100	ns	
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In		500		340		200	ns	
t_{AW}	Address Setup to $\overline{WR}$	230		210		200		ns	
t_{AD}	Address Setup to Data In		950		650		400	ns	
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	0		0		-1		ns	
t_{CA}	Control Pulse to ALE	10		10		0		ns	

NOTE 1: Control outputs CL = 80 pF
BUS outputs CL = 150 pF

NOTE 2: BUS High Impedance Load: 20 pF

CRYSTAL OSCILLATOR MODE

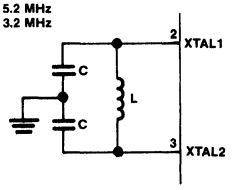


C1 = 5pF ± 1/2pF + STRAY < 5pF
C2 = CRYSTAL + STRAY < 8pF
C3 = 20pF ± 1pF + STRAY < 5pF

CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 75Ω AT 6 MHz LESS THAN 180Ω AT 3.6MHz

LC OSCILLATOR MODE

L	C	NOMINAL 1
45pH	20pF	5.2 MHz
120pH	20pF	3.2 MHz



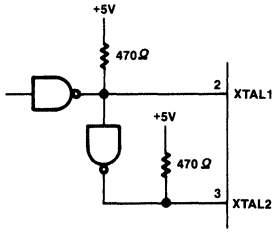
$$F = \frac{1}{2\pi\sqrt{LC}}$$

$$C = \frac{C + Cpp}{2}$$

Cpp = 5-10 pF PIN TO PIN CAPACITANCE

EACH C SHOULD BE APPROXIAMTELY 20pF. INCLUDING STRAY CAPACITANCE

DRIVING FROM EXTERNAL SOURCE



XTAL 1 MUST BE HIGH 35-65% OF THE PERIOD AND XTAL 2 MUST BE HIGH 35-65% OF THE PERIOD. RISE AND FALL TIMES MUST NOT EXCEED 20ns.

8048L

SPECIAL LOW POWER CONSUMPTION SINGLE COMPONENT 8-BIT MICROCOMPUTER

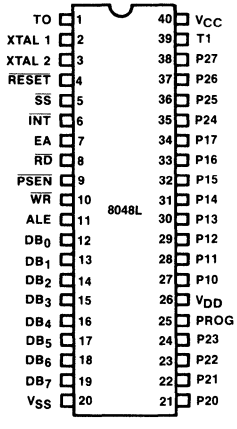
- **Typical Power Consumption 100mW**
- **Typical Standby Power 10mW**
  V_{DD} minimum of 2.2V
- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **4.17 μsec Instruction Cycle.**
  All Instructions 1 or 2 Cycles.
- **Over 90 Instructions: 70% Single Byte**
- **1K x 8 ROM**
  64 x 8 RAM
  27 I/O Lines
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with 8080/8085 Series Peripherals**
- **Two Single Level Interrupts**

The Intel® 8048L is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's advanced N-channel silicon gate HMOS process, using special techniques to reduce operating and standby power consumption. The 8048L contains a 1K X 8 program memory, a 64 X 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability the 8048L can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8048L can be used with external ROM AND RAM.

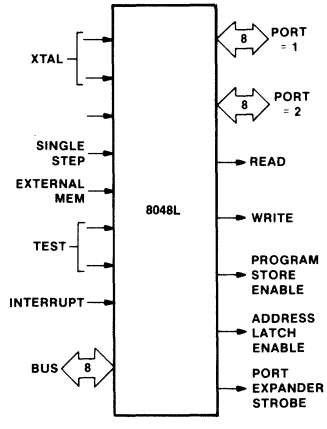
To reduce development problems to a minimum and provide maximum flexibility, a logically and functionally pin compatible version of the 8048L with UV-erasable user-programmable EPROM program memory is available. The 8748 will emulate the 8048L with greater power and other minor differences.

This microcontroller is designed to be an efficient controller as well as an arithmetic processor. The 8048L has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

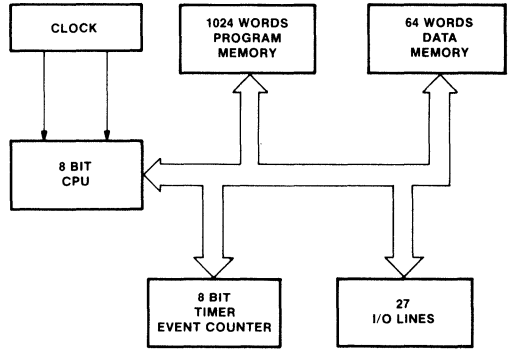
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



8048/8035L/8748/8748-6/8035
SINGLE COMPONENT 8-BIT MICROCOMPUTER

- 8048 Mask Programmable ROM
- 8035L CPU Only with Power Down Mode
- 8748 User Programmable EPROM
- 8748-6 Up to 55C 8748
- 8035 CPU Only (No Power Down Mode)

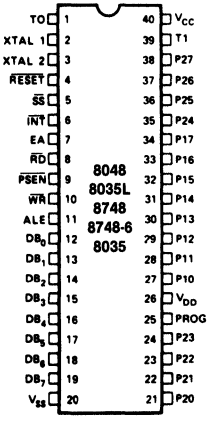
- 8-BIT CPU, ROM, RAM, I/O in Single Package
- Interchangeable ROM and EPROM Versions
- Single 5V Supply
- 2.5 μsec and 5.0 μsec Cycle Versions All Instructions 1 or 2 Cycles.
- Over 90 Instructions: 70% Single Byte
- 1K x 8 ROM/EPROM 64 x 8 RAM 27 I/O Lines
- Interval Timer/Event Counter
- Easily Expandable Memory and I/O
- Compatible with 8080/8085 Series Peripherals
- Single Level Interrupt

The Intel 8048/8748/8743-6/8748-8/8035/8035-8 are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's N-channel silicon gate MOS process.

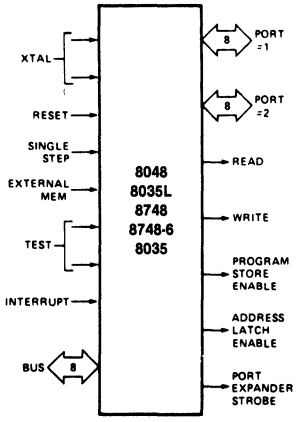
The 8048 contains a 1K x 8 program memory, a 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability, the 8048 can be expanded using standard memories and MCS-80 /MCS-85 peripherals. The 8035 is the equivalent of an 8048 without program memory and can be used with external ROM and RAM. The 8035L has the RAM power-down mode of the 8048 while the 8035 does not. The 8748-6 is a 6 MHz 8748 up to 55C. To reduce development problems to a minimum and provide maximum flexibility, three interchangeable pin-compatible versions of this single component microcomputer exist: the 8748 with user-programmable and erasable EPROM program memory for prototype and preproduction systems, the 8048 with factory-programmed mask ROM program memory for low cost, high volume production, and the 8035 without program memory for use with external program memories.

These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single bit instructions and no instructions over 2 bytes in length.

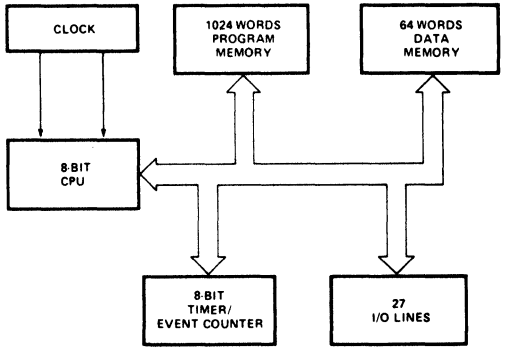
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 and 8035L.
V_{CC}	40	Main power supply; +5V during operation and programming.
PROG	25	Program pulse (+23V) input pin during 8748 programming.
		Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB_0-DB_7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched.
		Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

Designation	Pin #	Function
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device.
		Used as a read strobe to external data memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during PROM programming verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a bus write. (Active low)
		Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output.
		The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input expander port to A	1	2
	MOVD P, A	Output A to expander port	1	2
	ANLD P, A	And A to expander port	1	2
	ORLD P, A	Or A to expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on carry = 1	2	2
	JNC addr	Jump on carry = 0	2	2
	J Z addr	Jump on A zero	2	2
	JNZ addr	Jump on A not zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on accumulator bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL addr	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear carry	1	1
	CPL C	Complement carry	1	1
	CLR F0	Clear flag 0	1	1
	CPL F0	Complement flag 0	1	1
	CLR F1	Clear flag 1	1	1
	CPL F1	Complement flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from page 3	1	2
Timer/Counter	MOV A, T	Read timer/counter	1	1
	MOV T, A	Load timer/counter	1	1
	STRT T	Start timer	1	1
	STRT CNT	Start counter	1	1
	STOP TCNT	Stop timer/counter	1	1
	EN TCNTI	Enable timer/counter interrupt	1	1
	DIS TCNTI	Disable timer/counter interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable clock output on T0	1	1
	NOP	No operation	1	1

ABSOLUTE MAXIMUM RATINGS*

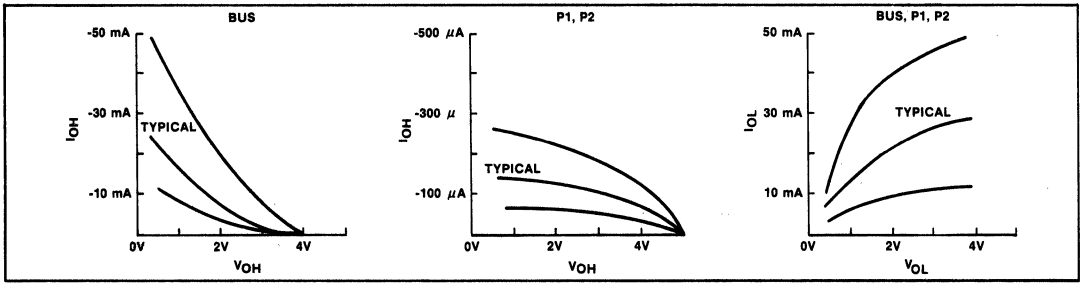
Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +125°C
Voltage On Any Pin With Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1.5 Watt

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

D.C. AND OPERATING CHARACTERISTICS

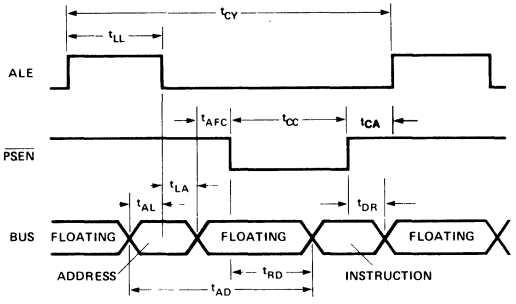
T_A = 0°C to 70°C, V_{CC} = V_{DD} = +5V ± 10%, V_{SS} = 0V
(T_A = 0°C to 55°C for 8748-6)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All Except $\overline{RESET}$, X1, X2)	-.5		.8	V	
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	-.5		.6	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage (X1, X2, $\overline{RESET}$)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS)			.45	V	V_{OL} = 2.0 mA
V_{OL1}	Output Low Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			.45	V	I_{OL} = 1.8 mA
V_{OL2}	Output Low Voltage (PROG)			.45	V	I_{OL} = 1.0 mA
V_{OL3}	Output Low Voltage (All Other Outputs)			.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage (BUS)	2.4			V	I_{OH} = −400 μA
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = −100 μA
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = -40 μA
I_{L1}	Input Leakage Current (T1, $\overline{INT}$)			± 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{LI1}	Input Leakage Current (P10-P17, P20-P27, EA, $\overline{SS}$)			-500	μA	V_{SS} + .45 $\leq V_{IN} \leq V_{CC}$
I_{L0}	Output Leakage Current (BUS, TO) (High Impedance State)			± 10	μA	V_{SS} + .45 $\leq V_{IN} \leq V_{CC}$
I_{DD}	V_{DD} Supply Current		5	15	mA	
I_{DD} + I_{CC}	Total Supply Current		60	135	mA	

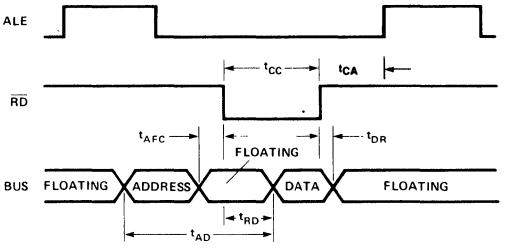


WAVEFORMS

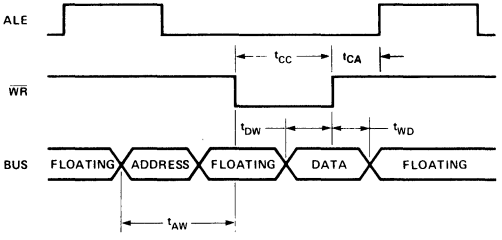
Instruction Fetch From External Program Memory



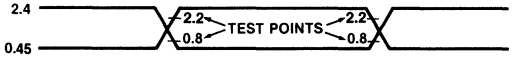
Read From External Data Memory



Write to External Data Memory



Input and Output Waveforms for A.C. Tests



A.C. CHARACTERISTICS T_A = 0°C to 70°C*, V_{CC} = V_{DD} = +5V ± 10%, V_{SS} = 0V

Symbol	Parameter	8048 8748-6 8748/8035/8035L		8748-8** 8035-8		Unit	Conditions (Note 1)
		Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	400		600		ns	
t_{AL}	Address Setup to ALE	120		150		ns	
t_{LA}	Address Hold from ALE	80		80		ns	
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)	700		1500		ns	
t_{DW}	Data Setup before $\overline{WR}$	500		640		ns	
t_{WD}	Data Hold After $\overline{WR}$	120		120		ns	C_L = 20pF
t_{CY}	Cycle Time	2.5	15.0	4.17	15.0	μs	6 MHz XTAL = 2.5 (3.6 MHz XTAL for −8)
t_{DR}	Data Hold	0	200	0	200	ns	
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In		500		750	ns	
t_{AW}	Address Setup to $\overline{WR}$	230		260		ns	
t_{AD}	Address Setup to Data In		950		1450	ns	
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	0		0		ns	
t_{CA}	Control Pulse to ALE	10		20		ns	

Note 1: Control outputs: C_L = 80 pF t_{CY} = 2.5 μs for standard parts
BUS Outputs: C_L = 150 pF = 4.17 μs for −8 parts

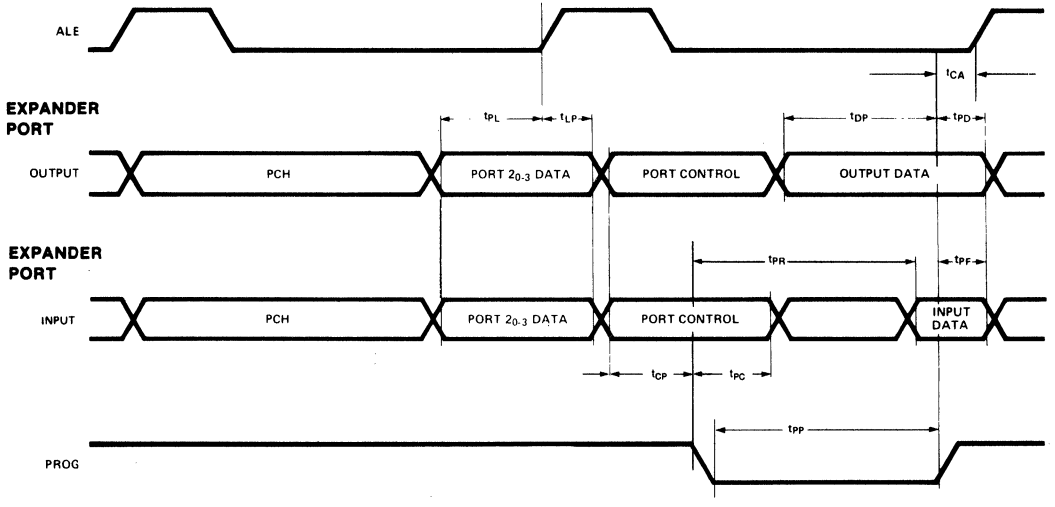
*T_A = 0°C to 55°C for 8748-6
**V_{CC} and V_{DD} for 8748-8 and 8035-8 are ±5%

A.C. CHARACTERISTICS (PORT 2 TIMING)

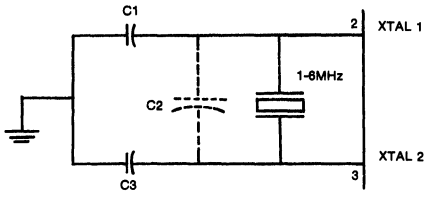
T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, V_{SS} = 0V
(T_A = 0°C to 55°C for 8748-6)
T_A = −40°C to +85°C, V_{CC} = 5V ± 10%, V_{SS} = 0V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of PROG	115		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	65		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		860	ns	
t_{PF}	Input Data Hold Time	0	160	ns	
t_{DP}	Output Data Setup Time	230		ns	
t_{PD}	Output Data Hold Time	25		ns	
t_{PP}	PROG Pulse Width	920		ns	
t_{PL}	Port 2 I/O Data Setup	300		ns	
t_{LP}	Port 2 I/O Data Hold	120		ns	

PORT 2 TIMING



CRYSTAL OSCILLATOR MODE

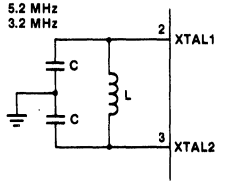


C2 = 5pF + 1/2pF + STRAY 5pF
C2 = CRYSTAL + STRAY 8pF
C3 = 20pF + 1pF + STRAY 5pF

CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 75 AT 6MHz; LESS THAN 180 AT 3.6MHz.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



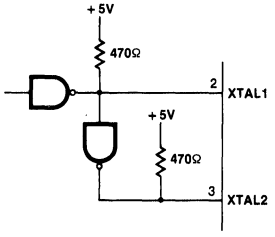
$f \approx \frac{1}{2\pi\sqrt{LC'}}$

$C' = \frac{C + 3C_{PP}}{2}$

$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

DRIVING FROM EXTERNAL SOURCE



FOR XTAL 1 AND XTAL 2 DEFINE "HIGH" AS VOLTAGES ABOVE 1.6V AND "LOW" AS VOLTAGES BELOW 1.6V. THE DUTY CYCLE REQUIREMENTS FOR EXTERNALLY DRIVING XTAL 1 AND XTAL 2 USING THE CIRCUIT SHOWN ABOVE ARE AS FOLLOWS:

FOR THE 8048, XTAL 1 MUST BE HIGH 35-65% OF THE PERIOD AND XTAL 2 MUST BE HIGH 35-65% OF THE PERIOD.

FOR THE 8748, XTAL MUST BE HIGH 45-50% OF THE PERIOD AND XTAL 2 MUST BE HIGH 50-55% OF THE PERIOD.

RISE AND FALL TIMES MUST NOT EXCEED 20ns.

PROGRAMMING, VERIFYING, AND ERASING THE 8748 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6 MHz)
$\overline{\text{RESET}}$	Initialization and Address Latching
TEST 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output during Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a mis-socketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify Sequence is:

1. V_{DD} = 5V, clock applied or internal oscillator operating, $\overline{\text{RESET}}$ = 0V, TEST 0 = 5V, EA = 5V, BUS and PROG floating.
2. Insert 8748 in programming socket.
3. TEST 0 = 0V (select program mode).
4. EA = 23V (activate program mode).
5. Address applied to BUS and P20-1.
6. $\overline{\text{RESET}}$ = 5V (latch address).
7. Data applied to BUS.
8. V_{DD} = 25V (programming power).
9. PROG = 0V followed by one 50 ms pulse to 23 V.
10. V_{DD} = 5V.
11. TEST 0 = 5V (verify mode).
12. Read and verify data on BUS.
13. TEST 0 = 0V.
14. $\overline{\text{RESET}}$ = 0V and repeat from step 5.
15. Programmer should be at conditions of step 1 when 8748 is removed from socket.

AC TIMING SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$ ↑	4tcy			
t_{WA}	Address Hold Time After $\overline{RESET}$ ↑	4tcy			
t_{DW}	Data in Setup Time to PROG ↑	4tcy			
t_{WD}	Data in Hold Time After PROG ↓	4tcy			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD}	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time Before EA ↑	4tcy			

Note: If Test 0 is high t_{DO} can be triggered by $\overline{RESET}$ ↑

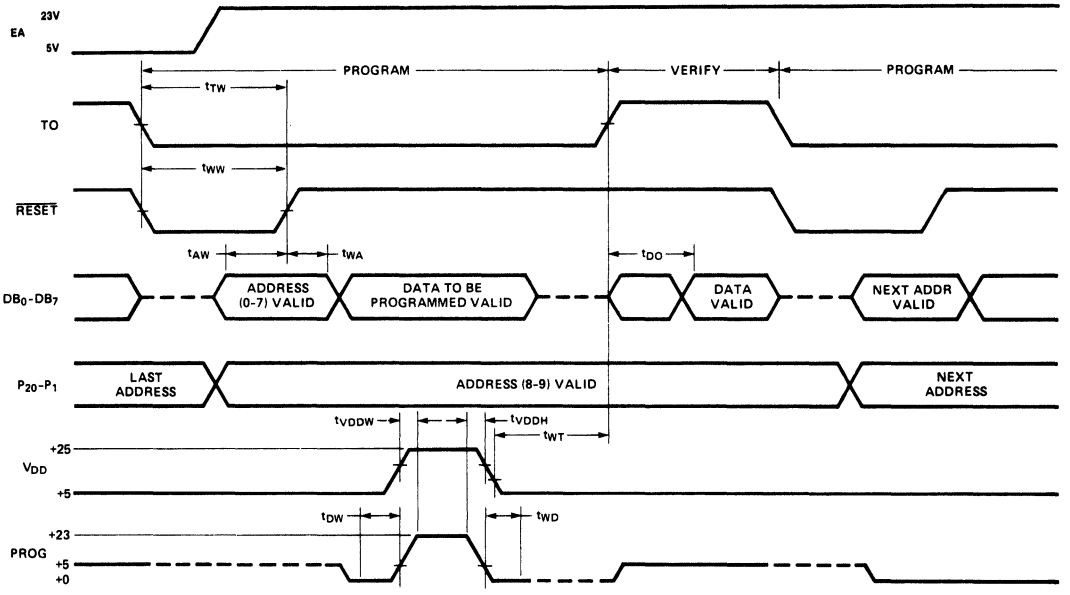
DC SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

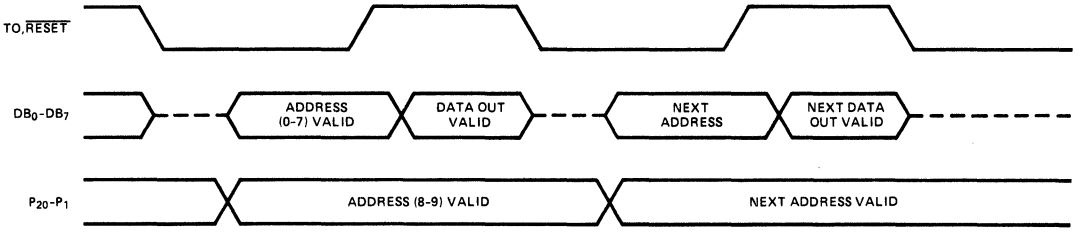
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	8748
V_{EAH1}	EA1 Verify Voltage High Level	11.4	12.6	V	8048
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. PROG MUST FLOAT IF EA IS LOW (i.e., ≠23V), OR IF T0 = 5V FOR THE 8748. FOR THE 8048 PROG MUST ALWAYS FLOAT.
2. X_1 AND X_2 DRIVEN BY 3 MHz CLOCK WILL GIVE 5μsec t_{CY}. THIS IS ACCEPTABLE FOR -8 PARTS AS WELL AS STANDARD PARTS.

The 8748 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP series) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

Note: See the ROM/PROM section for 8048 ROM ordering procedures. To minimize turnaround time on the first 25 pieces 8648 may be specified on the ROM order.

ID8048/8748/8035L
INDUSTRIAL TEMPERATURE RANGE
SINGLE COMPONENT 8-BIT MICROCOMPUTER

- **8048 Mask Programmable ROM**
- **8648 One-Time Factory Programmable EPROM**
- **8748 User Programmable/Erasable EPROM**
- **8035/8035L External ROM or EPROM**
- **−40°C to +85°C Operation**

- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **Interchangeable ROM and EPROM Versions**
- **Single 5V Supply**
- **2.5 μsec and 5.0 μsec Cycle Versions: All instructions 1 or 2 Cycles**
- **Over 90 Instructions: 70% Single Byte**
- **1K x 8 ROM/EPROM 64 x 8 RAM 27 I/O LINES**
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with 8080/8085 Series Peripherals**
- **Single Level Interrupt**

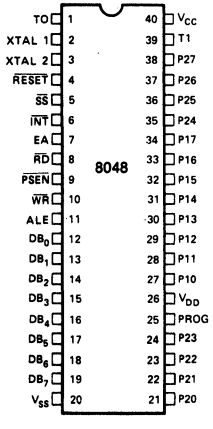
The Intel® 8048/8648/8748/8035 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8048 contains a 1K x 8 program memory, a 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability, the 8048 can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8035 is the equivalent of an 8048 without program memory. The 8035L has the RAM power-down mode of the 8048 while the 8035 does not. The 8648 is a one-time programmable (at the factory) 8748 which can be ordered as the first 25 pieces of a new 8048 ROM order. The substitution of 8648's for 8048's allows for very fast turnaround for initial code verification and evaluation units.

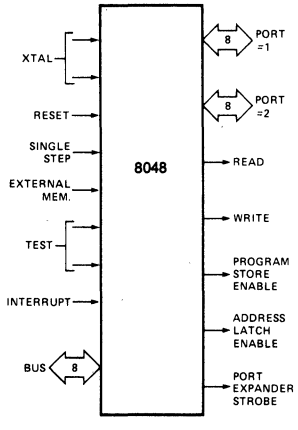
To reduce development problems to a minimum and provide maximum flexibility, three interchangeable pin-compatible versions of this single component microcomputer exist: the 8748 with user-programmable and erasable EPROM program memory for prototype and preproduction systems, the 8048 with factory-programmed mask ROM program memory for low cost, high volume production, and the 8035 without program memory for use with external program memories.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8048 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

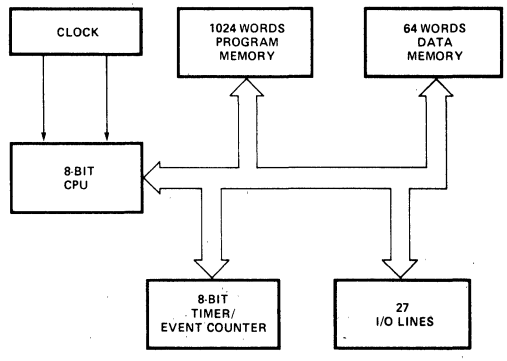
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 and 8035L.
V_{CC}	40	Main power supply; +5V during operation and programming.
PROG	25	Program pulse (+23V) input pin during 8748 programming.
		Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB_0-DB_7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched.
		Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

Designation	Pin #	Function
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device.
		Used as a read strobe to external data memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during PROM programming verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a bus write. (Active low)
		Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output.
		The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input expander port to A	1	2
	MOVD P, A	Output A to expander port	1	2
	ANLD P, A	And A to expander port	1	2
	ORLD P, A	Or A to expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on carry = 1	2	2
	JNC addr	Jump on carry = 0	2	2
	J Z addr	Jump on A zero	2	2
	JNZ addr	Jump on A not zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on accumulator bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL addr	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear carry	1	1
	CPL C	Complement carry	1	1
	CLR F0	Clear flag 0	1	1
	CPL F0	Complement flag 0	1	1
	CLR F1	Clear flag 1	1	1
	CPL F1	Complement flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from page 3	1	2
Timer/Counter	MOV A, T	Read timer/counter	1	1
	MOV T, A	Load timer/counter	1	1
	STRT T	Start timer	1	1
	STRT CNT	Start counter	1	1
	STOP TCNT	Stop timer/counter	1	1
	EN TCNTI	Enable timer/counter interrupt	1	1
	DIS TCNTI	Disable timer/counter interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable clock output on T0	1	1
	NOP	No operation	1	1

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . – 40°C to + 85°C
Storage Temperature . . . . . – 65°C to + 125°C
Voltage On Any Pin With Respect to Ground . . . . . – 0.5V to + 7V
Power Dissipation . . . . . 1.5 Watt

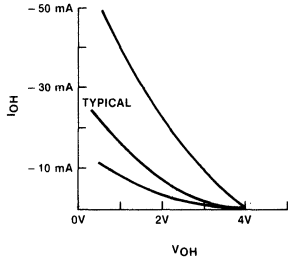
*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. AND OPERATING CHARACTERISTICS

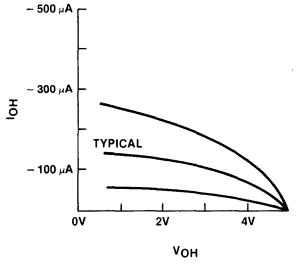
$T_A = -40°C$ to $+85°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All Except $\overline{RESET}$, XTAL1, XTAL2)	– 0.5		0.8	V	
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	– 0.5		0.6	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL 2, $\overline{RESET}$)	2.2		V_{CC}	V	
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	$I_{OL} = 1.6$ mA
V_{OL1}	Output Low Voltage (P10-P17, P20-P27)			0.45	V	$I_{OL} = 1.2$ mA
V_{OL2}	Output Low Voltage (All Other Outputs)			0.45	V	$I_{OL} = 0.8$ mA
V_{OH}	Output High Voltage (BUS)	2.4			V	$I_{OH} = -280\ \mu A$
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{ALE}$, PSEN)	2.4			V	$I_{OH} = -80\ \mu A$
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	$I_{OH} = -30\ \mu A$
I_{LI}	Input Leakage Current (T1, $\overline{INT}$)			± 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{LI1}	Input Leakage Current (P10-P17, P20-P27, EA, $\overline{SS}$)			– 600	μA	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current (BUS, T0) (High Impedance State)			± 10	μA	$V_{SS} + 0.45 \leq V_{IN} \leq V_{CC}$
I_{DD}	V_{DD} Supply Current		10	20	mA	
$I_{DD} + I_{CC}$	Total Supply Current		75	145	mA	

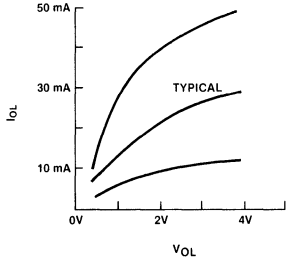
BUS



P1, P2

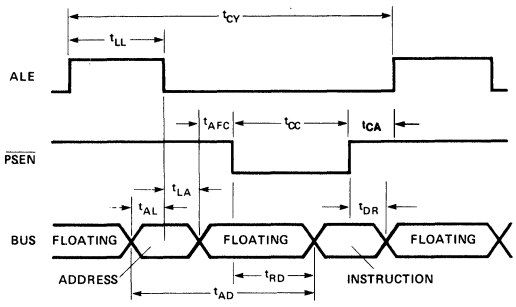


BUS, P1, P2

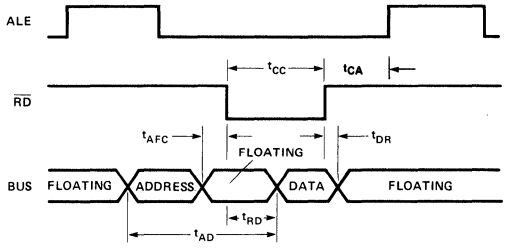


WAVEFORMS

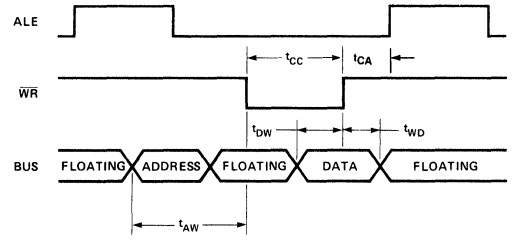
Instruction Fetch From External Program Memory



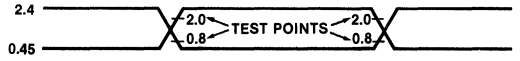
Read From External Data Memory



Write to External Data Memory



Input and Output Waveforms for A.C. Tests



A.C. CHARACTERISTICS

$T_A = -40°C$ to $+85°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	8048/8035L		8748/8035		Unit	Conditions (Note 1)
		Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	200		300		ns	
t_{AL}	Address Setup to ALE	120		120		ns	
t_{LA}	Address Hold from ALE	80		80		ns	
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)	400		600		ns	
t_{DW}	Data Setup Before $\overline{WR}$	420		600		ns	
t_{WD}	Data Hold After $\overline{WR}$	80		120		ns	$C_L = 20$ pF
t_{CY}	Cycle Time	2.5	15.0	4.17	15.0	μs	(3.6 MHz XTAL 8748/8035)
t_{DR}	Data Hold	0	200	0	200	ns	
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In		400		600	ns	
t_{AW}	Address Setup to $\overline{WR}$	230		260		ns	
t_{AD}	Address Setup to Data In		600		900	ns	
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	−40		−60		ns	
t_{CA}	Control Pulse to ALE	10		10		ns	

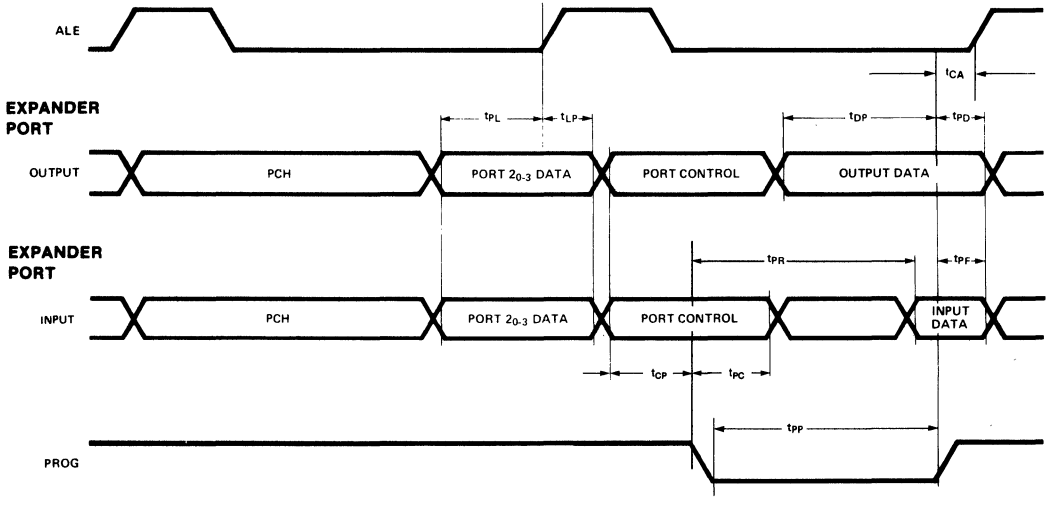
Note 1: Control Outputs: $C_L = 80$ pF $t_{CY} = 2.5$ μs for 8048/8035L
BUS Outputs: $C_L = 150$ pF 4.17 μs for 8748/8035

A.C. CHARACTERISTICS

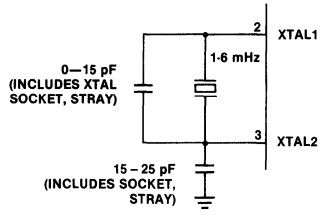
$T_A = -40°C$ to $+85°C$, $V_{CC} = 5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of PROG	115		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	65		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		860	ns	
t_{PF}	Input Data Hold Time	0	160	ns	
t_{DP}	Output Data Setup Time	230		ns	
t_{PD}	Output Data Hold Time	25		ns	
t_{PP}	PROG Pulse Width	920		ns	
t_{PL}	Port 2 I/O Data Setup	300		ns	
t_{LP}	Port 2 I/O Data Hold	120		ns	

PORT 2 TIMING

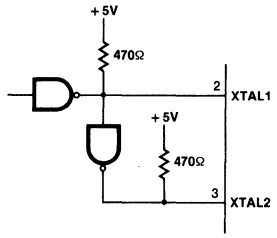


CRYSTAL OSCILLATOR MODE



CRYSTAL SERIES RESISTANCE SHOULD BE <75Ω AT 6 MHz; <180Ω AT 3.6 MHz.

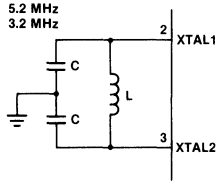
DRIVING FROM EXTERNAL SOURCE



BOTH X1 AND X2 SHOULD BE DRIVEN.
RESISTORS TO V_{CC} ARE NEEDED TO ENSURE V_{IH} = 3.8V IF TTL CIRCUITRY IS USED. THE MINIMUM HIGH AND THE MINIMUM LOW TIMES ARE 45%.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{PP}}{2}$$

$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

PROGRAMMING, VERIFYING, AND ERASING THE 8748 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6 MHz)
$\overline{\text{RESET}}$	Initialization and Address Latching
TEST 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output during Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a mis-socketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify Sequence is:

1. V_{DD} = 5V, clock applied or internal oscillator operating, $\overline{\text{RESET}}$ = 0V, TEST 0 = 5V, EA = 5V, BUS and PROG floating.
2. Insert 8748 in programming socket.
3. TEST 0 = 0V (select program mode).
4. EA = 23V (activate program mode).
5. Address applied to BUS and P20-1.
6. $\overline{\text{RESET}}$ = 5V (latch address).
7. Data applied to BUS.
8. V_{DD} = 25V (programming power).
9. PROG = 0V followed by one 50 ms pulse to 23 V.
10. V_{DD} = 5V.
11. TEST 0 = 5V (verify mode).
12. Read and verify data on BUS.
13. TEST 0 = 0V.
14. $\overline{\text{RESET}}$ = 0V and repeat from step 5.
15. Programmer should be at conditions of step 1 when 8748 is removed from socket.

AC TIMING SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}\uparrow$	4tcy			
t_{WA}	Address Hold Time After $\overline{RESET}\uparrow$	4tcy			
t_{DW}	Data in Setup Time to PROG↑	4tcy			
t_{WD}	Data in Hold Time After PROG↓	4tcy			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD}	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG↓	0			
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time Before EA↑	4tcy			

Note: If Test 0 is high t_{DO} can be triggered by $\overline{RESET}$!

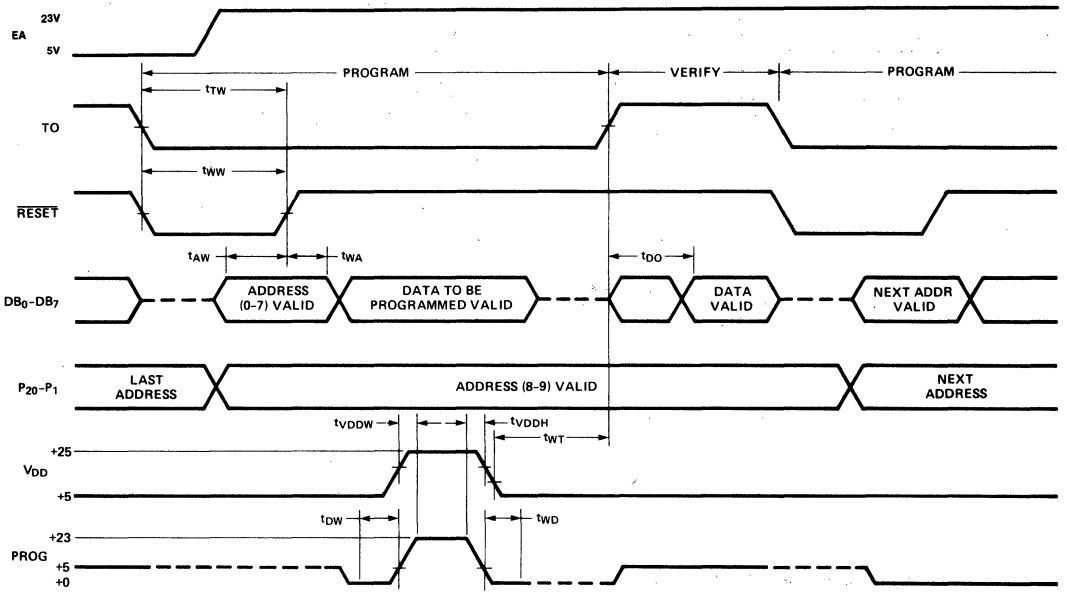
DC SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

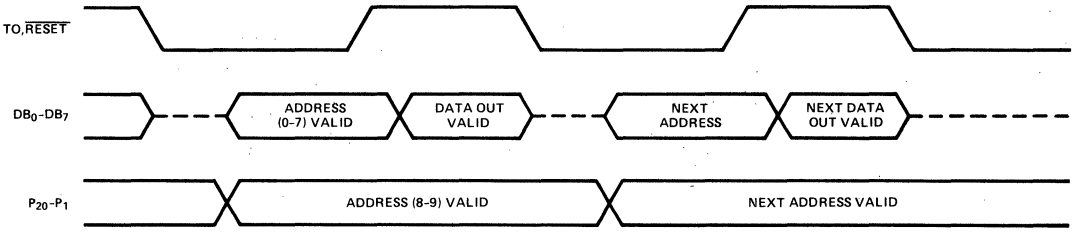
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DDH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	8748
V_{EAH1}	EA1 Verify Voltage High Level	11.4	12.6	V	8048
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. PROG MUST FLOAT IF EA IS LOW (i.e., ≠23V), OR IF T0 = 5V FOR THE 8748. FOR THE 8048 PROG MUST ALWAYS FLOAT.
2. X_1 AND X_2 DRIVEN BY 3 MHz CLOCK WILL GIVE 5μsec t_{CY}. THIS IS ACCEPTABLE FOR -8 PARTS AS WELL AS STANDARD PARTS.

The 8748 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP series) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

Note: See the ROM/PROM section for 8048 ROM ordering procedures. To minimize turnaround time on the first 25 pieces 8648 may be specified on the ROM order.

M8048/M8748/M8035L

SINGLE COMPONENT 8-BIT MICROCOMPUTER

*** 8048 Mask Programmable ROM**
*** 8748 User Programmable/Erasable EPROM**
*** 8035L Requires External ROM or EPROM**

- **-55°C to +125°C 6 MHz Operation (M8048/M8035L)**
- **-55°C to +100°C 3.6 MHz Operation (M8748)**
- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **Interchangeable ROM and EPROM Versions**
- **Single 5V Supply**
- **2.5 µsec and 5.0 µsec Cycle Versions All Instructions 1 or 2 Cycles.**
- **Over 90 Instructions: 70% Single Byte**
- **1K x 8 ROM/EPROM 64 x 8 RAM 27 I/O Lines**
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with 8080/8085 Series Peripherals**
- **Single Level Interrupt**

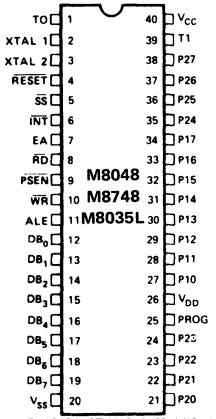
The Intel M8048/M8748/M8035L are totally self-sufficient 8-bit parallel computers fabricated on single silicon chips using Intel's N-Channel silicon gate MOS process.

The M8048 contains an 8-bit CPU, a 1K x 8 program memory, a 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability, the M8048 can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The M8035L is the equivalent of an M8048 without program memory, and has the RAM power down mode of the M8048. To reduce development problems to a minimum and provide maximum flexibility, three interchangeable pin-compatible* versions of this single component microcomputer exist: the M8748 with user-programmable and erasable EPROM program memory for prototype and preproduction systems, the M8048 with factory-programmed mask ROM program memory for low cost, high volume production, and the M8035L without program memory for use with external program memories.

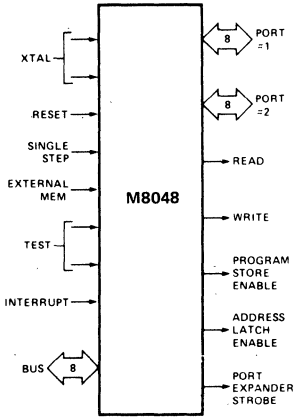
This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The M8048 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

*V_{DD} is used to program the M8748 and used for low power standby on the M8048/8035L.

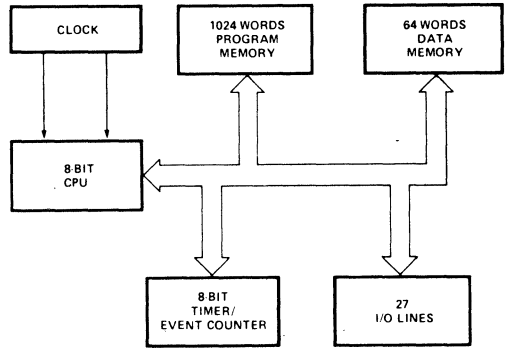
PIN CONFIGURATION



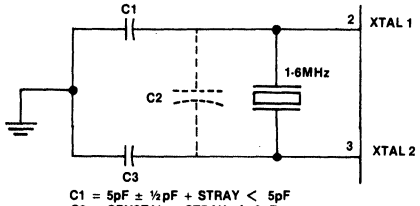
LOGIC SYMBOL



BLOCK DIAGRAM



CRYSTAL OSCILLATOR MODE



C1 = 5pF ± ½pF + STRAY < 5pF
C2 = CRYSTAL + STRAY < 8pF
C3 = 20pF ± 1pF + STRAY < 5pF

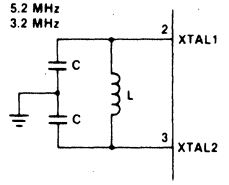
CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 75Ω AT 6MHz; LESS THAN 180Ω AT 3.6MHz.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz

$$f = \frac{1}{2\pi\sqrt{LC'}}$$

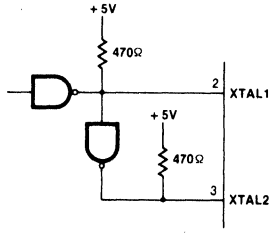
$$C' = \frac{C + 3C_{PP}}{2}$$



$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

DRIVING FROM EXTERNAL SOURCE



FOR XTAL 1 AND XTAL 2 DEFINE "HIGH" AS VOLTAGES ABOVE 1.6V AND "LOW" AS VOLTAGES BELOW 1.6V. THE DUTY CYCLE REQUIREMENTS FOR EXTERNALLY DRIVING XTAL 1 AND XTAL 2 USING THE CIRCUIT SHOWN ABOVE ARE AS FOLLOWS:

FOR THE 8048, XTAL 1 MUST BE HIGH 35-65% OF THE PERIOD AND XTAL 2 MUST BE HIGH 35-65% OF THE PERIOD.

FOR THE 8748, XTAL MUST BE HIGH 45-50% OF THE PERIOD AND XTAL 2 MUST BE HIGH 50-55% OF THE PERIOD.

RISE AND FALL TIMES MUST NOT EXCEED 20 ns.

PROGRAMMING, VERIFYING, AND ERASING THE 8748 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a missocketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. V_{DD} = 5v, Clock applied or internal oscillator operating, $\overline{\text{RESET}}$ = 0v, TEST 0 = 5v, EA = 5v, BUS and PROG floating.
2. Insert 8748 in programming socket
3. TEST 0 = 0v (select program mode)
4. EA = 23V (activate program mode)
5. Address applied to BUS and P20-1
6. $\overline{\text{RESET}}$ = 5v (latch address)
7. Data applied to BUS
8. V_{DD} = 25v (programming power)
9. PROG = 0v followed by one 50ms pulse to 23V
10. V_{DD} = 5v
11. TEST 0 = 5v (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0v
14. $\overline{\text{RESET}}$ = 0v and repeat from step 5
15. Programmer should be at conditions of step 1 when 8748 is removed from socket.

AC TIMING SPECIFICATION FOR PROGRAMMING

T_A = 25°C ± 5°C, V_{CC} = 5V ± 5%, V_{DD} = 25V ± 1V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$ ↑	4tcy			
t_{WA}	Address Hold Time After $\overline{RESET}$ ↑	4tcy			
t_{DW}	Data in Setup Time to PROG ↑	4tcy			
t_{WD}	Data in Hold Time After PROG ↓	4tcy			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD}	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	mS	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time Before EA ↓	4tcy			

Note: If Test 0 is high t_{DO} can be triggered by $\overline{RESET}$ ↓.

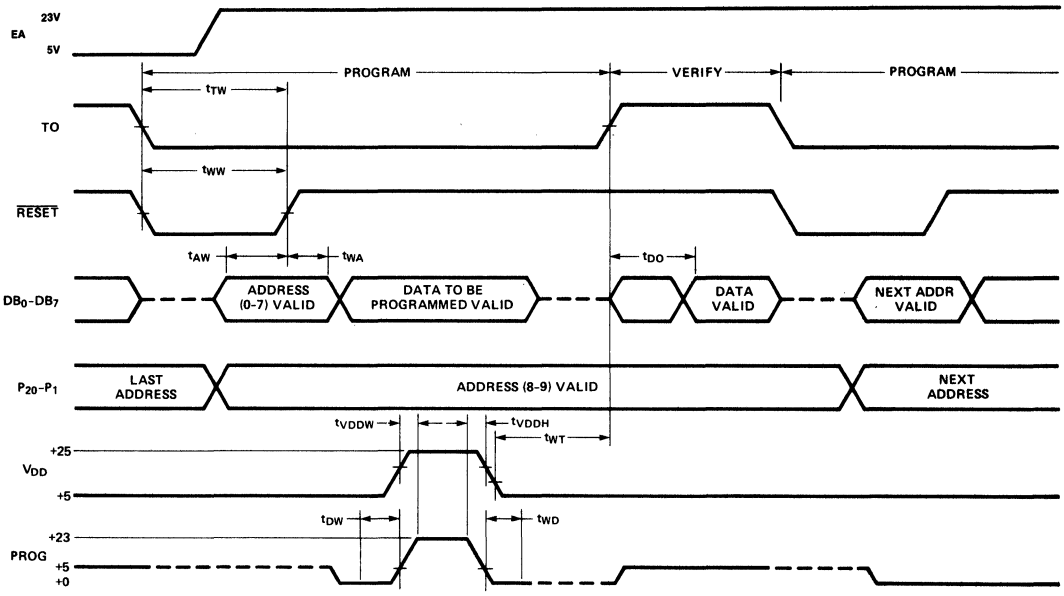
DC SPECIFICATION FOR PROGRAMMING

T_A = 25°C ± 5°C, V_{CC} = 5V ± 5%, V_{DD} = 25V ± 1V

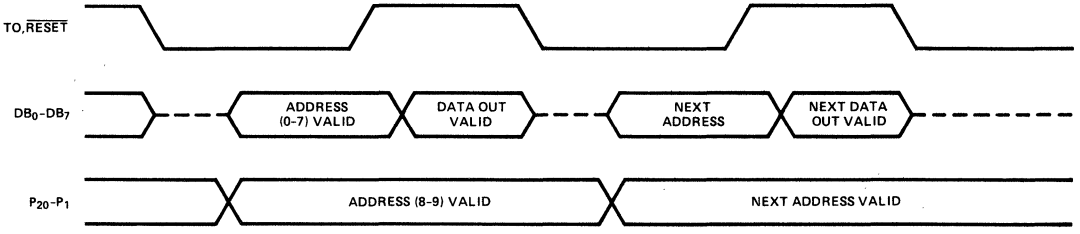
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	8748
V_{EAH1}	EA1 Verify Voltage High Level	11.4	12.6	V	8048
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. PROG MUST FLOAT IF EA IS LOW (i.e., ≠23V), OR IF T0 = 5V FOR THE 8748. FOR THE 8048 PROG MUST ALWAYS FLOAT.
2. X_1 AND X_2 DRIVEN BY 3 MHz CLOCK WILL GIVE 5μsec t_{CY}. THIS IS ACCEPTABLE FOR ALL PARTS.

The 8748 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP Series) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input expander port to A	1	2
	MOVD P, A	Output A to expander port	1	2
	ANLD P, A	And A to expander port	1	2
	ORLD P, A	Or A to expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on carry = 1	2	2
	JNC addr	Jump on carry = 0	2	2
	JZ addr	Jump on A zero	2	2
	JNZ addr	Jump on A not zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on accumulator bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL addr	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear carry	1	1
	CPL C	Complement carry	1	1
	CLR F0	Clear flag 0	1	1
	CPL F0	Complement flag 0	1	1
	CLR F1	Clear flag 1	1	1
	CPL F1	Complement flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to a	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCHA, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from page 3	1	2
Timer/Counter	MOV A, T	Read timer/counter	1	1
	MOV T, A	Load timer/counter	1	1
	STRT T	Start timer	1	1
	STRT CNT	Start counter	1	1
	STOP TCNT	Stop timer/counter	1	1
	EN TCNTI	Enable timer/counter interrupt	1	1
	DIS TCNTI	Disable timer/counter interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable clock output on T0	1	1
	NOP	No operation	1	1

PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 and 8035L.
V_{CC}	40	Main power supply; +5V during operation and programming.
PROG	25	Program pulse (+23V) input pin during 8748 programming. Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB_0-DB_7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENTO CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.

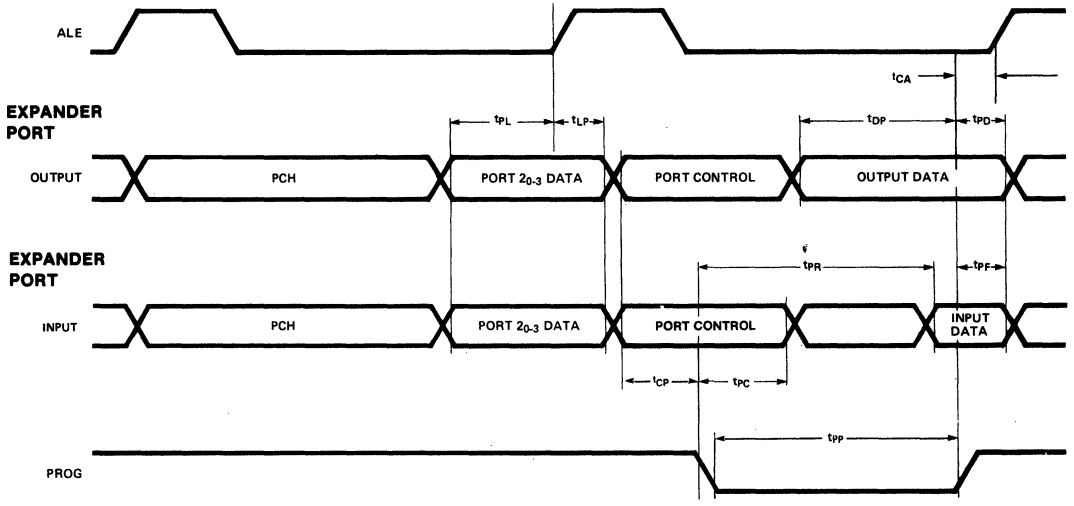
Designation	Pin#	Function
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during PROM programming verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in junction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})
XTAL2	3	Other side of crystal input.

A.C. CHARACTERISTICS (PORT 2 TIMING)

$T_A = 55°C$ to (100°C M8748/125°C M8048/M8035L), $V_{CC} = +5V \pm 10\%$, $V_{SS} = 0V$

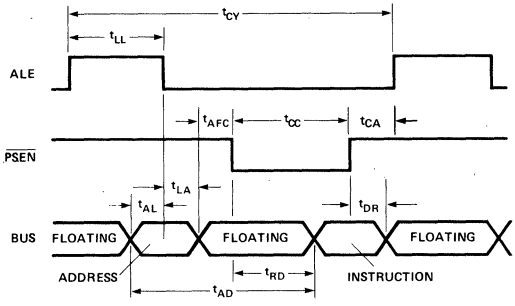
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of $\overline{PROG}$	115		ns	
t_{PC}	Port Control Hold After Falling Edge of $\overline{PROG}$	65		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		860	ns	
t_{PF}	Input Data Hold Time	0	160	ns	
t_{DP}	Output Data Setup Time	230		ns	
t_{PD}	Output Data Hold Time	25		ns	
t_{PP}	$\overline{PROG}$ Pulse Width	920		ns	
t_{PL}	Port 2 I/O Data Setup	300		ns	
t_{LP}	Port 2 I/O Data Hold	120		ns	

PORT 2 TIMING

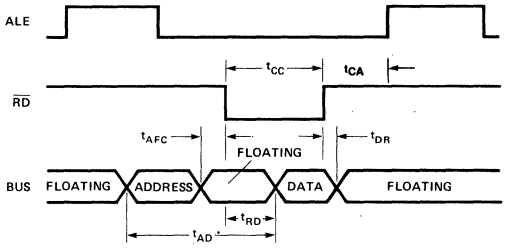


WAVEFORMS

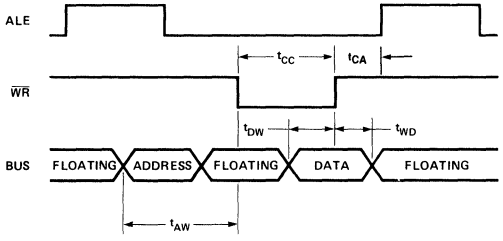
Instruction Fetch From External Program Memory



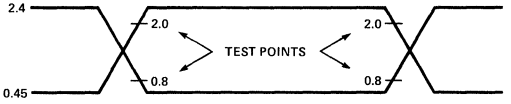
Read From External Data Memory



Write to External Data Memory



Input and Output Waveforms for A.C. Tests



A.C. CHARACTERISTICS

$T_A = -55°C$ to (100°C M8748/125°C M8048/M8035L), $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	M8048 M8035L		M8748		Unit	Conditions (Note 1)
		Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	200		300		ns	
t_{AL}	Address Setup to ALE	120		120		ns	
t_{LA}	Address Hold from ALE	80		80		ns	
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)	400		600		ns	
t_{DW}	Data Setup before $\overline{WR}$	420		600		ns	
t_{WD}	Data Hold After $\overline{WR}$	80		120		ns	C_L = 20pF
t_{CY}	Cycle Time	2.5	15.0	4.17	15.0	μs	(3.6 MHz XTAL 8748)
t_{DR}	Data Hold	0	200	0	200	ns	
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In		400		600	ns	
t_{AW}	Address Setup to $\overline{WR}$	230		260		ns	
t_{AD}	Address Setup to Data In		600		900	ns	
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	-40		-60		ns	
t_{CA}	Control Pulse to ALE	10		10		ns	

Note 1: Control outputs: $C_L = 80$ pF $t_{CY} = 2.5\mu s$ for 8048/8035L
BUS Outputs: $C_L = 150$ pF $4.17\mu s$ for 8748

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias
- 8748 ..... -55°C to +100°C
- 8048/8035L ..... -55°C to +125°C

Storage Temperature ..... -65°C to +125°C

Voltage On Any Pin With Respect to Ground ..... -0.5 to +7V

Power Dissipation ..... 1.5 Watt

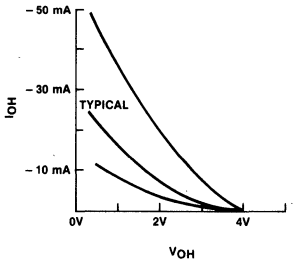
COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C.AND OPERATING CHARACTERISTICS

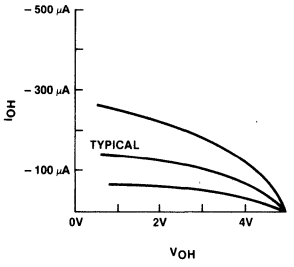
$T_A = -55°C$ to (100°C M8748/125°C M8048/M8035L), $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All Except $\overline{RESET}$, X1, X2)	-.5		.7	V	
V_{IL1}	Input Low Voltage ($\overline{RESET}$, X1, X2)	-.5		.5	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL 2, $\overline{RESET}$)	2.3		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, X1, X2)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			.45	V	$I_{OL} = 1.2mA$
V_{OL1}	Output Low Voltage (All Other Outputs)			.45	V	$I_{OL} = 0.8mA$
V_{OH}	Output High Voltage (BUS)	2.4			V	$I_{OH} = -240\mu A$
V_{OH1}	Output High Voltage ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	$I_{OH} = -50\mu A$
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	$I_{OH} = -30\mu A$
I_{LI}	Input Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{LI1}	Input Leakage Current (P10-P17, P20-P27, EA, $\overline{SS}$)			-700	μA	$V_{SS}+.45 \leq V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current (BUS, TO) (High Impedance State)			±10	μA	$V_{SS}+.45 \leq V_{IN} \leq V_{CC}$
I_{DD}	V_{DD} Supply Current		10	25	mA	
$I_{DD}+I_{CC}$	Total Supply Current		80	155	mA	

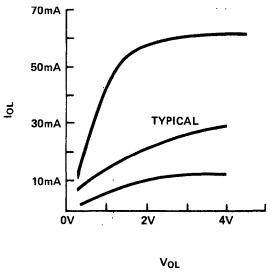
BUS



P1, P2



BUS, P1, P2



NEW HIGH PERFORMANCE 8049/8039/8039-6 SINGLE COMPONENT 8-BIT MICROCOMPUTER

*8049 Mask Programmable ROM
*8039 External ROM or EPROM
*New 11 MHz Operation

- 8-Bit CPU, ROM, RAM, I/O in Single Package
- Single 5V ± 10% Supply
- 1.36 μsec Cycle; All Instructions 1 or 2 Cycles
- Over 90 Instructions: 70% Single Byte
- Pin Compatible with 8048/8748
- 2K x 8 ROM
  128 x 8 RAM
  27 I/O Lines
- Interval Timer/Event Counter
- Easily Expandable Memory and I/O
- Compatible with MCS Memory and I/O
- Single Level Interrupt

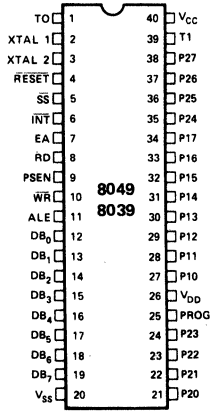
The Intel® 8049/8039/8039-6 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8049 contains a 2K x 8 program memory, a 128 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on board oscillator and clock circuits. For systems that require extra capability, the 8049 can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8039 is the equivalent to an 8049 without program memory. The 8039-6 is a lower speed (6 MHz) version of the 8039.

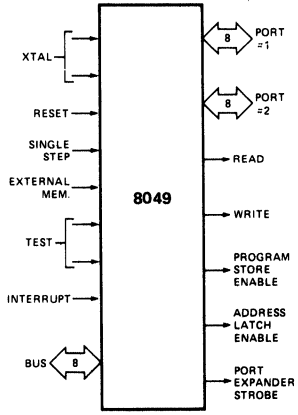
To reduce development problems to a minimum and provide maximum flexibility, two interchangeable pin-compatible versions of this single component microcomputer exist: the 8049 with factory-programmed mask ROM program memory for low-cost high volume production, and the 8039 without program memory for use with external program memories in prototype and preproduction systems.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8049 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

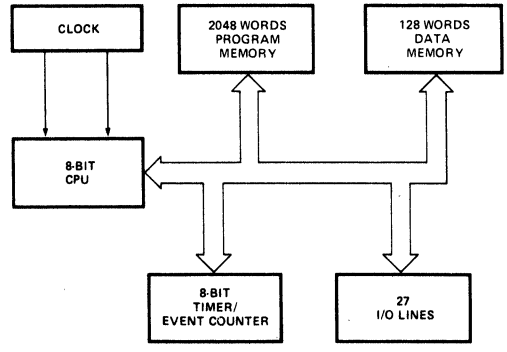
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	+5V during operation. Low power standby pin.
V_{CC}	40	Main power supply; +5V during operation.
PROG	25	Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243
D0-D7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

Designation	Pin #	Function
$\overline{RD}$	8	**Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device.** Used as a Read Strobe to External Data Memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to External Data Memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Not TTL Compatible)
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycle
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	Move A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin With Respect to Ground ..... -0.5V to +7V
Power Dissipation ..... 1.5 Watt

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

D.C. AND OPERATING CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, X1, X2)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	$I_{OL} = 2.0mA$
V_{OL1}	Output Low Voltage (All Other Outputs Except PROG)			0.45	V	$I_{OL} = 1.6mA$
V_{OL2}	Output Low Voltage (PROG)			0.45	V	$I_{OL} = 1.0mA$
V_{OH}	Output High Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	$I_{OH} = -100\mu A$
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			V	$I_{OH} = -50\mu A$
I_{IL}	Input Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leqslant V_{IN} \leqslant V_{CC}$
I_{OL}	Output Leakage Current (Bus, T0) (High Impedance State)			±10	μA	$V_{SS} + 0.45 \leqslant V_{IN} \leqslant V_{CC}$
I_{DD}	Power Down Supply Current		25	50	mA	$T_A = 25°C$
$I_{DD}+I_{CC}$	Total Supply Current		100	170	mA	$T_A = 25°C$

A.C. CHARACTERISTICS

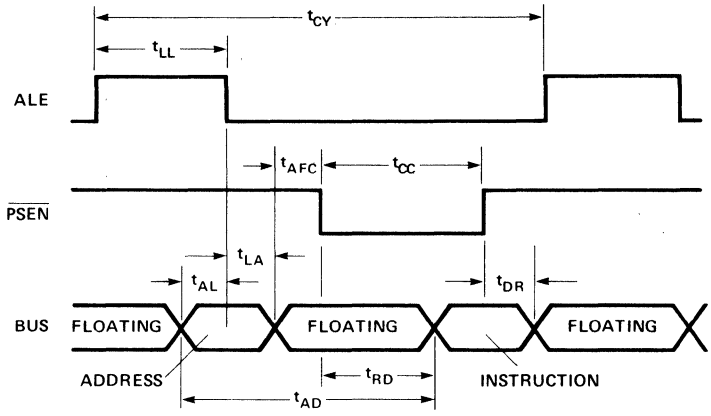
$T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Min.	Max.	Unit	Conditions (Note 2)
t_{LL}	ALE Pulse Width	200		ns	
t_{AL}	Address Setup to ALE	120		ns	
t_{LA}	Address Hold from ALE	80		ns	
t_{CC}	Control Pulse Width (PSEN, RD, WR)	400		ns	
t_{DW}	Data Set-Up Before WR	420		ns	
t_{WD}	Data Hold After WR	80		ns	$C_L = 20pF$
t_{CY}	Cycle Time	2.5	15.0	μs	11MHz XTAL
t_{DR}	Data Hold	0	200	ns	
t_{RD}	PSEN, RD to Data In		400	ns	
t_{AW}	Address Setup to WR	230		ns	
t_{AD}	Address Setup to Data In		600	ns	
t_{AFC}	Address Float to RD, PSEN	-40		ns	

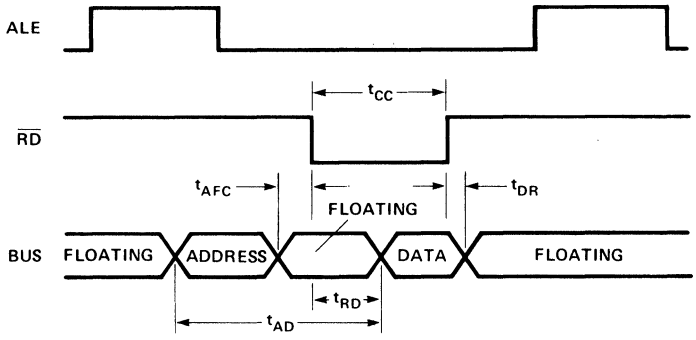
Notes: 1. 8039-6 specifications are also valid for 8049/8039 operating at 6MHz.
2. Control Outputs: $C_L = 80pF$
BUS Outputs: $C_L = 150pF$

WAVEFORMS

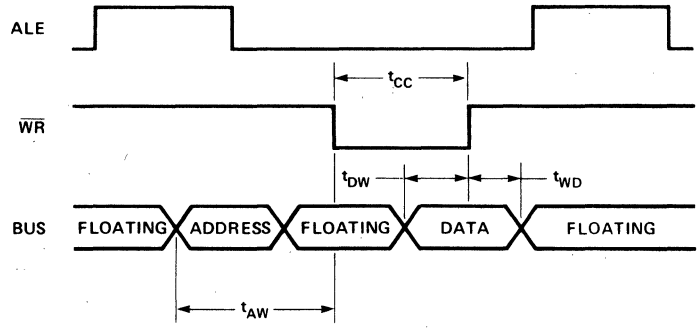
INSTRUCTION FETCH FROM EXTERNAL PROGRAM MEMORY



READ FROM EXTERNAL DATA MEMORY



WRITE TO EXTERNAL DATA MEMORY



NEW HIGH PERFORMANCE
I8049/8039
SINGLE COMPONENT 8-BIT MICROCOMPUTER

*8049 Mask Programmable ROM
*8039 External ROM or EPROM
*6 MHz Operation

- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **Single 5V ± 10% Supply**
- **2.5 μsec Cycle; All Instructions 1 or 2 Cycles**
- **Over 90 Instructions: 70% Single Byte**
- **Pin Compatible with 8048/8748**
- **2K x 8 ROM**
  128 x 8 RAM
  27 I/O Lines
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with MCS Memory and I/O**
- **Single Level Interrupt**

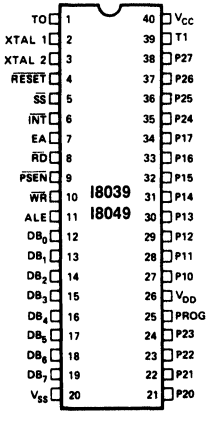
The Intel® I8049/8039 is totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8049 contains a 2K x 8 program memory, a 128 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability, the 8049 can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8039 is the equivalent of an 8049 without program memory.

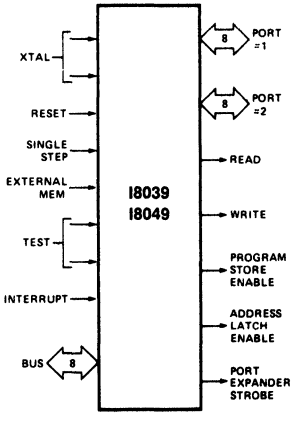
To reduce development problems to a minimum and provide maximum flexibility, two interchangeable pin-compatible versions of this single component microcomputer exist: the 8049 with factory-programmed mask ROM program memory for low-cost high volume production, and the 8039 without program memory for use with external program memories in prototype and preproduction systems.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8049 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

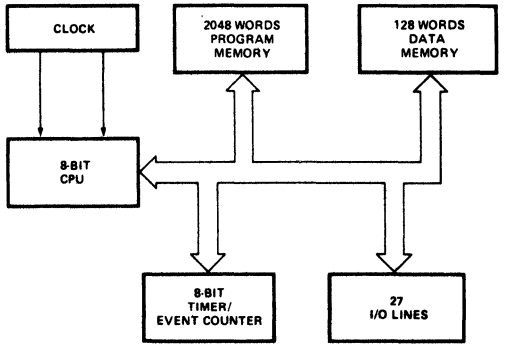
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



AFN-00737-01

PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	+5V during operation. Low power standby pin.
V_{CC}	40	Main power supply; +5V during operation.
PROG	25	Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243
D0-D7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)

Designation	Pin #	Function
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. Used as a Read Strobe to External Data Memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to External Data Memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Not TTL Compatible)
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	Move A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCHA,@R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... -40°C to +85°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin With
Respect to Ground .....-0.5V to +7V
Power Dissipation ..... 1.5 Watt

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

D.C. AND OPERATING CHARACTERISTICS

T_A = -40°C to +85°C, V_{CC} = V_{DD} = +5V ± 10%, V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	-0.5		0.6	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.2		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, X1, X2)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	I_{OL} = 1.6 ma
V_{OL1}	Output Low Voltage (All Other Outputs Except PROG)			0.45	V	I_{OL} = 1.2 ma
V_{OL2}	Output Low Voltage (PROG)			0.45	V	I_{OL} = 0.8 ma
V_{OH}	Output High Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = -80 μa
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = -30 μa
I_{IL}	Input Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leqslant V_{IN} \leqslant V_{CC}$
I_{OL}	Output Leakage Current (Bus, T0) (High Impedance State)			±10	μA	$V_{SS} + 0.45 \leqslant V_{IN} \leqslant V_{CC}$
I_{DD}	Power Down Supply Current			50	mA	T_A = 25°C
$I_{DD}+I_{CC}$	Total Supply Current			170	mA	T_A = 25°C

A.C. CHARACTERISTICS

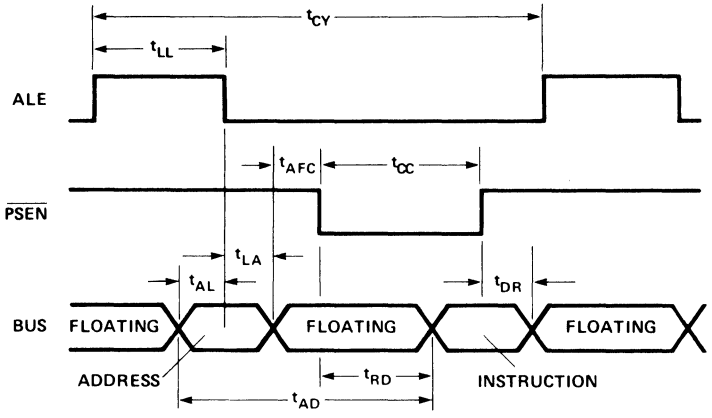
T_A = -40°C to +85°C, V_{CC} = V_{DD} = +5V ±10%, V_{SS} = 0V

Symbol	Parameter	Min.	Max.	Unit	Conditions (Note 2)
t_{LL}	ALE Pulse Width	200		ns	
t_{AL}	Address Setup to ALE	120		ns	
t_{LA}	Address Hold from ALE	80		ns	
t_{CC}	Control Pulse Width (PSEN, RD, WR)	400		ns	
t_{DW}	Data Set-Up Before WR	420		ns	
t_{WD}	Data Hold After WR	80		ns	C_L = 20pF
t_{CY}	Cycle Time	2.5	15.0	μs	(6 MHz XTAL for ID8049)
t_{DR}	Data Hold	0	200	ns	
t_{RD}	PSEN, RD to Data In		400	ns	
t_{AW}	Address Setup to WR	230		ns	
t_{AD}	Address Setup to Data In		600	ns	
t_{AFC}	Address Float to RD, PSEN	-40		ns	

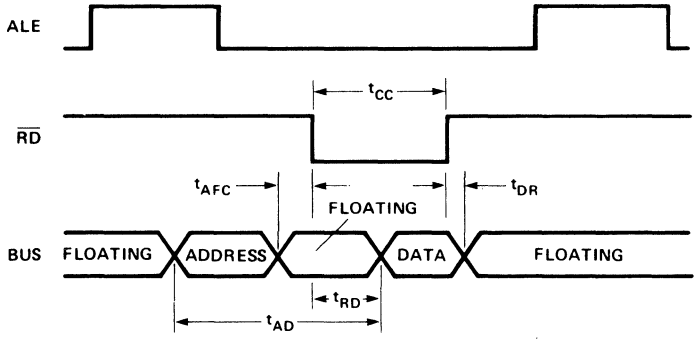
Notes: 1. 8039-6 specifications are also valid for 8049/8039 operating at 6MHz.
2. Control Outputs: C_L = 80pF
BUS Outputs: C_L = 150pF

WAVEFORMS

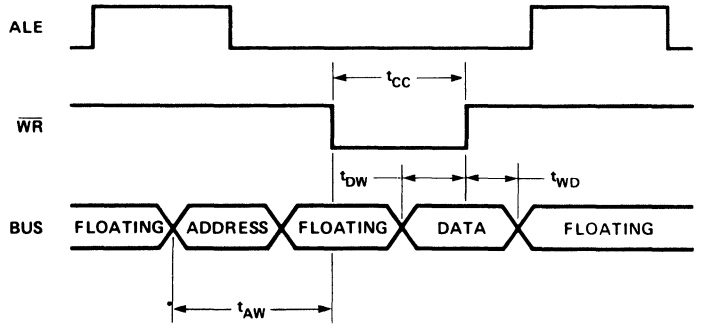
INSTRUCTION FETCH FROM EXTERNAL PROGRAM MEMORY



READ FROM EXTERNAL DATA MEMORY



WRITE TO EXTERNAL DATA MEMORY



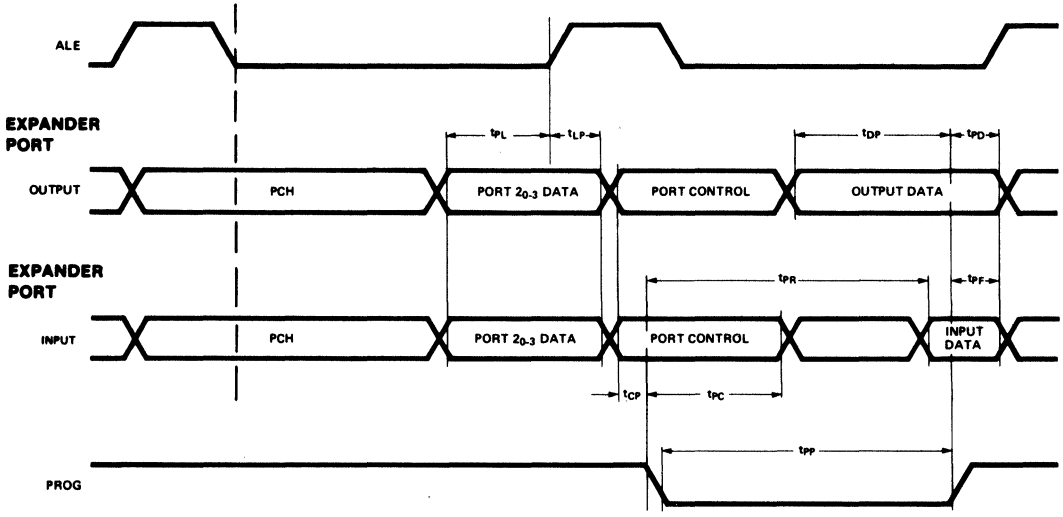
A.C. CHARACTERISTICS

T_A = -40°C to +85°C, V_{CC} = V_{DD} = +5V ±10%, V_{SS} = 0V

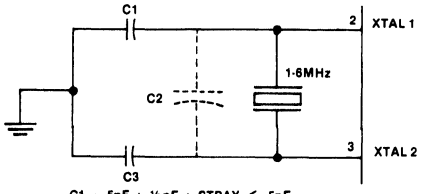
Symbol	Parameter	Min.	Max.	Unit	Conditions (Note 2)
t_{CP}	Port Control Setup Before Falling Edge of PROG	115		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	65		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		860	ns	
t_{DP}	Output Data Setup Time	230		ns	
t_{PD}	Output Data Hold Time	25		ns	
t_{PF}	Input Data Hold Time	0	160	ns	
t_{PP}	PROG Pulse Width	920		ns	
t_{PL}	Port 2 I/O Data Setup	300		ns	
t_{LP}	Port 2 I/O Data Hold	120		ns	

WAVEFORMS

PORT 2 TIMING



CRYSTAL OSCILLATOR MODE

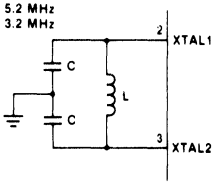


C1 = 5pF ± ½pF + STRAY < 5pF
C2 = CRYSTAL + STRAY < 8pF
C3 = 20pF ± 1pF + STRAY < 5pF

CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 75Ω AT 6MHz; LESS THAN 180Ω AT 3.6MHz.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



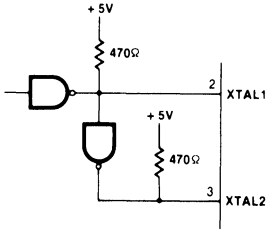
$$f \simeq \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{PP}}{2}$$

$C_{PP} \simeq 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

DRIVING FROM EXTERNAL SOURCE



FOR XTAL 1 AND XTAL 2 DEFINE "HIGH" AS VOLTAGES ABOVE 1.6V AND "LOW" AS VOLTAGES BELOW 1.6V. THE DUTY CYCLE REQUIREMENTS FOR EXTERNALLY DRIVING XTAL 1 AND XTAL 2 USING THE CIRCUIT SHOWN ABOVE ARE AS FOLLOWS:

FOR THE 8048, XTAL 1 MUST BE HIGH 35-65% OF THE PERIOD AND XTAL 2 MUST BE HIGH 35-65% OF THE PERIOD.

FOR THE 8748, XTAL MUST BE HIGH 45-50% OF THE PERIOD AND XTAL 2 MUST BE HIGH 50-55% OF THE PERIOD.

RISE AND FALL TIMES MUST NOT EXCEED 20 ns.

intel®

8021

SINGLE COMPONENT 8-BIT MICROCOMPUTER

- 8-Bit CPU, ROM, RAM, I/O in Single 28-Pin Package
- Single 5V Supply (+4.5V to 6.5V)
- 8.38 μsec Cycle With 3.58 MHz XTAL; All Instructions 1 or 2 Cycles
- Instructions—8048 Subset
- High Current Drive Capability—2 Pins
- 1K x 8 ROM
  64 x 8 RAM
  21 I/O Lines
- Interval Timer/Event Counter
- Clock Generated With Single Inductor or Crystal
- Zero-Cross Detection Capability
- Easily Expandable I/O

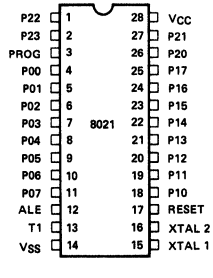
The Intel® 8021 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process. The features of the 8021 include a subset of the 8048 optimized for low cost, high volume applications, plus additional I/O flexibility and power.

The 8021 contains 1K X 8 program memory, a 64 X 8 data memory, 21 I/O lines, and an 8-bit timer/event counter, in addition to on-board oscillator and clock circuits. For systems that require extra I/O capability, the 8021 can be expanded using the 8243 or discrete logic.

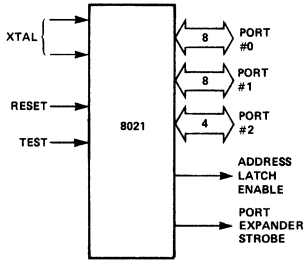
This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8021 has bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

To minimize the development problems and maximize flexibility, an 8021 system can be easily designed using the 8021 emulation board, the EM-1. The EM-1 contains a 40-pin socket which can accommodate either the 8748 shipped with the board or an ICE-49 plug. Also, the necessary discrete logic to reproduce the 8021's additional I/O features is included.

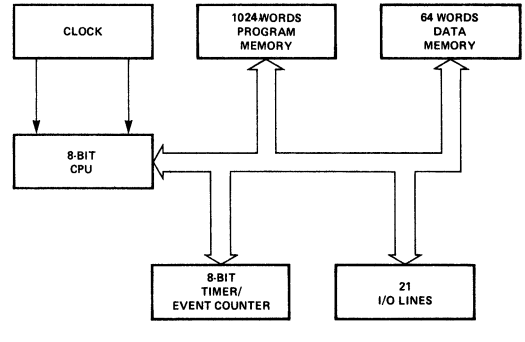
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin with
Respect to Ground .....−0.5V to + 7V
Power Dissipation .....1 W

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.5V ± 1V, V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	−0.5		0.8	V	
V_{IH}	Input High Voltage (All except XTAL 1 & 2, T1 RESET)	3.0		V_{CC}	V	
V_{IH_1}	Input High Voltage (XTAL 1 & 2, T1 RESET)	3.8		V_{CC}	V	
$V_{IH(10\%)}$	Input high voltage (all except XTAL 1 & 2, T1, RESET)	2.0		V_{CC}	V	V_{CC} = 5.0V ± 10%
$V_{IH_1(10\%)}$	Input high voltage (XTAL 1 & 2, T1, RESET)	3.5		V_{CC}	V	V_{CC} = 5.0V ± 10%
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.6 mA
V_{OL_1}	Output Low Voltage (P10, P11)			2.5	V	I_{OL} = 7 mA
V_{OH}	Output High Voltage (All unless Open Drain)	2.4			V	I_{OH} = 40 μA
I_{LO}	Output Leakage Current (Open Drain Option—Port 0)			± 10	μA	$V_{SS}+0.45 \leq V_{IN} \leq V_{CC}$
I_{CC}	V_{CC} Supply Current		40	75	mA	

T1 ZERO CROSS CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.5V ± 1V, V_{SS} = OV, C_L = 80pF

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{ZX}	Zero-Cross Detection Input (T1)	1	3	V_{PP}	AC Coupled, C = .2μF
A_{ZX}	Zero-Cross Accuracy		± 135	mV	60 Hz Sine Wave
F_{ZX}	Zero-Cross Detection Input Frequency (T1)	0.05	1	kHZ	

PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	14	Circuit GND potential
V_{CC}	28	+5V power supply
PROG	3	Output strobe for 8243 I/O Expander
P00-P07 Port 0	4-11	8-bit quasi-bidirectional port
P10-P17 Port 1	18-25	8-bit quasi-bidirectional port
P20-P23	26-27	4-bit quasi-bidirectional port
Port 2	1-2	P20-P23 also serve as a 4-bit I/O expander bus for 8243
T1	13	Input pin testable using the JT1 and JNT1 instructions. Can be designated the timer/event counter input using the STRT CNT instruction. Also allows zero-crossover sensing of slowly moving inputs.
RESET	17	Input used to initialize the processor by clearing status flip-flops and setting program counters to zero.
ALE	12	Address Latch Enable. Signal occurring once every 30 input clocks, used as an output clock.
XTAL1	15	One side of crystal or inductor input for internal oscillator. Also input for external source. (Not TTL compatible.)
XTAL2	16	Other side of timing control element.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
Accumulator	ADD A,R_r	Add register to A	1	1	68-6F
	ADD A,@ R	Add data memory to A	1	1	60-61
	ADD A,#data	Add immediate to A	2	2	03
	ADDC A,R_r	Add register with carry	1	1	78-7F
	ADDC A,@ R	Add data memory with carry	1	1	70-71
	ADDC A,#data	Add immediate with carry	2	2	13
	ANL A,R_r	And register to A	1	1	58-5F
	ANL A,@ R	And data memory to A	1	1	50-51
	ANL A,#data	And immediate to A	2	2	53
	ORL A,R_r	Or register to A	1	1	48-4F
	ORL A,@ R	Or data memory to A	1	1	40-41
	ORL A,#data	Or immediate to A	2	2	43
	XRL A,R_r	Exclusive Or register to A	1	1	D8-DF
	XRL A,@ R	Exclusive Or data memory to A	1	1	D0-D1
	XRL A,#data	Exclusive Or immediate to A	2	2	D3
	INC A	Increment A	1	1	17
	DEC A	Decrement A	1	1	07
	CLR A	Clear A	1	1	27
	CPL A	Complement A	1	1	37
	DA A	Decimal adjust A	1	1	57
	SWAP A	Swap nibbles of A	1	1	47
	RL A	Rotate A left	1	1	E7
	RLC A	Rotate A left through carry	1	1	F7
	RR A	Rotate A right	1	1	77
	RRC A	Rotate A right through carry	1	1	67
Input/Output	IN A, P_p	Input port to A	1	2	08,09,0A
	OUTL P_p,A	Output A to port	1	2	90,39,3A
	MOVD A,P_p	Input expander port to A	1	2	0C-0F
	MOVD P_p,A	Output A to expander port	1	2	3C-3F
	ANLD P_p,A	And A to expander port	1	2	9C-9F
	ORLD P_p,A	Or A to expander port	1	2	8C-8F
Registers	INC R_r	Increment register	1	1	18-1F
	INC @ R	Increment data memory	1	1	10-11
Branch	JMP addr	Jump unconditional	2	2	04,24,44,64,
	JMPP @ A	Jump indirect	1	2	B3
	DJNZ R_r, addr	Decrement register and jump on R not zero	2	2	E8-EF
	JC addr	Jump on carry=1	2	2	F6
	JNC addr	Jump on carry=0	2	2	E6
	JZ addr	Jump on A zero	2	2	C6
	JNZ addr	Jump on A not zero	2	2	96
	JT1 addr	Jump on T1=1	2	2	56
	JNT1 addr	Jump on T1=0	2	2	46
	JTF addr	Jump on timer flag	2	2	16
Subroutine	CALL addr	Jump to subroutine	1	2	14,34,54,74
	RET	Return	1	2	83
Flags	CLR C	Clear carry	1	1	97
	CPL C	Complement carry	1	1	A7
Data Moves	MOV A,R_r	Move register to A	1	1	F8-FF
	MOV A,@ R	Move data memory to A	1	1	F0-F1
	MOV A,#data	Move immediate to A	2	2	23
	MOV R_r,A	Move A to register	1	1	A8-AF
	MOV @ R,A	Move A to data memory	1	1	A0-A1
	MOV R_r,#data	Move immediate to register	2	2	B8-BF
	MOV@R,#data	Move immediate to data memory	2	2	B0-B1
	XCH A,R_r	Exchange A and register	1	1	28-2F
	XCH A,@ R	Exchange A and data memory	1	1	20-21
	XCHD A,@ R	Exchange nibble of A and register	1	1	30-31
	MOVP A,@ A	Move to A from current page	1	2	A3
Timer/Counter	MOV A,T	Read timer/counter	1	1	42
	MOV T,A	Load timer/counter	1	1	62
	STRT T	Start timer	1	1	55
	STRT CNT	Start counter	1	1	45
	STOP TCNT	Stop timer/counter	1	1	65
	NOP	No operation	1	1	00

8022

SINGLE COMPONENT 8-BIT MICROCOMPUTER WITH ON-CHIP A/D CONVERTER

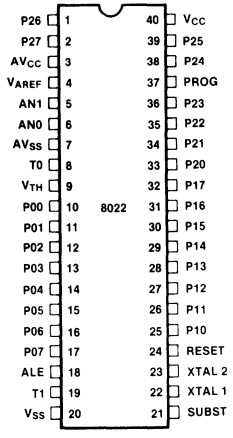
- 8-Bit CPU, ROM, RAM, I/O in Single 40-Pin Package
- On-Chip 8-Bit A/D Converter; Two Input Channels
- 8 Comparator Inputs (Port 0)
- Zero-Cross Detection Capability
- Single 5V Supply (4.5V to 6.5V)
- High Current Drive Capability—2 Pins
- Two Interrupts—External and Timer
- 2K x 8 ROM, 64 x 8 RAM, 28 I/O Lines
- 8.38 μsec Cycle; All Instructions 1 or 2 Cycles
- Instructions—8048 Subset
- Interval Timer/Event Counter
- Clock Generated with Single Inductor or Crystal
- Easily Expanded I/O

The Intel® 8022 is the newest member of the MCS-48™ family of single chip 8-bit microcomputers. It is designed to satisfy the requirements of low cost, high volume applications which involve analog signals, capacitive touchpanel keyboards, and/or large ROM space. The 8022 addresses these applications by integrating many new functions on-chip, such as A/D conversion, comparator inputs and zero-cross detection.

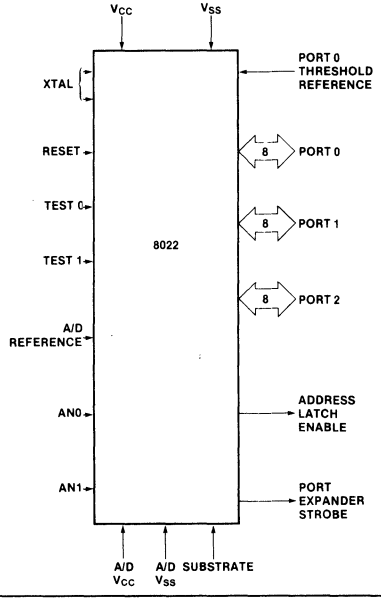
The features of the 8022 include 2K bytes of program memory (ROM), 64 bytes of data memory (RAM), 28 I/O lines, an on-chip A/D converter with two input channels, an 8-bit port with comparator inputs for interfacing to low voltage capacitive touchpanels or other non-TTL interfaces, external and timer interrupts, and zero-cross detection capability. In addition, it contains the 8-bit interval timer/event counter, on-board oscillator and clock circuitry, single 5V power supply requirement, and easily expandable I/O structure common to all members of the MCS-48 family.

The 8022 is designed to be an efficient controller as well as an arithmetic processor. It has bit handling capability plus facilities for both binary and BCD arithmetic. Efficient use of program memory results from using the MCS-48 instruction set which consists mostly of single byte instructions and has extensive conditional jump and direct table lookup capability. Program memory usage is further reduced via the 8022's hardware implementation of the A/D converter which simplifies interfacing to analog signals.

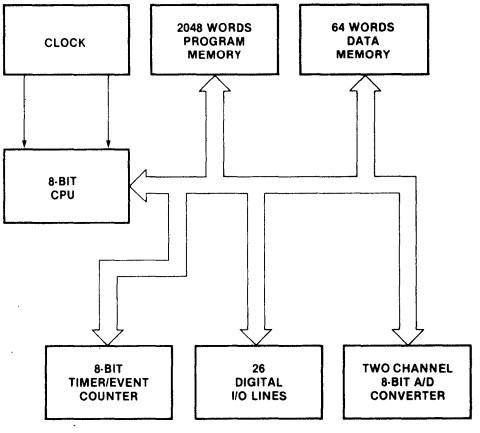
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM

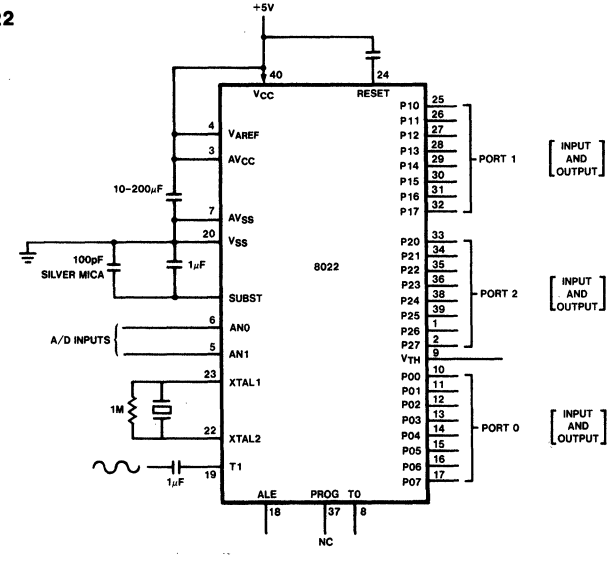


PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential.
V_{CC}	40	+ 5V circuit power supply.
PROG	37	Output strobe for Intel® 8243 I/O expander.
P00-P07 Port 0	10-17	8-bit open-drain port with comparator inputs. The switching threshold is set externally by V_{TH}. Optional pull-up resistors may be added via ROM mask selection.
V_{TH}	9	Port 0 threshold reference pin.
P10-P17 Port 1	25-32	8-bit quasi-bidirectional port.
P20-P27	33-36	8-bit quasi-bidirectional port.
Port 2	38-39 1-2	P20-23 also serve as a 4-bit I/O expander for Intel® 8243.
T0	8	Interrupt input and input pin testable using the conditional transfer instructions JT0 and JNT0. Initiates an interrupt following a low level input if interrupt is enabled. Interrupt is disabled after a reset.
T1	19	Input pin testable using the JT1 and JNT1 conditional transfer instructions. Can be designated the timer/event counter input using the STRT CNT instruction. Also serves as the zero-cross detection input to allow zero-crossover sensing of slowly moving AC inputs. Optional pull-up resistor may be added via ROM mask selection.

Designation	Pin #	Function
RESET	24	Input used to initialize the processor by clearing status flip-flops and setting the program counter to zero.
AV_{SS}	7	A/D converter GND Potential. Also establishes the lower limit of the conversion range.
AV_{CC}	3	A/D + 5V power supply.
SUBST	21	Substrate pin used with a bypass capacitor to stabilize the substrate voltage and improve A/D accuracy.
V_{AREF}	4	A/D converter reference voltage. Establishes the upper limit of the conversion range.
AN0, AN1	6,5	Analog inputs to A/D converter. Software selectable on-chip via SEL AN0 and SEL AN1 instructions.
ALE	18	Address Latch Enable. Signal occurring once every 30 input clocks (once every cycle), used as an output clock.
XTAL 1	22	One side of crystal or inductor input for internal oscillator. Also input for external frequency source. (Not TTL compatible.)
XTAL 2	23	Other side of timing control element. This pin is not connected when an external frequency source is used.

THE STAND ALONE 8022



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +180°C
Voltage on Any Pin with
Respect to Ground .....−0.5V to +7V
Power Dissipation .....1 Watt

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

D.C. AND OPERATING CHARACTERISTICS

T_A=0°C to 70°C, V_{CC}=5.5V ± 1V, V_{SS}=0V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	−0.5		0.8	V	V_{TH} Floating
V_{IL1}	Input Low Voltage (Port 0)	−0.5		V_{TH}−0.1	V	
V_{IH}	High Voltage (All except XTAL 1, RESET)	2.0		V_{CC}	V	V_{CC} = 5.0V ± 10% V_{TH} Floating
V_{IH1}	Input High Voltage (All except XTAL 1, RESET)	3.0		V_{CC}	V	V_{CC} = 5.5V ± 1V V_{TH} Floating
V_{IH2}	Input High Voltage (Port 0)	V_{TH}+0.1		V_{CC}	V	
V_{IH3}	Input High Voltage (RESET, XTAL 1)	3.0		V_{CC}	V	V_{CC} = 5.0V ± 10%
V_{TH}	Port 0 Threshold Reference Voltage	0		.4V_{CC}	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.6 mA
V_{OL1}	Output Low Voltage (P10, P11)			2.5	V	I_{OL} = 7 mA
V_{OH}	Output High Voltage (All unless Open Drain Option—Port 0)	2.4			V	I_{OH} = 50 μA
I_{LI}	Input Current (T1)			±200	μA	$V_{CC} \geq V_{IN} \geq V_{SS}$+.45V
I_{LO}	Output Leakage Current (Open Drain Option—Port 0)			± 10	μA	$V_{CC} \geq V_{IN} \geq V_{SS}$+0.45V
I_{CC}	V_{CC} Supply Current		50	100	mA	

A.C. CHARACTERISTICS

T_A=0°C to 70°C, V_{CC}=5.5V ± 1V, V_{SS}=0V

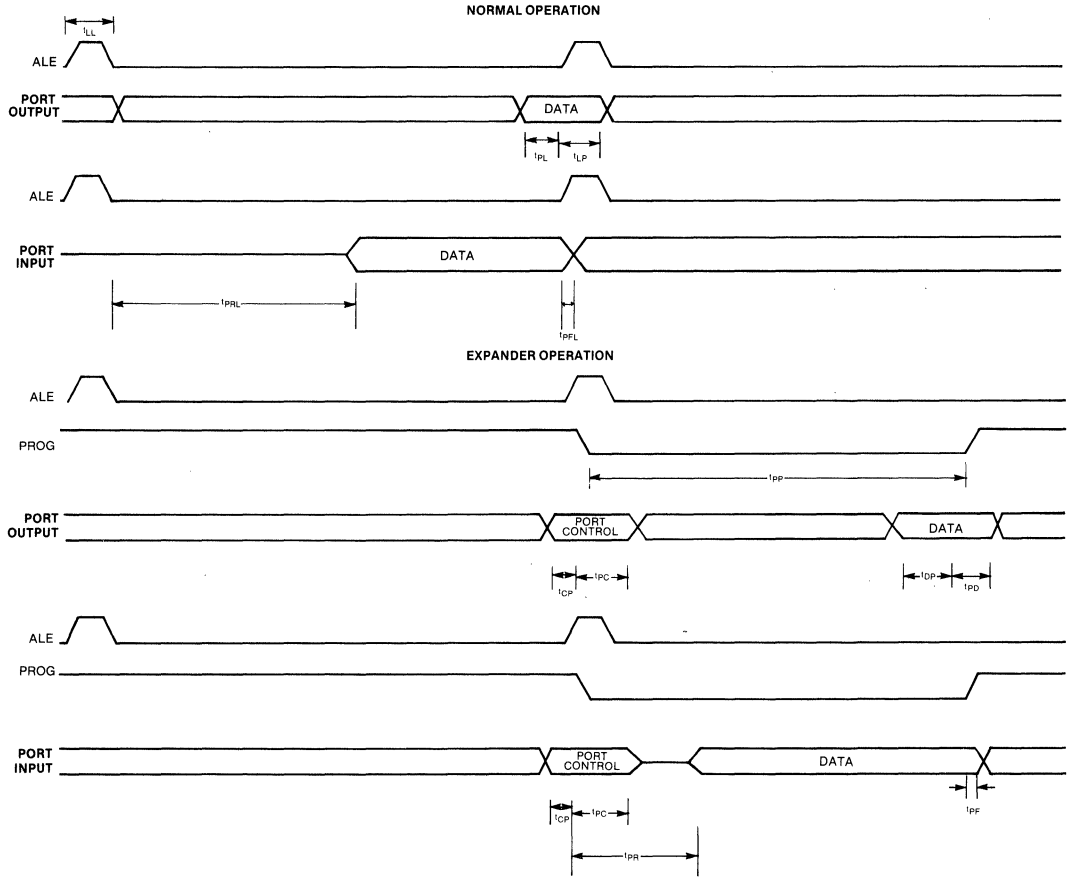
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Cycle Time	8.38	50.0	μs	3 MHz XTAL=10 μs t_{CY}
V_{ZX}	Zero-Cross Detection Input (T1)	1	3	VACpp	AC Coupled
A_{ZX}	Zero-Cross Accuracy		± 135	mV	60 Hz Sine Wave
F_{ZX}	Zero-Cross Detection Input Frequency (T1)	0.05	1	kHz	

A.C. CHARACTERISTICS

T_A=0°C to 70°C, V_{CC}=5.5V ± 1V, V_{SS}=0V Test Conditions: C_L=80 pF t_{CY}=8.38 μs

	Symbol	Parameter	Min.	Max.	Unit	Notes
Expander Operation	t_{CP}	Port Control Setup Before Falling Edge of PROG	0.5		μs	
	t_{PC}	Port Control Hold After Falling Edge of PROG	0.8		μs	
	t_{PR}	PROG to Time P2 Input Must Be Valid		1.0	μs	
	t_{DP}	Output Data Setup Time	7.0		μs	
	t_{PD}	Output Data Hold Time	8.3		μs	
	t_{PF}	Input Data Hold Time	0	.150	μs	
	t_{PP}	PROG Pulse Width	8.3		μs	
	t_{PRL}	ALE to Time P2 Input Must Be Valid		3.6	μs	
Normal Operation	t_{PL}	Output Data Setup Time	0.8		μs	
	t_{LP}	Output Data Hold Time	1.6		μs	
	t_{PFL}	Input Data Hold Time	0		μs	
	t_{LL}	ALE Pulse Width	3.9	23.0	μs	t_{CY}=8.38 μs for min

PORT 2 TIMING

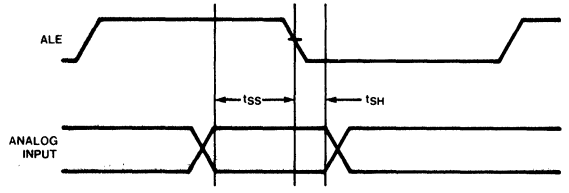


A/D CONVERTER CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$, $AV_{CC} = 5.5V \pm 1V$, $AV_{SS} = 0V$, $AV_{CC}/2 \leqslant V_{AREF} \leqslant AV_{CC}$

Parameter	Min.	Typ.	Max.	Unit	Comments
Resolution	8			Bits	
Absolute Accuracy			.8% FSR ± ½ LSB	LSB	(Note 1)
Sample Setup Before Falling Edge of ALE (t_{SS})		0.20		t_{CY}	
Sample Hold After Falling Edge of ALE (t_{SH})		0.10		t_{CY}	
Input Capacitance (AN0, AN1)		1		pF	
Conversion Time	4		4	t_{CY}	

ANALOG INPUT TIMING



NOTE:

1. The analog input must be maintained at a constant voltage during the sample time ($t_{SS} + t_{SH}$).

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
Accumulator	ADD A,R_r	Add register to A	1	1	68-6F
	ADD A,@ R	Add data memory to A	1	1	60-61
	ADD A,#data	Add immediate to A	2	2	03
	ADDC A,R_r	Add register with carry	1	1	78-7F
	ADDC A,@ R	Add data memory with carry	1	1	70-71
	ADDC A,#data	Add immediate with carry	2	2	13
	ANL A,R_r	And register to A	1	1	58-5F
	ANL A,@ R	And data memory to A	1	1	50-51
	ANL A,#data	And immediate to A	2	2	53
	ORL A,R_r	Or register to A	1	1	48-4F
	ORL A,@ R	Or data memory to A	1	1	40-41
	ORL A,#data	Or immediate to A	2	2	43
	XRL A,R_r	Exclusive Or register to A	1	1	D8-DF
	XRL A,@ R	Exclusive Or data memory to A	1	1	D0-D1
	XRL A,#data	Exclusive Or immediate to A	2	2	D3
	INC A	Increment A	1	1	17
	DEC A	Decrement A	1	1	07
	CLR A	Clear A	1	1	27
	CPL A	Complement A	1	1	37
	DA A	Decimal adjust A	1	1	57
	SWAP A	Swap nibbles of A	1	1	47
	RL A	Rotate A left	1	1	E7
	RLC A	Rotate A left through carry	1	1	F7
	RR A	Rotate A right	1	1	77
	RRC A	Rotate A right through carry	1	1	67
Input/Output	IN A, P_p	Input port to A	1	2	08,09,0A
	OUTL P_p,A	Output A to port	1	2	90,39,3A
	MOVD A,P_p	Input expander port to A	1	2	0C-0F
	MOVD P_p,A	Output A to expander port	1	2	3C-3F
	ANLD P_p,A	And A to expander port	1	2	9C-9F
	ORLD P_p,A	Or A to expander port	1	2	8C-8F
Registers	INC R_r	Increment register	1	1	18-1F
	INC @ R	Increment data memory	1	1	10-11
Branch	JMP addr	Jump unconditional	2	2	04,24,44,64, 84,A4,C4,E4
	JMPP @ A	Jump indirect	1	2	B3
	DJNZ R,addr	Decrement register and jump on R not zero	2	2	E8-EF
	JC addr	Jump on carry=1	2	2	F6
	JNC addr	Jump on carry=0	2	2	E6
	JZ addr	Jump on A zero	2	2	C6
	JNZ addr	Jump on A not zero	2	2	96

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
	JTO addr	Jump on T0=1	2	2	36
	JNTO addr	Jump on T0=0	2	2	26
	JT1 addr	Jump on T1=1	2	2	56
	JNT1 addr	Jump on T1=0	2	2	46
	JTF addr	Jump on timer flag	2	2	16
Subroutine	CALL addr	Jump to subroutine	1	2	14,34,54,74 94,B4,D4,F4
	RET	Return	1	2	83
Flags	CLR C	Clear carry	1	1	97
	CPL C	Complement carry	1	1	A7
Data Moves	MOV A,R_r	Move register to A	1	1	F8-FF
	MOV A,@ R	Move data memory to A	1	1	F0-F1
	MOV A,#data	Move immediate to A	2	2	23
	MOV R_r,A	Move A to register	1	1	A8-AF
	MOV @ R,A	Move A to data memory	1	1	A0-A1
	MOV R_r,#data	Move immediate to register	2	2	B8-BF
	MOV@R,#data	Move immediate to data memory	2	2	B0-B1
	XCH A,R_r	Exchange A and register	1	1	28-2F
	XCH A,@ R	Exchange A and data memory	1	1	20-21
	XCHD a,@ R	Exchange nibble of A and register	1	1	30-31
	MOVP A,@ A	Move to A from current page	1	2	A3
Timer/Counter	MOV A,T	Read timer/counter	1	1	42
	MOV T,A	Load timer/counter	1	1	62
	STRT T	Start timer	1	1	55
	STRT CNT	Start counter	1	1	45
	STOP TCNT	Stop timer/counter	1	1	65
A/D Converter	RAD	Move conversion result register to A	1	2	80
	SEL AN0	Select analog input zero	1	1	85
	SEL AN1	Select analog input one	1	1	95
Interrupts	EN I	Enable external interrupt	1	1	05
	DIS I	Disable external interrupt	1	1	15
	EN TCNTI	Enable timer/counter interrupt	1	1	25
	DIS TCNTI	Disable timer/counter interrupt	1	1	35
	RET I	Return from interrupt	1	2	93
	NOP	No operation	1	1	00

SYMBOLS AND ABBREVIATIONS USED

A	Accumulator
addr	11-Bit Program Memory Address
AN0, AN1	Analog Input 0, Analog Input 1
CNT	Event Counter
data	8-Bit Number or Expression
I	Interrupt
P	Mnemonic for "in-page" Operation
P_p	Port Designator (P=0, 1, 2 or 4-7)
R_r	Register Designator (r=0-7)
T	Timer
T0, T1	Test 0, Test 1
#	Immediate Data Prefix
@	Indirect Address Prefix

8243
MCS-48™ INPUT/OUTPUT EXPANDER

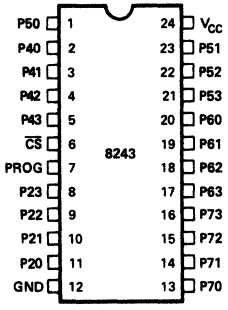
- **Low Cost**
- **Simple Interface to MCS-48™ Microcomputers**
- **Four 4-Bit I/O Ports**
- **AND and OR Directly to Ports**
- **24-Pin DIP**
- **Single 5V Supply**
- **High Output Drive**
- **Direct Extension of Resident 8048 I/O Ports**

The Intel® 8243 is an input/output expander designed specifically to provide a low cost means of I/O expansion for the MCS-48™ family of single chip microcomputers. Fabricated in 5 volts NMOS, the 8243 combines low cost, single supply voltage and high drive current capability.

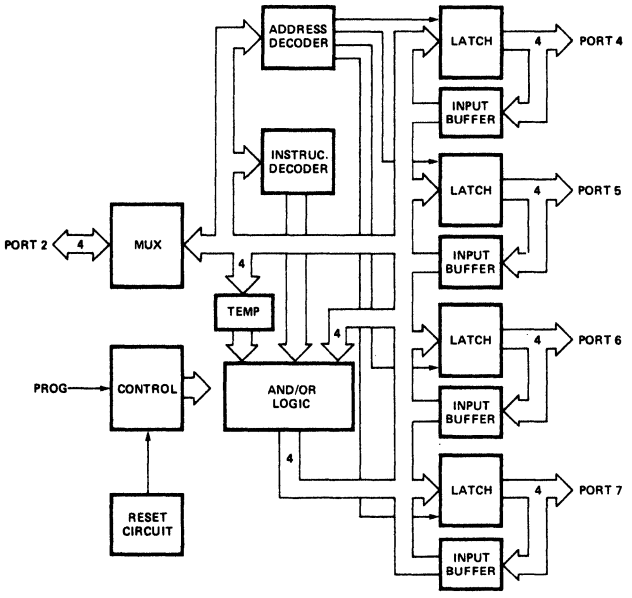
The 8243 consists of four 4-bit bidirectional static I/O ports and one 4-bit port which serves as an interface to the MCS-48 microcomputers. The 4-bit interface requires that only 4 I/O lines of the 8048 be used for I/O expansion, and also allows multiple 8243's to be added to the same bus.

The I/O ports of the 8243 serve as a direct extension of the resident I/O facilities of the MCS-48 microcomputers and are accessed by their own MOV, ANL, and ORL instructions.

PIN CONFIGURATION



BLOCK DIAGRAM



AFN-00214A-01

PIN DESCRIPTION

Symbol	Pin No.	Function
PROG	7	Clock Input. A high to low transistion on PROG signifies that address and control are available on P20-P23, and a low to high transition signifies that data is available on P20-23.
$\overline{CS}$	6	Chip Select Input. A high on CS inhibits any change of output or internal status.
P20-P23	11-8	Four (4) bit bi-directional port contains the address and control bits on a high to low transition of PROG. During a low to high transition contains the data for a selected output port if a write operation, or the data from a selected port before the low to high transition if a read operation.
GND	12	0 volt supply.
P40-P43 P50-P53 P60-P63 P70-P73	2-5 1,23-21 20-17 13-16	Four (4) bit bi-directional I/O ports. May be programmed to be input (during read), low impedance latched output (after write) or a tri-state (after read). Data on pins P20-23 may be directly written, ANDed or ORed with previous data.
V_{CC}	24	+5 volt supply.

FUNCTIONAL DESCRIPTION

General Operation

The 8243 contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports 4-7. The following operations may be performed on these ports:

- Transfer Accumulator to Port.
- Transfer Port to Accumulator.
- AND Accumulator to Port.
- OR Accumulator to Port.

All communication between the 8048 and the 8243 occurs over Port 2 (P20-P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

The first containing the "op code" and port address and the second containing the actual 4-bits of data.

A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the 4-bit bus and chip selected using additional output lines from the 8048/8748/8035.

Power On Initialization

Initial application of power to the device forces input/output ports 4, 5, 6, and 7 to the tri-state and port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi,A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi,A takes new data AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are tri-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the tri-stated mode while port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the 8243 output. A read of any port will leave that port in a high impedance state.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... -65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... -0.5 V to +7V
Power Dissipation ..... 1 Watt

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ±10%

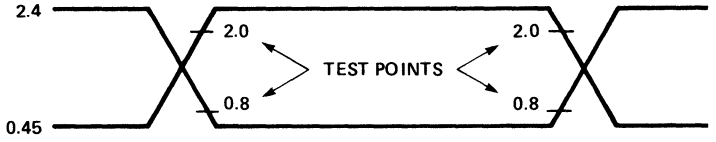
SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage Ports 4-7			0.45	V	I_{OL} = 4.5 mA*
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4-7	2.4			V	I_{OH} = 240μA
I_{IL1}	Input Leakage Ports 4-7	-10		20	μA	V_{in} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, CS, PROG	-10		10	μA	V_{in} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			.45	V	I_{OL} = 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	
V_{OH2}	Output Voltage Port 2	2.4				I_{OH} = 100μA
I_{OL}	Sum of all IO_L from 16 Outputs			72	mA	4.5 mA Each Pin

*See following graph for additional sink current capability

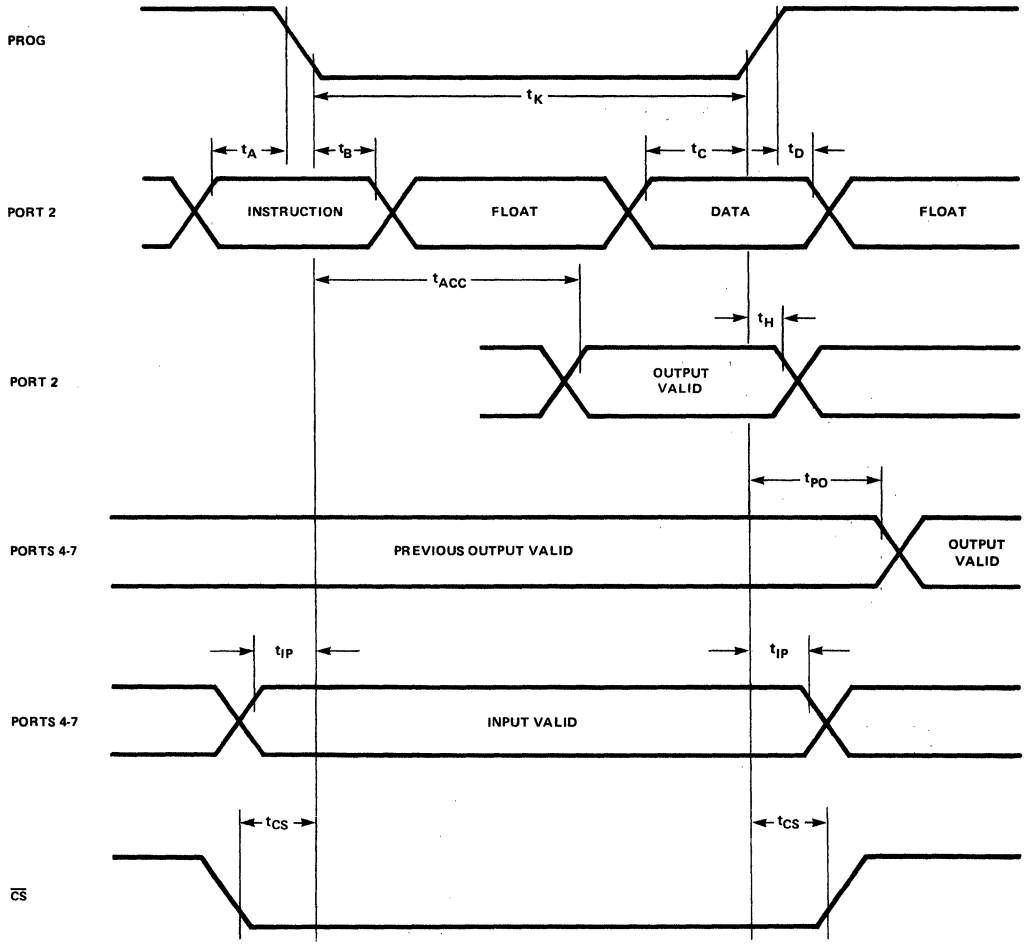
A.C. CHARACTERISTICS

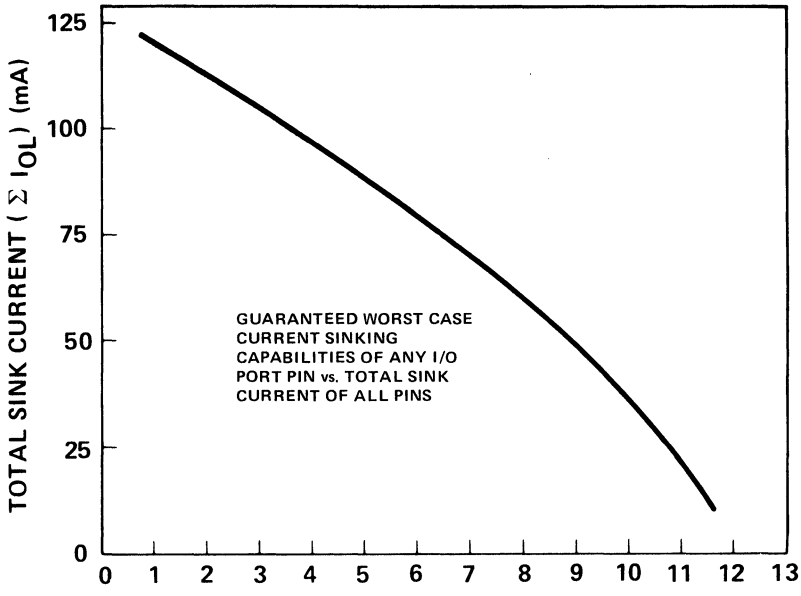
T_A = 0°C to 70°C, V_{CC} = 5V ±10%

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_A	Code Valid Before PROG	100		ns	80 pF Load
t_B	Code Valid After PROG	60		ns	20 pF Load
t_C	Data Valid Before PROG	200		ns	80 pF Load
t_D	Data Valid After PROG	20		ns	20 pF Load
t_H	Floating After PROG	0	150	ns	20 pF Load
t_K	PROG Negative Pulse Width	700		ns	
t_{CS}	CS Valid Before/After PROG	50		ns	
t_{PO}	Ports 4-7 Valid After PROG		700	ns	100 pF Load
t_{LP1}	Ports 4-7 Valid Before/After PROG	100		ns	
t_{ACC}	Port 2 Valid After PROG		650	ns	80 pF Load



WAVEFORMS


PROG
t_K
t_A
t_B
t_C
t_D
PORT 2
INSTRUCTION
FLOAT
DATA
FLOAT
t_{ACC}
t_H
PORT 2
OUTPUT VALID
t_{PO}
PORTS 4-7
PREVIOUS OUTPUT VALID
OUTPUT VALID
t_{IP}
t_{IP}
PORTS 4-7
INPUT VALID
t_{CS}
t_{CS}
$\overline{CS}$



Sink Capability

The 8243 can sink 5 mA@.45V on each of its 16 I/O lines simultaneously. If, however, all lines are not sinking simultaneously or all lines are not fully loaded, the drive capability of any individual line increases as is shown by the accompanying curve.

For example, if only 5 of the 16 lines are to sink current at one time, the curve shows that each of those 5 lines is capable of sinking 9 mA@.45V (if any lines are to sink 9 mA the total I_{OL} must not exceed 45 mA or five 9 mA loads).

Example: How many pins can drive 5 TTL loads (1.6 mA) assuming remaining pins are unloaded?

$I_{OL} = 5 \times 1.6\ mA = 8\ mA$
$\varepsilon I_{OL} = 60$ mA from curve
pins = 60 mA ÷ 8 mA/pin = 7.5 = 7

In this case, 7 lines can sink 8 mA for a total of 56 mA. This leaves 4 mA sink current capability which can be divided in any way among the remaining 8 I/O lines of the 8243.

Example: This example shows how the use of the 20 mA sink capability of Port 7 affects the sinking capability of the other I/O lines.

An 8243 will drive the following loads simultaneously.

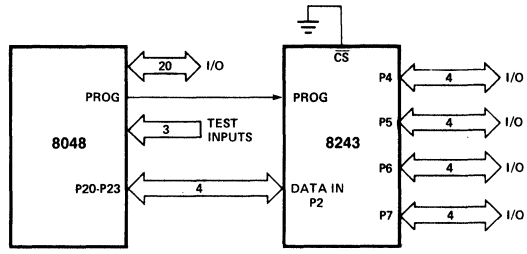
2 loads — 20 mA@1V (port 7 only)
8 loads — 4 mA@.45V
6 loads — 3.2 mA@.45V
Is this within the specified limits?

$\varepsilon I_{OL} = (2 \times 20) + (8 \times 4) + (6 \times 3.2) = 91.2$ mA.
From the curve: for $I_{OL} = 4$ mA, $\varepsilon I_{OL} \approx 93$ mA
since 91.2 mA < 93 mA the loads are within specified limits.

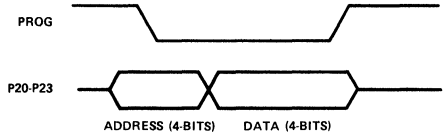
Although the 20 mA@1V loads are used in calculating εI_{OL}, it is the largest current required@.45V which determines the maximum allowable εI_{OL}.

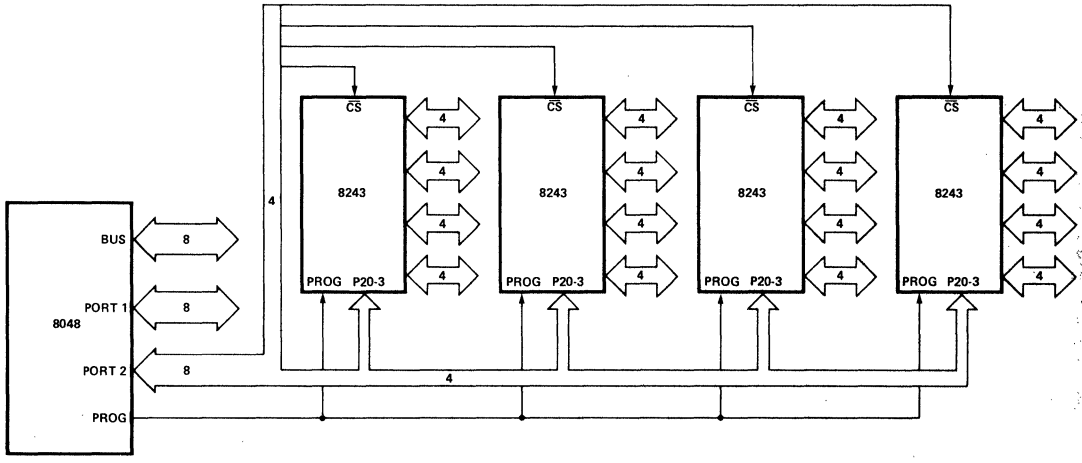
Note: A 10 to 50KΩ pullup resistor to +5V should be added to 8243 outputs when driving to 5V CMOS directly.

EXPANDER INTERFACE



OUTPUT EXPANDER TIMING





USING MULTIPLE 8243's

ID8243
MCS-48™ INPUT/OUTPUT EXPANDER

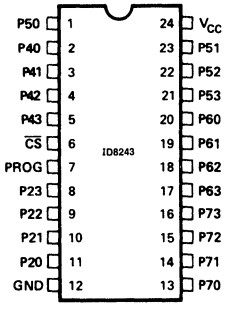
- **−40°C to +85°C Operation**
- **Low Cost**
- **Simple Interface to MCS-48™ Microcomputers**
- **Four 4-Bit I/O Ports**
- **AND and OR Directly to Ports**
- **24-Pin DIP**
- **Single 5V Supply**
- **High Output Drive**
- **Direct Extension of Resident 8048 I/O Ports**

The Intel® 8243 is an input/output expander designed specifically to provide a low cost means of I/O expansion for the MCS-48™ family of single chip microcomputers. Fabricated in 5 volts NMOS, the 8243 combines low cost, single supply voltage and high drive current capability.

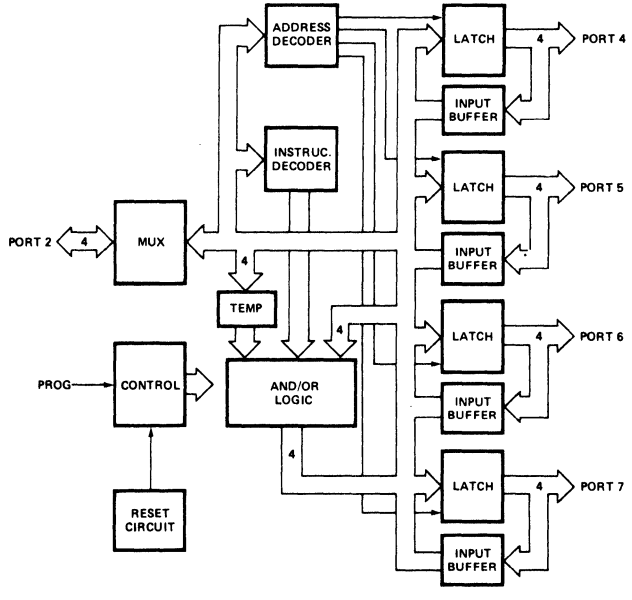
The 8243 consists of four 4-bit bidirectional static I/O ports and one 4-bit port which serves as an interface to the MCS-48 microcomputers. The 4-bit interface requires that only 4 I/O lines of the 8048 be used for I/O expansion, and also allows multiple 8243's to be added to the same bus.

The I/O ports of the 8243 serve as a direct extension of the resident I/O facilities of the MCS-48 microcomputers and are accessed by their own MOV, ANL, and ORL instructions.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN DESCRIPTION

Symbol	Pin No.	Function
PROG	7	Clock Input. A high to low transistion on PROG signifies that address and control are available on P20-P23, and a low to high transition signifies that data is available on P20-23.
$\overline{CS}$	6	Chip Select Input. A high on CS inhibits any change of output or internal status.
P20-P23	11-8	Four (4) bit bi-directional port contains the address and control bits on a high to low transition of PROG. During a low to high transition contains the data for a selected output port if a write operation, or the data from a selected port before the low to high transition if a read operation.
GND	12	0 volt supply.
P40-P43 P50-P53 P60-P63 P70-P73	2-5 1,23-21 20-17 13-16	Four (4) bit bi-directional I/O ports. May be programmed to be input (during read), low impedance latched output (after write) or a tri-state (after read). Data on pins P20-23 may be directly written, ANDed or ORed with previous data.
V_{CC}	24	+5 volt supply.

FUNCTIONAL DESCRIPTION

General Operation

The 8243 contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports 4-7. The following operations may be performed on these ports:

- Transfer Accumulator to Port.
- Transfer Port to Accumulator.
- AND Accumulator to Port.
- OR Accumulator to Port.

All communication between the 8048 and the 8243 occurs over Port 2 (P20-P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

The first containing the "op code" and port address and the second containing the actual 4-bits of data.

A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the 4-bit bus and chip selected using additional output lines from the 8048/8748/8035.

Power On Initialization

Initial application of power to the device forces input/output ports 4, 5, 6, and 7 to the tri-state and port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi,A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi,A takes new data AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are tri-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the tri-stated mode while port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the 8243 output. A read of any port will leave that port in a high impedance state.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....-40°C to +85°C
Storage Temperature..... -65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... -0.5 V to +7V
Power Dissipation .....1 Watt

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

D.C. AND OPERATING CHARACTERISTICS

T_A = -40°C to +85°C, V_{CC}= 5V ±10%

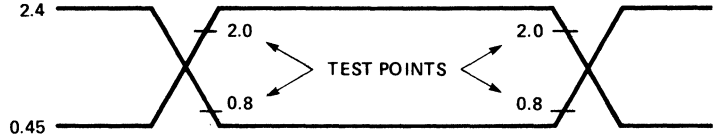
SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage Ports 4-7			0.45	V	I_{OL} = 4.5 mA*
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4-7	2.4			V	I_{OH}= 240μA
I_{IL1}	Input Leakage Ports 4-7	-10		20	μA	V_{in} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, CS, PROG	-10		10	μA	V_{in} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			.45	V	I_{OL}= 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	
V_{OH2}	Output Voltage Port 2	2.4				I_{OH}= 100μA
I_{OL}	Sum of all IO_L from 16 Outputs			80	mA	4.5 mA Each Pin

*See following graph for additional sink current capability

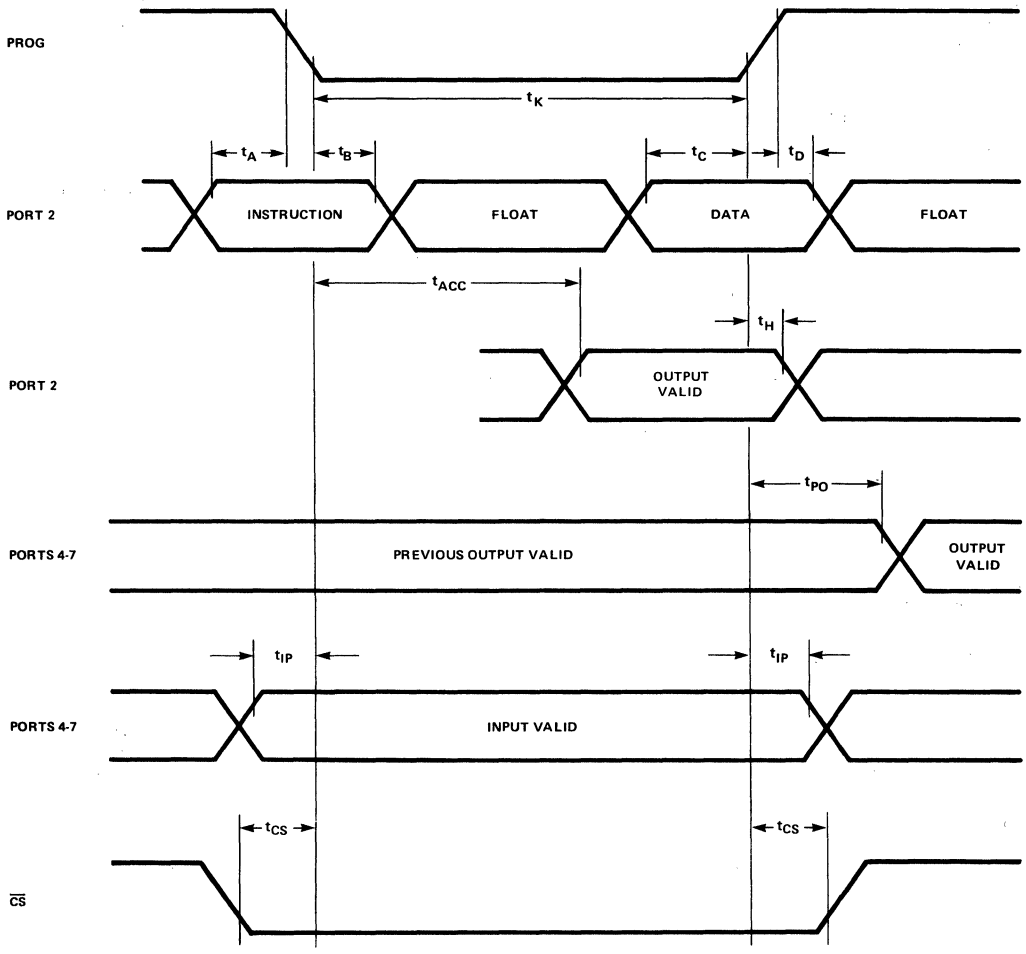
A.C. CHARACTERISTICS

T_A = -40°C to +85°C, V_{CC}= 5V ±10%

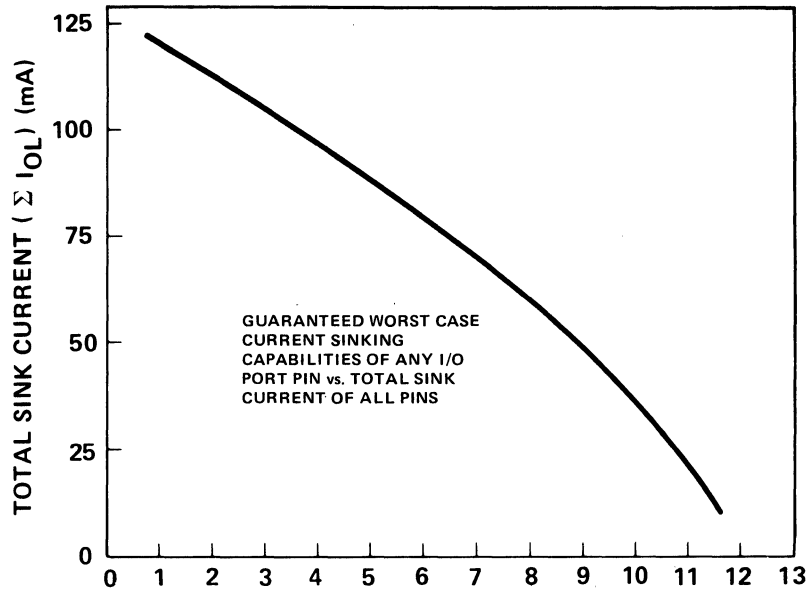
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_A	Code Valid Before PROG	100		ns	80 pF Load
t_B	Code Valid After PROG	60		ns	20 pF Load
t_C	Data Valid Before PROG	200		ns	80 pF Load
t_D	Data Valid After PROG	20		ns	20 pF Load
t_H	Floating After PROG	0	150	ns	20 pF Load
t_K	PROG Negative Pulse Width	700		ns	
t_{CS}	CS Valid Before/After PROG	50		ns	
t_{PO}	Ports 4-7 Valid After PROG		700	ns	100 pF Load
t_{LP1}	Ports 4-7 Valid Before/After PROG	100		ns	
t_{ACC}	Port 2 Valid After PROG		650	ns	80 pF Load



WAVEFORMS



AFN-00861A-04



Sink Capability

The 8243 can sink 5 mA@.45V on each of its 16 I/O lines simultaneously. If, however, all lines are not sinking simultaneously or all lines are not fully loaded, the drive capability of any individual line increases as is shown by the accompanying curve.

For example, if only 5 of the 16 lines are to sink current at one time, the curve shows that each of those 5 lines is capable of sinking 9 mA@.45V (if any lines are to sink 9 mA the total I_{OL} must not exceed 45 mA or five 9 mA loads).

Example: How many pins can drive 5 TTL loads (1.6 mA) assuming remaining pins are unloaded?

$I_{OL} = 5 \times 1.6\ mA = 8\ mA$
$\varepsilon I_{OL} = 60$ mA from curve
pins = 60 mA ÷ 8 mA/pin = 7.5 = 7

In this case, 7 lines can sink 8 mA for a total of 56 mA. This leaves 4 mA sink current capability which can be divided in any way among the remaining 8 I/O lines of the 8243.

Example: This example shows how the use of the 20 mA sink capability of Port 7 affects the sinking capability of the other I/O lines.

An 8243 will drive the following loads simultaneously.

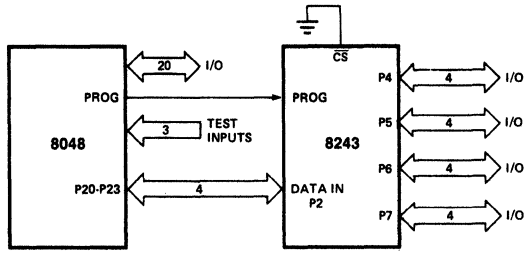
2 loads — 20 mA@1V (port 7 only)
8 loads — 4 mA@.45V
6 loads — 3.2 mA@.45V
Is this within the specified limits?

$\varepsilon I_{OL} = (2 \times 20) + (8 \times 4) + (6 \times 3.2) = 91.2$ mA.
From the curve: for $I_{OL} = 4$ mA, $\varepsilon I_{OL} \approx 93$ mA since 91.2 mA < 93 mA the loads are within specified limits.

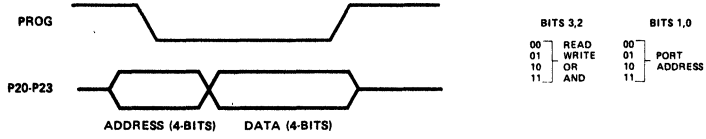
Although the 20 mA@1V loads are used in calculating εI_{OL}, it is the largest current required@.45V which determines the maximum allowable εI_{OL}.

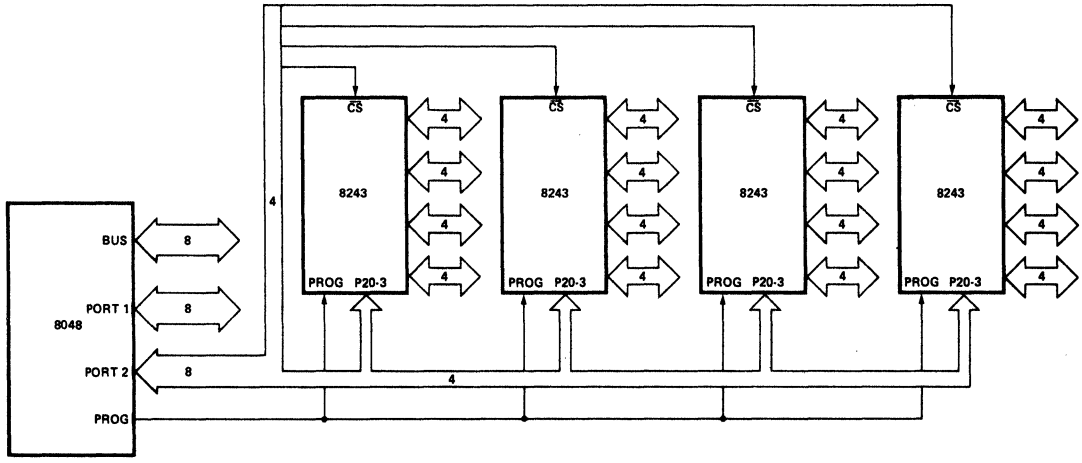
Note: A 10 to 50KΩ pullup resistor to +5V should be added to 8243 outputs when driving to 5V CMOS directly.

EXPANDER INTERFACE



OUTPUT EXPANDER TIMING





USING MULTIPLE 8243's

8355*/8355-2**

16,384-BIT ROM WITH I/O

*Directly Compatible with 8085A CPU

**Directly Compatible with 8085A-2

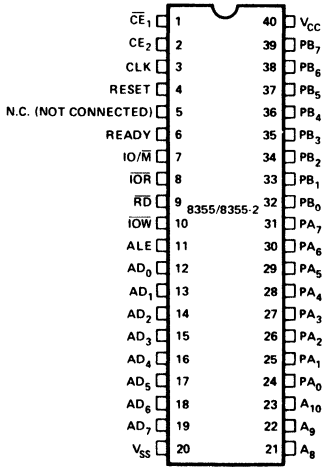
- 2048 Words x 8 Bits
- Single +5V Power Supply
- Internal Address Latch
- 2 General Purpose 8-Bit I/O Ports
- Each I/O Port Line Individually Programmable as Input or Output
- Multiplexed Address and Data Bus
- 40-Pin DIP

The Intel® 8355 is a ROM and I/O chip to be used in the MCS-85™ microcomputer system. The ROM portion is organized as 2048 words by 8 bits. It has a maximum access time of 400 ns to permit use with no wait states in the 8085A CPU.

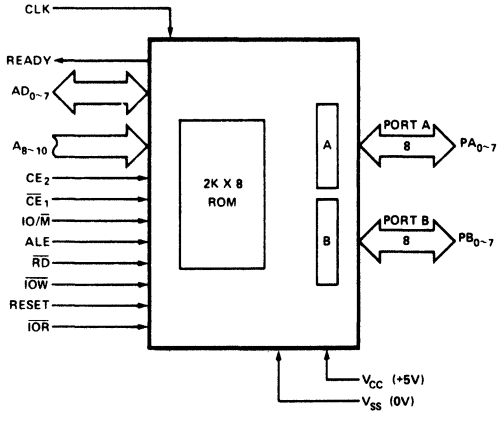
The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is indivdually programmable as input or output.

The 8355-2 has a 300ns access time for compatibility with the 8085A-2 microprocessor.

PIN CONFIGURATION



BLOCK DIAGRAM



Symbol	Function
ALE (Input)	When ALE (Address Latch Enable is high, AD_{0-7}, IO/$\overline{M}$, A_{8-10}, CE, and $\overline{CE}$ enter address latched. The signals (AD, IO/$\overline{M}$, A_{8-10}, CE, $\overline{CE}$) are latched in at the trailing edge of ALE.
AD_{0-7} (Input)	Bidirectional Address/Data bus. The lower 8-bits of the ROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B are selected based on the latched value of AD_0. If $\overline{RD}$ or $\overline{IOR}$ is low when the latched chip enables are active, the output buffers present data on the bus.
A_{8-10} (Input)	These are the high order bits of the ROM address. They do not affect I/O operations.
$\overline{CE_1}$ CE_2 (Input)	Chip Enable Inputs: $\overline{CE_1}$ is active low and CE_2 is active high. The 8355 can be accessed only when BOTH Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD_{0-7} and READY outputs will be in a high impedance state.
IO/$\overline{M}$ (Input)	If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the ROM.
$\overline{RD}$ (Input)	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected ROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are 3-state.
$\overline{IOW}$ (Input)	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}. The state of IO/$\overline{M}$ is ignored.

Symbol	Function
CLK (Input)	The CLK is used to force the READY into its high impedance state after it has been forced low by $\overline{CE}$ low, CE high and ALE high.
READY (Output)	Ready is a 3-state output controlled by $\overline{CE_1}$, CE_2, ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK (see Figure 6).
PA_{0-7} (Input/ Output)	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD_0. Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD_0 low, or IO/$\overline{M}$ high, $\overline{RD}$ low, active chip enables, and AD_0 low.
PB_{0-7} (Input/ Output)	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0.
RESET (Input)	An input high on RESET causes all pins in Port A and B to assume input mode.
$\overline{IOR}$ (Input)	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination IO/$\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V_{CC} ("1").
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... -0.5V to +7V
Power Dissipation 1.5W

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	V_{CC} = 5.0V
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤V_{OUT} ≤V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

		8355		8355-2 (Preliminary)		
Symbol	Parameter	Min.	Max.	Min.	Max.	Units
t_{CYC}	Clock Cycle Time	320		320		ns
T_1	CLK Pulse Width	80		80		ns
T_2	CLK Pulse Width	120		120		ns
t_f,t_r	CLK Rise and Fall Time		30		30	ns
t_{AL}	Address to Latch Set Up Time	50		30		ns
t_{LA}	Address Hold Time after Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{AD}	Address Stable to Data Out Valid		400		330	ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float after READ	0	100	0	85	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data In to Write Set Up Time	150		150		ns
t_{WD}	Data In Hold Time After WRITE	10		10		ns
t_{WP}	WRITE to Port Output		400		400	ns
t_{PR}	Port Input Set Up Time	50		50		ns
t_{RP}	Port Input Hold Time	50		50		ns
t_{RYH}	READY HOLD Time	0	160	0	160	ns
t_{ARY}	ADDRESS (CE) to READY		160		160	ns
t_{RV}	Recovery Time Between Controls	300		200		ns
t_{RDE}	READ Control to Data Bus Enable	10		10		ns

Note: C_{LOAD} = 150pF

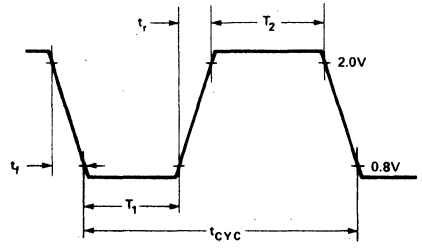


Figure 1. Clock Specification for 8355

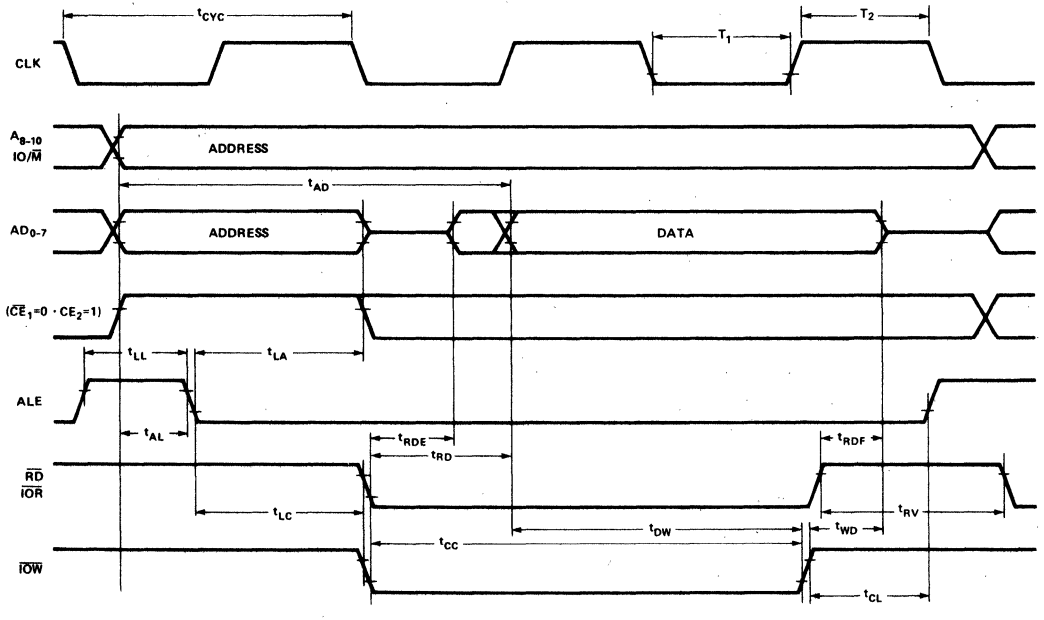
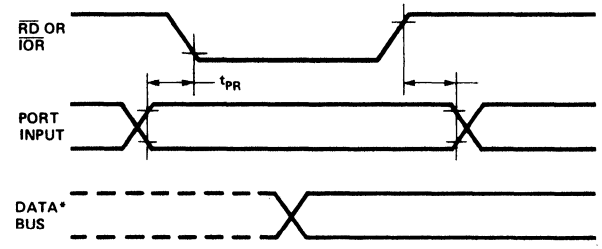
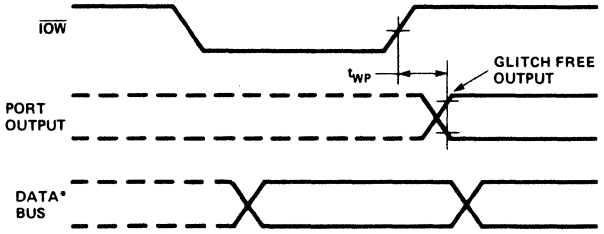


Figure 2. ROM Read and I/O Read and Write

a. Input Mode



b. Output Mode



*DATA BUS TIMING IS SHOWN IN FIGURE 4.

Figure 3. I/O Port Timing

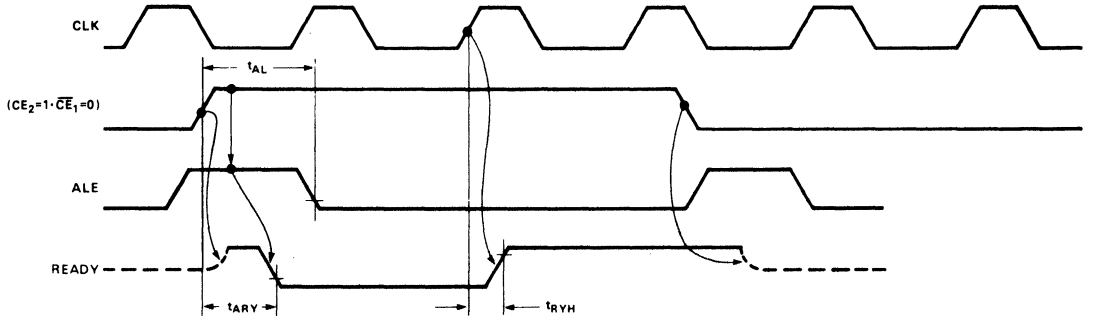


Figure 4. Wait State Timing (READY = 0)

8755A
16,384-BIT EPROM WITH I/O

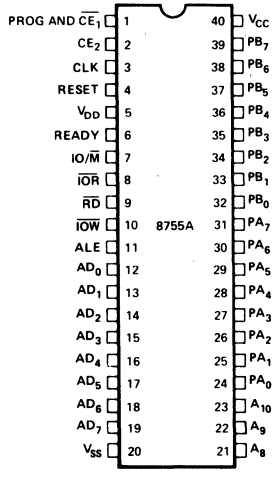
- **Directly Compatible with 8085A CPU**

- **2048 Words x 8 Bits**
- **Single +5V Power Supply (V_{CC})**
- **U.V. Erasable and Electrically Reprogrammable**
- **Internal Address Latch**
- **2 General Purpose 8-Bit I/O Ports**
- **Each I/O Port Line Individually Programmable as Input or Output**
- **Multiplexed Address and Data Bus**
- **40-Pin DIP**

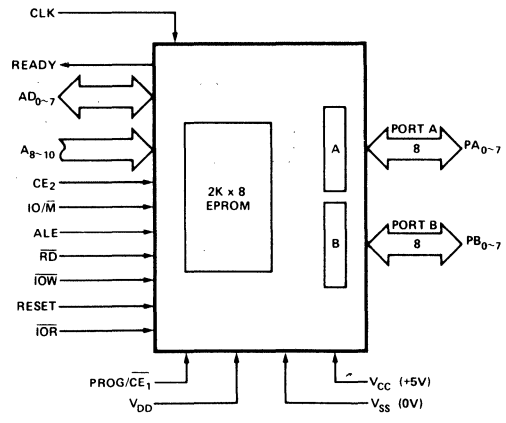
The Intel® 8755A is an erasable and electrically reprogrammable ROM (EPROM) and I/O chip to be used in the MCS-85™ microcomputer system. The EPROM portion is organized as 2048 words by 8 bits. It has a maximum access time of 450 ns to permit use with no wait states in an 8085A CPU.

The I/O portion consists of 2 general purpose I/O ports. Each I/O port has 8 port lines, and each I/O port line is individually programmable as input or output.

PIN CONFIGURATION



BLOCK DIAGRAM



8755A FUNCTIONAL PIN DEFINITION

Symbol	Function
ALE (input)	When Address Latch Enable goes high, AD_{0-7}, IO/M, A_{8-10}, CE_2, and $\overline{CE_1}$ enter the address latches. The signals (AD, IO/M, A_{8-10}, CE) are latched in at the trailing edge of ALE.
AD_{0-7} (input/output)	Bidirectional Address/Data bus. The lower 8-bits of the PROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B are selected based on the latched value of AD_0. If $\overline{RD}$ or $\overline{IOR}$ is low when the latched Chip Enables are active, the output buffers present data on the bus.
A_{8-10} (input)	These are the high order bits of the PROM address. They do not affect I/O operations.
PROG/$\overline{CE_1}$ CE_2 (input)	Chip Enable Inputs: $\overline{CE_1}$ is active low and CE_2 is active high. The 8755A can be accessed only when *BOTH* Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD_{0-7} and READY outputs will be in a high impedance state. $\overline{CE_1}$ is also used as a programming pin. (See section on programming.)
IO/$\overline{M}$ (input)	If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the PROM.
$\overline{RD}$ (input)	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected PROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are 3-stated.
$\overline{IOW}$ (input)	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}. The state of IO/$\overline{M}$ is ignored.
CLK (input)	The CLK is used to force the READY into its high impedance state after it has been forced low by $\overline{CE_1}$ low, CE_2 high, and ALE high.

Symbol	Function
READY (output)	READY is a 3-state output controlled by CE_2, $\overline{CE_1}$, ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK. (See Figure 6.)
PA_{0-7} (input/output)	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD_0, AD_1. Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD_0 and AD_1 low, *or* IO/$\overline{M}$ high, $\overline{RD}$ low, active Chip Enables, and AD_0 and AD_1 low.
PB_{0-7} (input/output)	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0 and a 0 from AD_1.
RESET (input)	In normal operation, an input high on RESET causes all pins in Ports A and B to assume input mode (clear DDR register).
$\overline{IOR}$ (input)	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination of IO/$\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V_{CC} ("1").
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.
V_{DD}	V_{DD} is a programming voltage, and must be tied to +5V when the 8755A is being read. For programming, a high voltage is supplied with V_{DD} = 25V, typical. (See section on programming.)

ERASURE CHARACTERISTICS

The erasure characteristics of the 8755A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8755A in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8755A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8755 window to prevent unintentional erasure.

The recommended erasure procedure for the 8755A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8755A should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

PROGRAMMING

Initially, and after each erasure, all bits of the EPROM portions of the 8755A are in the "1" state. Information is introduced by selectively programming "0" into the desired bit locations. A programmed "0" can only be changed to a "1" by UV erasure.

The 8755A can be programmed on the Intel® Universal PROM Programmer (UPP), and the PROMPT™ 80/85 and PROMPT-48™ design aids. The appropriate programming modules and adapters for use in programming both 8755A's and 8755's are shown in Table 1.

The program mode itself consists of programming a single address at a time, giving a single 50 msec pulse for every address. Generally, it is desirable to have a verify cycle after a program cycle for the same address as shown in the attached timing diagram. In the verify cycle (i.e., normal memory read cycle) 'V_{DD}' should be at +5V.

Preliminary timing diagrams and parameter values pertaining to the 8755A programming operation are contained in Figure 6.

TABLE 1. 8755A PROGRAMMING MODULE CROSS REFERENCE

MODULE NAME	USE WITH
UPP 955	UPP(4)
UPP UP2(2)	UPP 855
PROMPT 975	PROMPT 80/85(3)
PROMPT 475	PROMPT 48(1)

NOTES:
1. Described on p. 11-9 of 1978 System Data Catalog.
2. Special adaptor socket.
3. Described on p. 11-3 of 1978 System Data Catalog.
4. Described on p. 10-85 of 1978 System Data Catalog.

SYSTEM APPLICATIONS

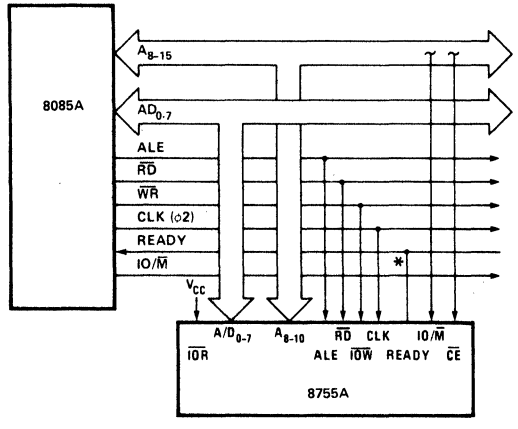
System Interface with 8085A

A system using the 8755A can use either one of the two I/O Interface techniques:

- Standard I/O
- Memory Mapped I/O

If a standard I/O technique is used, the system can use the feature of both CE_2 and $\overline{CE}_1$. By using a combination of unused address lines A_{11-15} and the Chip Enable inputs, the 8085A system can use up to 5 each 8755A's without requiring a CE decoder.

If a memory mapped I/O approach is used the 8755A will be selected by the combination of both the Chip Enables and IO/$\overline{M}$ using the AD_{8-15} address lines. See Figure 1.



*NOTE: Optional connection.

Figure 1. 8755A in 8085A System (Memory-Mapped I/O)

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... -10°C to +70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... -0.5 to +7V*
Power Dissipation 1.5W

*Except for programming voltage.

*COMMENT

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤V_{OUT} ≤V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_{CYC}	Clock Cycle Time	320		ns	
T_1	CLK Pulse Width	80		ns	C_{LOAD} = 150 pF
T_2	CLK Pulse Width	120		ns	(See Figure 3)
t_f,t_r	CLK Rise and Fall Time		30	ns	
t_{AL}	Address to Latch Set Up Time	50		ns	
t_{LA}	Address Hold Time after Latch	80		ns	
t_{LC}	Latch to READ/WRITE Control	100		ns	
t_{RD}	Valid Data Out Delay from READ Control		170	ns	
t_{AD}	Address Stable to Data Out Valid		450	ns	150 pF Load
t_{LL}	Latch Enable Width	100		ns	
t_{RDF}	Data Bus Float after READ	0	100	ns	
t_{CL}	READ/WRITE Control to Latch Enable	20		ns	
t_{CC}	READ/WRITE Control Width	250		ns	
t_{DW}	Data In to WRITE Set Up Time	150		ns	
t_{WD}	Data In Hold Time After WRITE	30		ns	
t_{WP}	WRITE to Port Output		400	ns	
t_{PR}	Port Input Set Up Time	50		ns	
t_{RP}	Port Input Hold Time	50		ns	
t_{RYH}	READY HOLD TIME	0	160	ns	
t_{ARY}	ADDRESS (CE) to READY		160	ns	
t_{RV}	Recovery Time between Controls	300		ns	
t_{RDE}	Data Out Delay from READ Control	10		ns	
t_{LD}	ALE to Data Out Valid		350	ns	Preliminary

WAVEFORMS

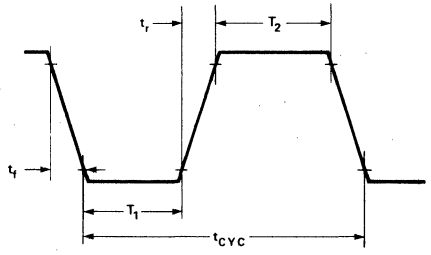


Figure 2. Clock Specification for 8755A

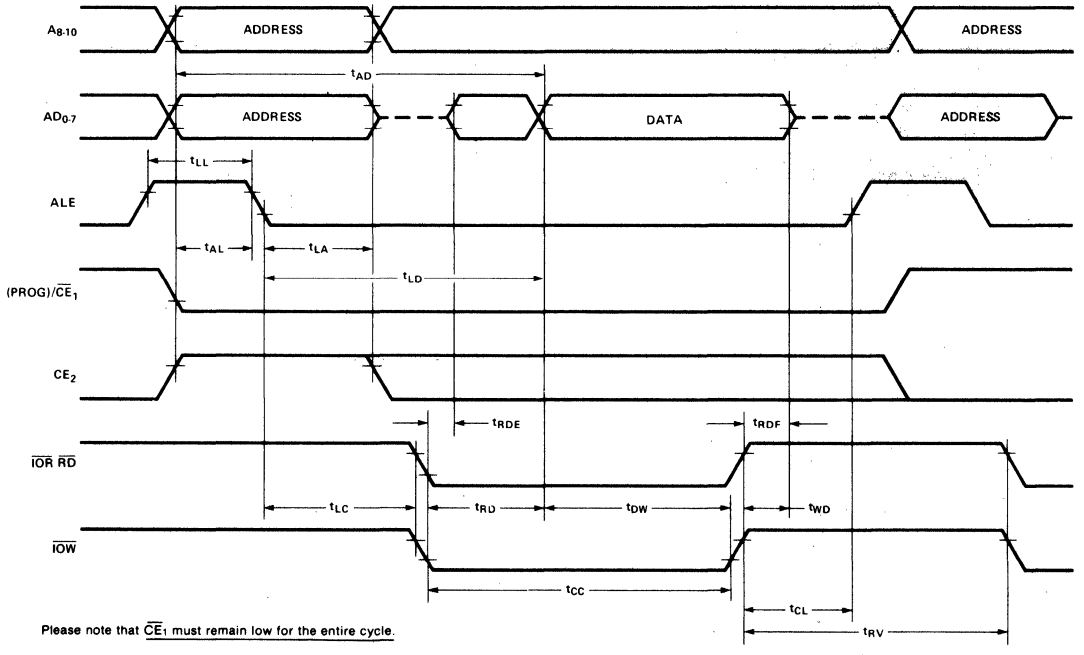
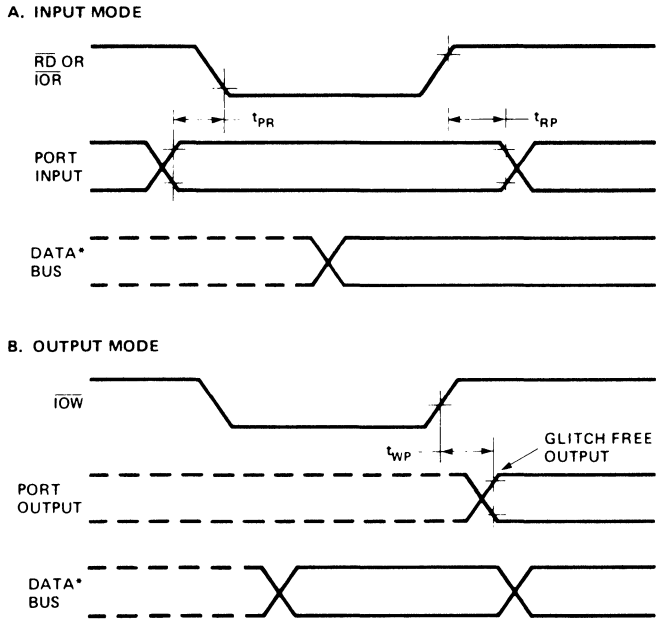


Figure 3. PROM Read, I/O Read and Write Timing



*DATA BUS TIMING IS SHOWN IN FIGURE 4.

Figure 4. I/O Port Timing

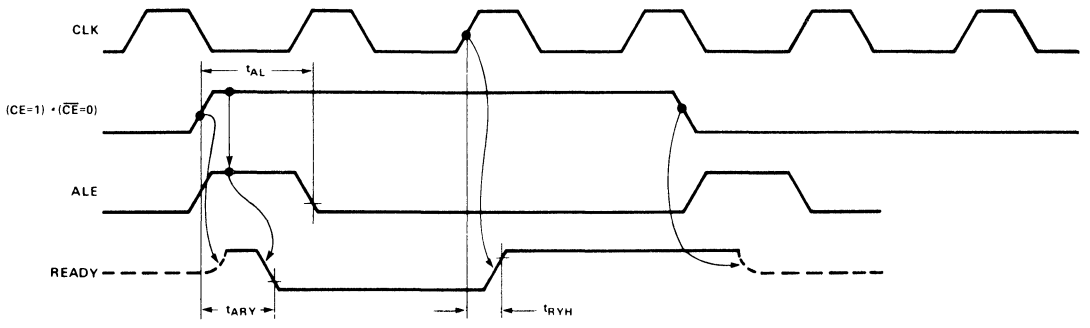


Figure 5. Wait State Timing (READY = 0)

D.C. SPECIFICATION PROGRAMMING

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

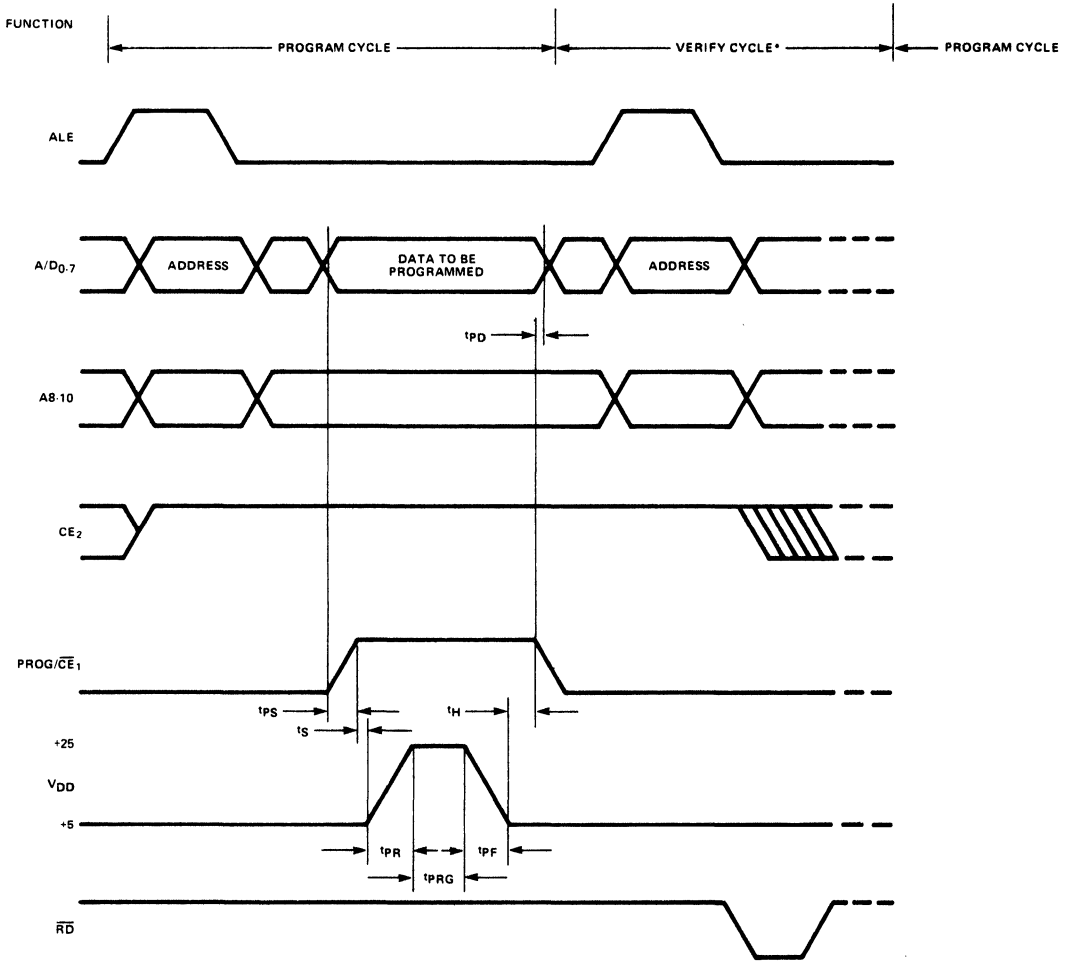
Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{DD}	Programming Voltage (during Write to EPROM)	24	25	26	V
I_{DD}	Prog Supply Current		15	30	mA

A.C. SPECIFICATION FOR PROGRAMMING

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

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{PS}	Data Setup Time	10			ns
t_{PD}	Data Hold Time	0			ns
t_S	Prog Pulse Setup Time	2			μs
t_H	Prog Pulse Hold Time	2			μs
t_{PR}	Prog Pulse Rise Time	0.01	2		μs
t_{PF}	Prog Pulse Fall Time	0.01	2		μs
t_{PRG}	Prog Pulse Width	45	50		msec

WAVEFORMS



*VERIFY CYCLE IS A REGULAR MEMORY READ CYCLE (WITH V_{DD} = +5V FOR 8755A)

Figure 6. 8755A Program Mode Timing Diagram

8155/8156/8155-2/8156-2
2048 BIT STATIC MOS RAM WITH I/O PORTS AND TIMER

8085A	8085A-2	Compatible CPU / Chip Enable
8155	8155-2	ACTIVE LOW
8156	8156-2	ACTIVE HIGH

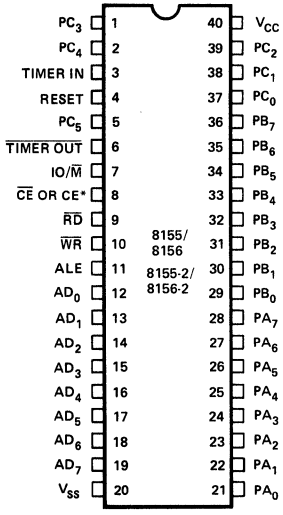
- **256 Word x 8 Bits**
- **Single +5V Power Supply**
- **Completely Static Operation**
- **Internal Address Latch**
- **2 Programmable 8 Bit I/O Ports**
- **1 Programmable 6-Bit I/O Port**
- **Programmable 14-Bit Binary Counter/ Timer**
- **Multiplexed Address and Data Bus**
- **40 Pin DIP**

The 8155 and 8156 are RAM and I/O chips to be used in the MCS-85™ microcomputer system. The RAM portion is designed with 2048 static cells organized as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085A CPU. The 8155-2 and 8156-2 have maximum access times of 330 ns for use with the 8085A-2.

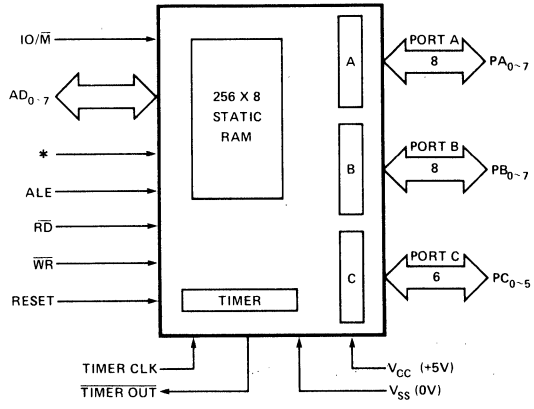
The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

PIN CONFIGURATION



BLOCK DIAGRAM



*: 8155/8155-2 = $\overline{CE}$, 8156/8156-2 = CE

8155/8156 PIN FUNCTIONS

Symbol	Function
RESET (input)	Pulse provided by the 8085A to initialize the system (connect to 8085A RESET OUT). Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two 8085A clock cycle times.
AD_{0-7} (input)	3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch inside the 8155/56 on the falling edge of ALE. The address can be either for the memory section or the I/O section depending on the IO/$\overline{M}$ input. The 8-bit data is either written into the chip or read from the chip, depending on the $\overline{WR}$ or $\overline{RD}$ input signal.
CE or $\overline{CE}$ (input)	Chip Enable: On the 8155, this pin is $\overline{CE}$ and is ACTIVE LOW. On the 8156, this pin is CE and is ACTIVE HIGH.
$\overline{RD}$ (input)	Read control: Input low on this line with the Chip Enable active enables and AD_{0-7} buffers. If IO/$\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/status registers will be read to the AD bus.
$\overline{WR}$ (input)	Write control: Input low on this line with the Chip Enable active causes the data on the Address/Data bus to be written to the RAM or I/O ports and command/status register depending on IO/$\overline{M}$.
ALE (input)	Address Latch Enable: This control signal latches both the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
IO/$\overline{M}$ (input)	Selects memory if low and I/O and command/status registers if high.
$PA_{0-7}(8)$ (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
$PB_{0-7}(8)$ (input/output)	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
$PC_{0-5}(6)$ (input/output)	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the command register. When PC_{0-5} are used as control signals, they will provide the following: PC_0 — A INTR (Port A Interrupt) PC_1 — $\overline{ABF}$ (Port A Buffer Full) PC_2 — $\overline{A\ STB}$ (Port A Strobe) PC_3 — B INTR (Port B Interrupt) PC_4 — $\overline{B\ BF}$ (Port B Buffer Full) PC_5 — B STB (Port B Strobe)
TIMER IN (input)	Input to the counter-timer.
$\overline{TIMER\ OUT}$ (output)	Timer output. This output can be either a square wave or a pulse depending on the timer mode.
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.

DESCRIPTION

The 8155/8156 contains the following:

- 2k Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
- 14-bit timer-counter

The IO/$\overline{M}$ (IO/Memory Select) pin selects either the five registers (Command, Status, PA_{0-7}, PB_{0-7}, PC_{0-5}) or the memory (RAM) portion. (See Figure 1.)

The 8-bit address on the Address/Data lines, Chip Enable input CE or $\overline{CE}$, and IO/$\overline{M}$ are all latched on-chip at the falling edge of ALE. (See Figure 2.)

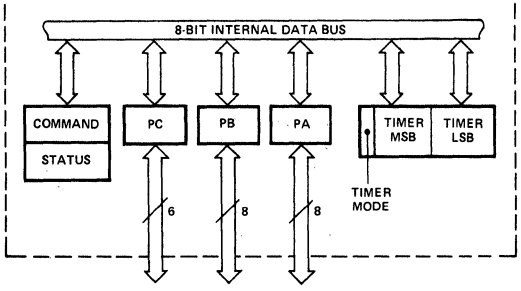
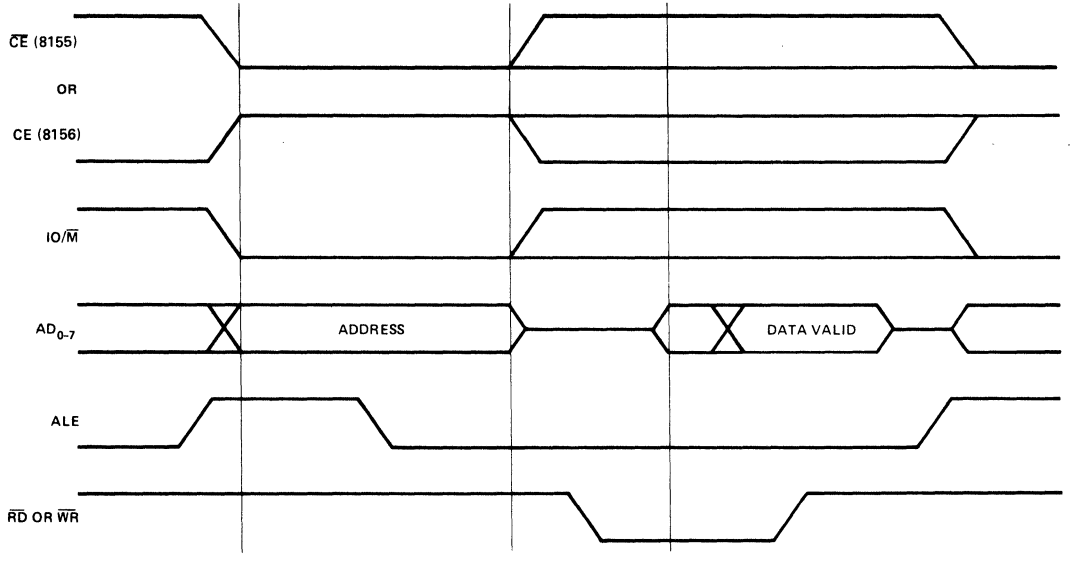


Figure 1. 8155/8156 Internal Registers



NOTE: FOR DETAILED TIMING INFORMATION, SEE FIGURE 12 AND A.C. CHARACTERISTICS.

Figure 2. 8155/8156 On-Board Memory Read/Write Cycle

PROGRAMMING OF THE COMMAND REGISTER

The command register consists of eight latches. Four bits (0-3) define the mode of the ports, two bits (4-5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6-7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation with the Chip Enable active and IO/$\overline{M}$ = 1. The meaning of each bit of the command byte is defined in Figure 3. The contents of the command register may never be read.

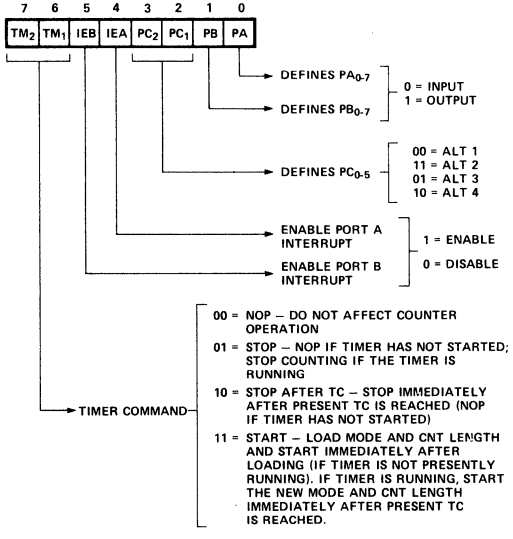


Figure 3. Command Register Bit Assignment

READING THE STATUS REGISTER

The status register consists of seven latches, one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in Figure 4. Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

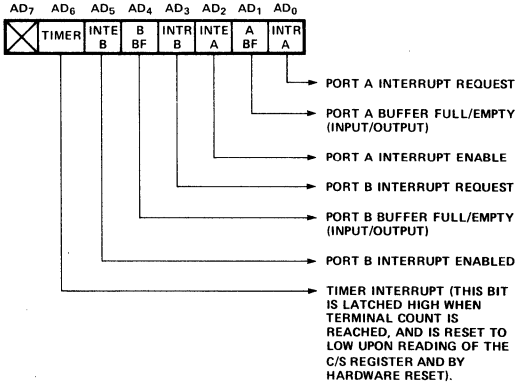


Figure 4. Status Register Bit Assignment

INPUT/OUTPUT SECTION

The I/O section of the 8155/8156 consists of five registers: (See Figure 5.)

- **Command/Status Register (C/S)** — Both registers are assigned the address XXXXX000. The C/S address serves the dual purpose.

  When the C/S registers are selected during WRITE operation, a command is written into the command register. The contents of this register are *not* accessible through the pins.

  When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer becomes available on the AD_{0-7} lines.

- **PA Register** — This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (See timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.

- **PB Register** — This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.

- **PC Register** — This register has the address XXXXX011 and contains only 6 bits. The 6 bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.

  When PC_{0-5} is used as a control port, 3 bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155 sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. (See Table 1.)

When the 'C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

CONTROL	INPUT MODE	OUTPUT MODE
BF	Low	Low
INTR	Low	High
STB	Input Control	Input Control

I/O ADDRESS†								SELECTION
A7	A6	A5	A4	A3	A2	A1	A0	
X	X	X	X	X	0	0	0	Interval Command/Status Register
X	X	X	X	X	0	0	1	General Purpose I/O Port A
X	X	X	X	X	0	1	0	General Purpose I/O Port B
X	X	X	X	X	0	1	1	Port C — General Purpose I/O or Control
X	X	X	X	X	1	0	0	Low-Order 8 bits of Timer Count
X	X	X	X	X	1	0	1	High 6 bits of Timer Count and 2 bits of Timer Mode

X: Don't Care.
†: I/O Address must be qualified by CE = 1 (8156) or $\overline{CE}$ = 0 (8155) and IO/$\overline{M}$ = 1 in order to select the appropriate register.

Figure 5. I/O port and Timer Addressing Scheme

Figure 6 shows how I/O PORTS A and B are structured within the 8155 and 8156:

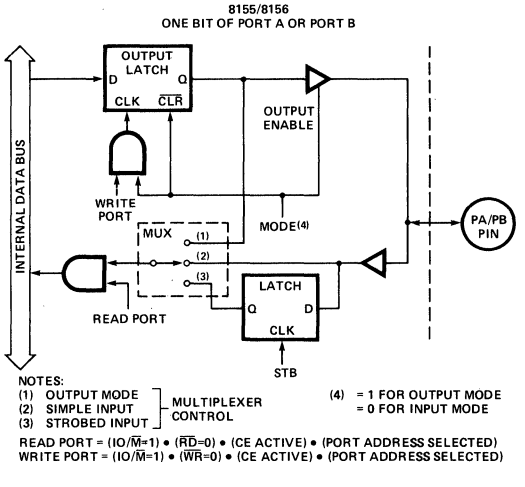


Figure 6. 8155/8156 Port Functions

TABLE 1. TABLE OF PORT CONTROL ASSIGNMENT.

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

The outputs of the 8155/8156 are "glitch-free" meaning that you can write a "1" to a bit position that was previously "1" and the level at the output pin will not change.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go low. When the 8155/56 is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of PORT C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

Figure 7 shows how the 8155/8156 I/O ports might be configured in a typical MCS-85 system.

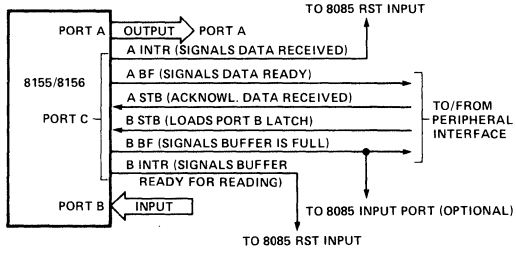


Figure 7. Example: Command Register = 00111001

TIMER SECTION

The timer is a 14-bit down-counter that counts the TIMER IN pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register. (See Figure 5).

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0-13 of the high order count register will specify the length of the next count and bits 14-15 of the high order register will specify the timer output mode (see Figure 8). The value loaded into the count length register can have any value from 2H through 3FFH in Bits 0-13.

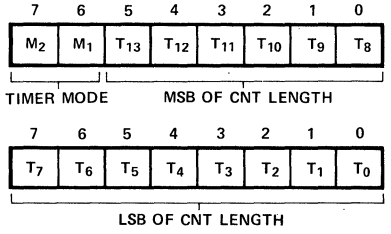


Figure 8. Timer Format

There are four modes to choose from: M2 and M1 define the timer mode, as shown in Figure 9.

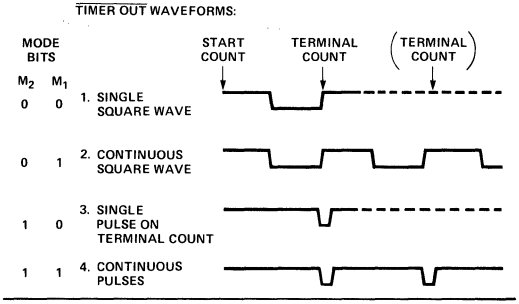


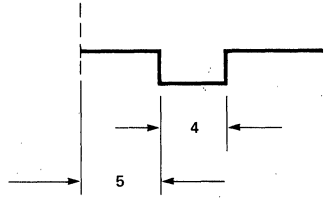
Figure 9. Timer Modes

Bits 6-7 (TM_2 and TM_1) of command register contents are used to start and stop the counter. There are four commands to choose from:

TM_2	TM_1	
0	0	NOP — Do not affect counter operation.
0	1	STOP — NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC — Stop immediately after present TC is reached (NOP if timer has not started)
1	1	START — Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

Note that while the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be used by the counter, you must issue a START command to the counter. This applies even though you may only want to change the count and use the previous mode.

In case of an odd-numbered count, the first half-cycle of the squarewave output, which is high, is one count longer than the second (low) half-cycle, as shown in Figure 10.



NOTE: 5 AND 4 REFER TO THE NUMBER OF CLOCKS IN THAT TIME PERIOD

Figure 10. Asymmetrical Square-Wave Output Resulting from Count of 9

The counter in the 8155 is not initialized to any particular mode or count when hardware RESET occurs, but RESET does *stop* the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

Please note that the timer circuit on the 8155/8156 chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary) or 2 (decimal). (For the detection of single pulses, it is suggested that one of the hardware interrupt pins on the 8085A be used.) After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count
2. Read in the 16-bit value from the count length registers
3. Reset the upper two mode bits
4. Reset the carry and rotate right one position all 16 bits through carry
5. If carry is set, add 1/2 of the full original count (1/2 full count — 1 if full count is odd).

Note: If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the 8155/56 always counts out the right number of pulses in generating the TIMER OUT waveforms.

EXAMPLE PROGRAM

Following is an actual sequence of program steps that adjusts the 8155/56 count register contents to obtain the count, extracted from Intel® Application Note AP38. "Application Techniques for the Intel 8085A Bus." First store the value of the full original count in register HL of the 8085A. Then stop the count to avoid getting an incorrect count value. Then sample the timer-counter, storing the lower-order byte of the current count register in register C and the higher-order count byte in register B. Then, call the following 8080A/8085A subroutine:

```
ADJUST, 78      MOV A,B     ;Load accumulator with upper half
                            ; of count.
        E63F    ANI 3F      ;Reset upper 2 bits and clear carry.
        1F      RAR         ;Rotate right through carry.
        47      MOV B,A     ;Store shifted value back in B.
        79      MOV A,C     ;Load accumulator with lower half.
        1F      RAR         ;Rotate right through carry.
        4F      MOV C,A     ;Store lower byte in C.
        DØ      RNC         ;If in 2nd half of count, return.
                            ;If in 1st half, go on.
        3F      CMC         ;Clear carry.
        7C      MOV A,H     ;Divide full count by 2. (If HL
                            ;is odd, disregard remainder.)
        1F      RAR
        67      MOV H,A
        7D      MOV A,L
        1F      RAR
        6F      MOV L,A
        09      DAD B       ;Double-precision add HL and BC.
        44      MOV B,H     ;Store results back in BC.
        4D      MOV C,L
        C9      RET         ;Return.
```

After executing the subroutine, BC will contain the remaining count in the current count cycle.

8085A MINIMUM SYSTEM CONFIGURATION

Figure 11 shows a minimum system using three chips, containing:

- 256 Bytes RAM
- 2K Bytes ROM
- 38 I/O Pins
- 1 Interval Timer
- 4 Interrupt Levels

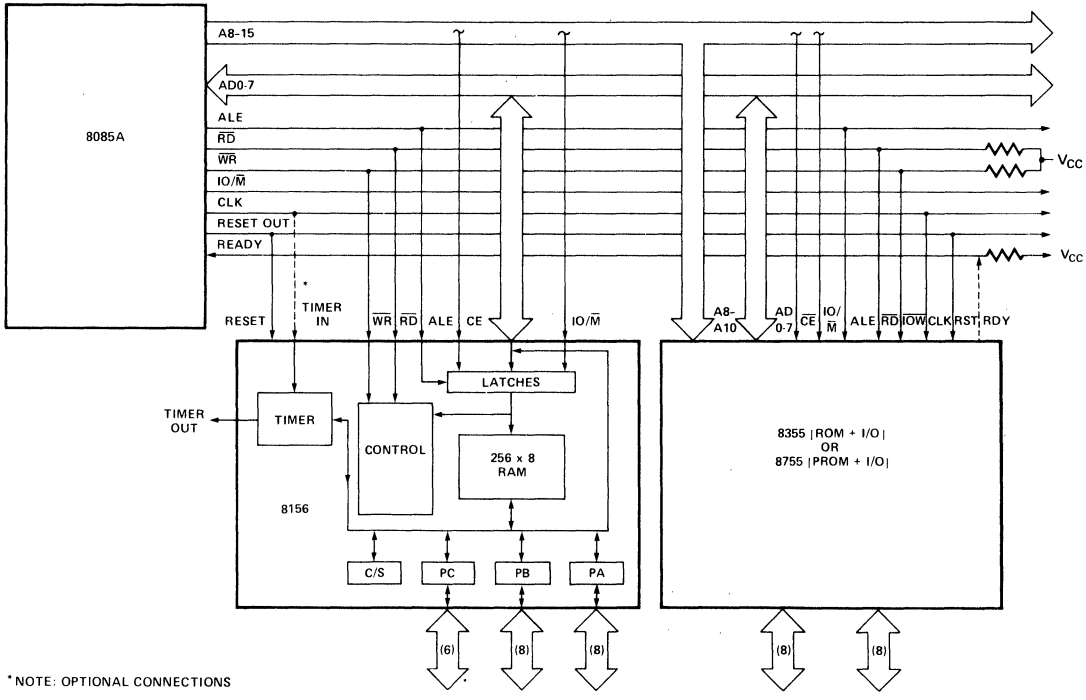


Figure 11. 8085A Minimum System Configuration. (Memory Mapped I/O)

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin
With Respect to Ground ..... -0.5V to +7V
Power Dissipation 1.5W

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

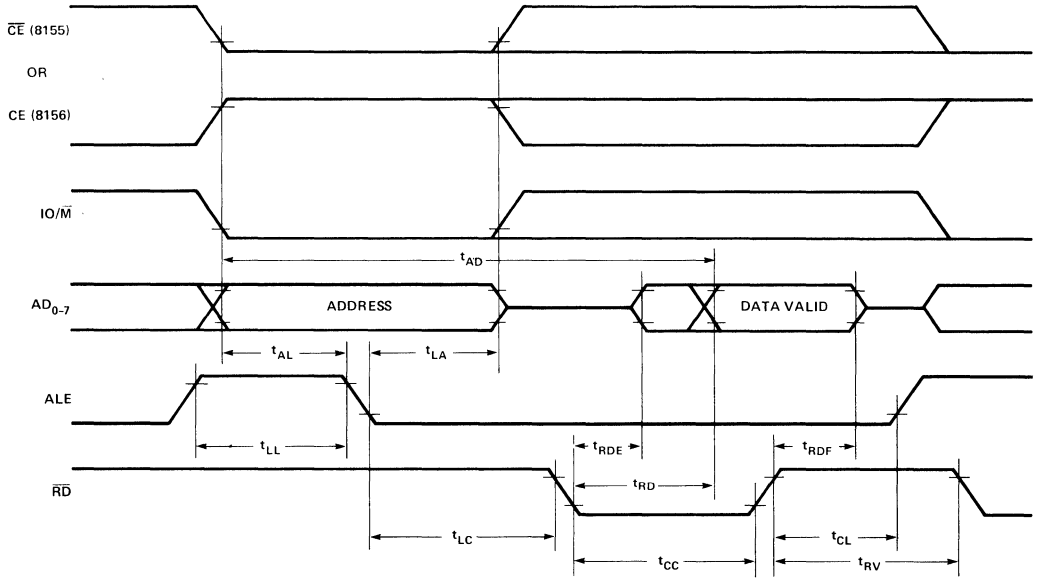
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	
I_{IL}(CE)	Chip Enable Leakage 8155 8156		 +100 −100	 μA μA	V_{IN} = V_{CC} to 0V

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

		8155/8156		8155-2/8156-2 (Preliminary)		
SYMBOL	PARAMETER	MIN.	MAX.	MIN.	MAX.	UNITS
t_{AL}	Address to Latch Set Up Time	50		30		ns
t_{LA}	Address Hold Time after Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{AD}	Address Stable to Data Out Valid		400		330	ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float After READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data In to WRITE Set Up Time	150		100		ns
t_{WD}	Data In Hold Time After WRITE	0		0		ns
t_{RV}	Recovery Time Between Controls	300		200		ns
t_{WP}	WRITE to Port Output		400		300	ns
t_{PR}	Port Input Setup Time	70		50		ns
t_{RP}	Port Input Hold Time	50		10		ns
t_{SBF}	Strobe to Buffer Full		400		300	ns
t_{SS}	Strobe Width	200		150		ns
t_{RBE}	READ to Buffer Empty		400		300	ns
t_{SI}	Strobe to INTR On		400		300	ns
t_{RDI}	READ to INTR Off		400		300	ns
t_{PSS}	Port Setup Time to Strobe Strobe	50		0		ns
t_{PHS}	Port Hold Time After Strobe	120		100		ns
t_{SBE}	Strobe to Buffer Empty		400		300	ns
t_{WBF}	WRITE to Buffer Full		400		300	ns
t_{WI}	WRITE to INTR Off		400		300	ns
t_{TL}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ Low		400		300	ns
t_{TH}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ High		400		300	ns
t_{RDE}	Data Bus Enable from READ Control	10		10		ns
t_1	TIMER-IN Low Time	80		40		ns
t_2	TIMER-IN High Time	120		70		ns

WAVEFORMS

a. Read Cycle



b. Write Cycle

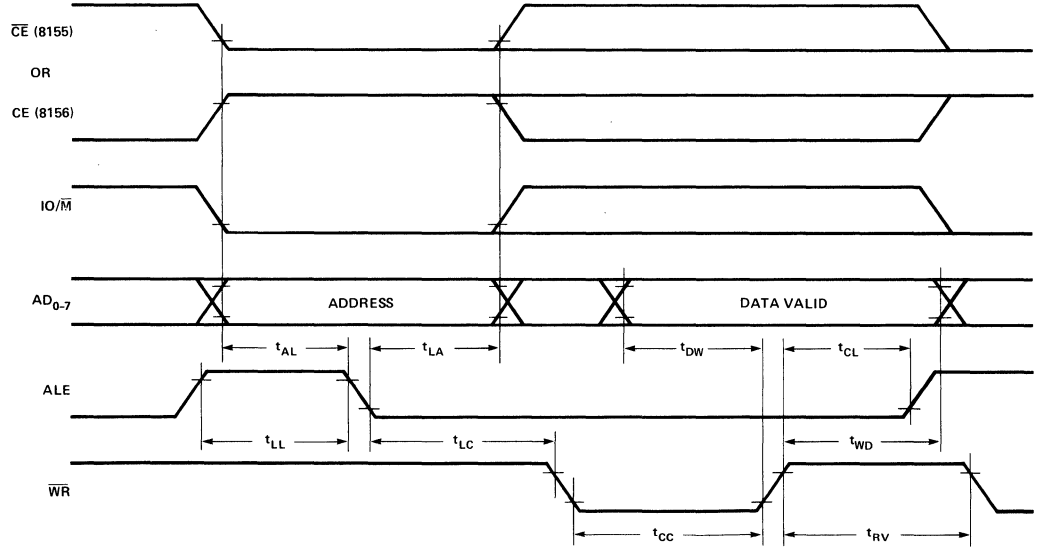
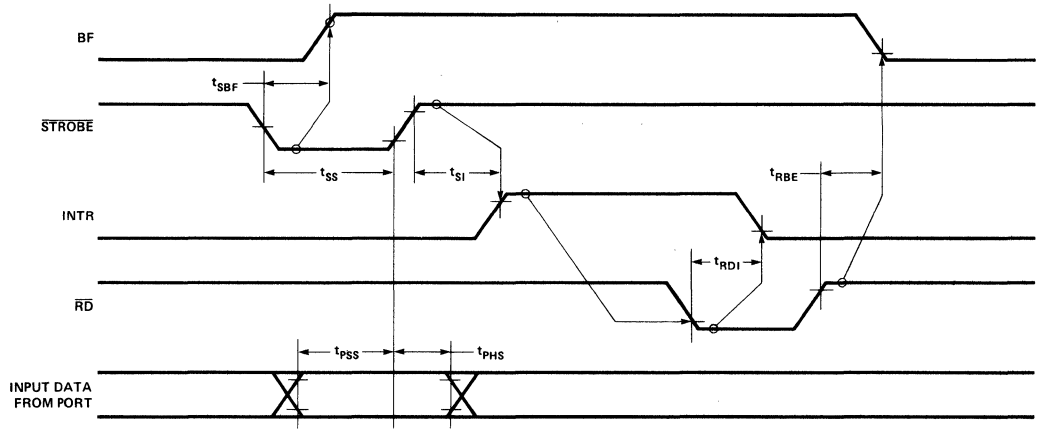


Figure 12. 8155/8156 Read/Write Timing Diagrams

a. Strobed Input Mode



b. Strobed Output Mode

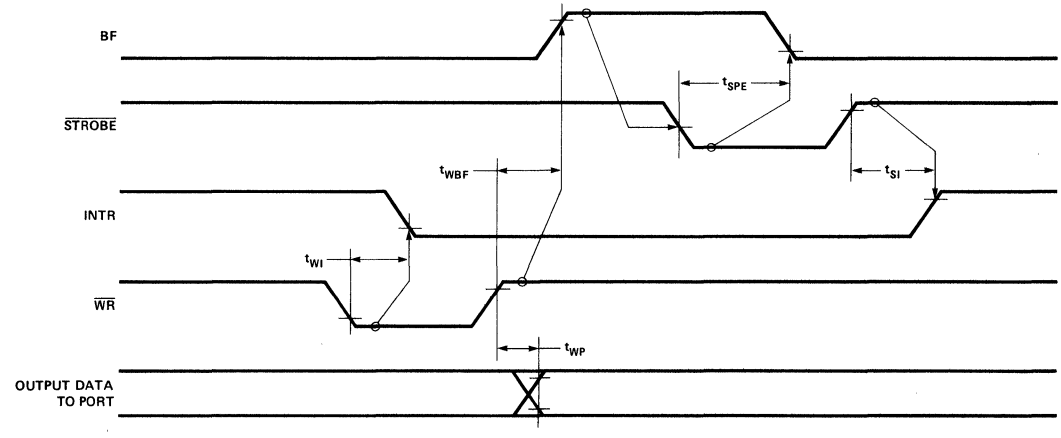
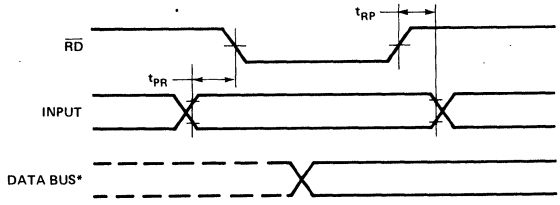
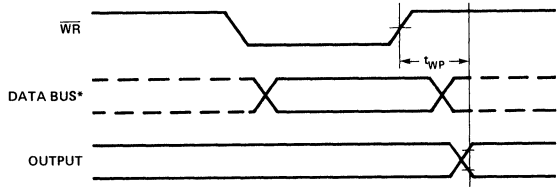


Figure 13. Strobed I/O Timing

a. Basic Input Mode

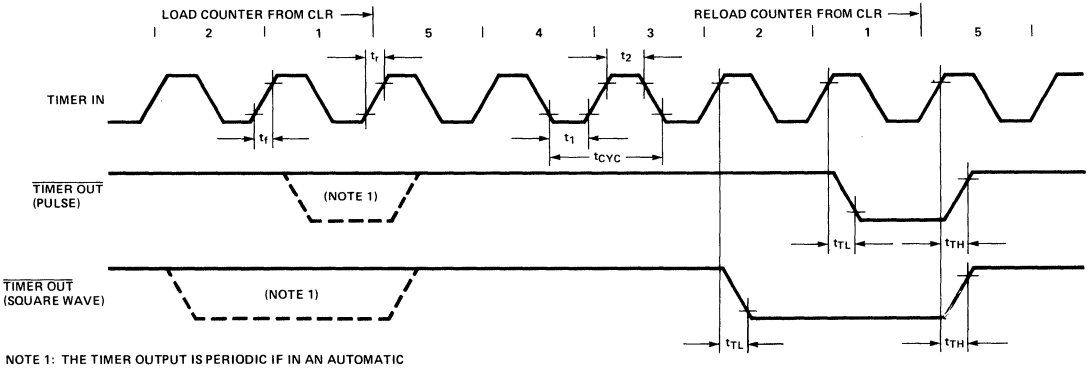


b. Basic Output Mode



*DATA BUS TIMING IS SHOWN IN FIGURE 7.

Figure 14. Basic I/O Timing Waveform



NOTE 1: THE TIMER OUTPUT IS PERIODIC IF IN AN AUTOMATIC RELOAD MODE (M_1 MODE BIT = 1)

Figure 15. Timer Output Waveform Countdown from 5 to 1

8185/8185-2
1024 x 8-BIT STATIC RAM FOR MCS-85™

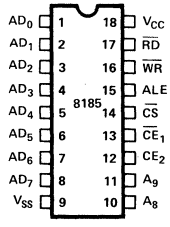
- **Multiplexed Address and Data Bus**
- **Directly Compatible with 8085A and 8088 Microprocessors**
- **Low Operating Power Dissipation**
- **Low Standby Power Dissipation**
- **Single +5V Supply**
- **High Density 18-Pin Package**

The Intel® 8185 is an 8192-bit static random access memory (RAM) organized as 1024 words by 8-bits using N-channel Silicon-Gate MOS technology. The multiplexed address and data bus allows the 8185 to interface directly to the 8085A and 8088 microprocessors to provide a maximum level of system integration.

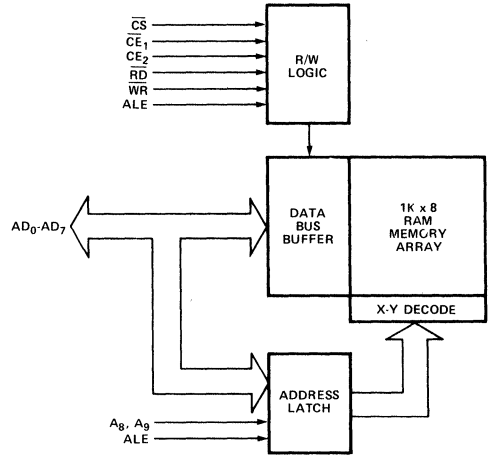
The low standby power dissipation minimizes system power requirements when the 8185 is disabled.

The 8185-2 is a high-speed selected version of the 8185 that is compatible with the 5 MHz 8085A-2 and the full speed 5 MHz 8088.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

AD_0-AD_7	ADDRESS/DATA LINES
A_8, A_9	ADDRESS LINES
$\overline{CS}$	CHIP SELECT
$\overline{CE}_1$	CHIP ENABLE (IO/$\overline{M}$)
CE_2	CHIP ENABLE
ALE	ADDRESS LATCH ENABLE
$\overline{RD}$	READ ENABLE
$\overline{WR}$	WRITE ENABLE

OPERATIONAL DESCRIPTION

The 8185 has been designed to provide for direct interface to the multiplexed bus structure and bus timing of the 8085A microprocessor.

At the beginning of an 8185 memory access cycle, the 8-bit address on AD_{0-7}, A_8 and A_9, and the status of $\overline{CE}_1$ and CE_2 are all latched internally in the 8185 by the falling edge of ALE. If the latched status of both $\overline{CE}_1$ and CE_2 are active, the 8185 powers itself up, but no action occurs until the $\overline{CS}$ line goes low and the appropriate $\overline{RD}$ or $\overline{WR}$ control signal input is activated.

The $\overline{CS}$ input is not latched by the 8185 in order to allow the maximum amount of time for address decoding in selecting the 8185 chip. Maximum power consumption savings will occur, however, only when $\overline{CE}_1$ and CE_2 are activated selectively to power down the 8185 when it is not in use. A possible connection would be to wire the 8085A's IO/$\overline{M}$ line to the 8185's $\overline{CE}_1$ input, thereby keeping the 8185 powered down during I/O and interrupt cycles.

TABLE 1.
TRUTH TABLE FOR
POWER DOWN AND FUNCTION ENABLE

$\overline{CE}_1$	CE_2	$\overline{CS}$	(CS*)[2]	8185 Status
1	X	X	0	Power Down and Function Disable[1]
X	0	X	0	Power Down and Function Disable[1]
0	1	1	0	Powered Up and Function Disable[1]
0	1	0	1	Powered Up and Enabled

Notes:
X: Don't Care.
1: Function Disable implies Data Bus in high impedance state and not writing.
2: CS* = ($\overline{CE}_1 = 0$) • ($CE_2 = 1$) • ($\overline{CS} = 0$)
CS* = 1 signifies all chip enables and chip select active

TABLE 2.
TRUTH TABLE FOR
CONTROL AND DATA BUS PIN STATUS

(CS*)	$\overline{RD}$	$\overline{WR}$	AD_{0-7} During Data Portion of Cycle	8185 Function
0	X	X	Hi-Impedance	No Function
1	0	1	Data from Memory	Read
1	1	0	Data to Memory	Write
1	1	1	Hi-Impedance	Reading, but not Driving Data Bus

Note:
X: Don't Care.

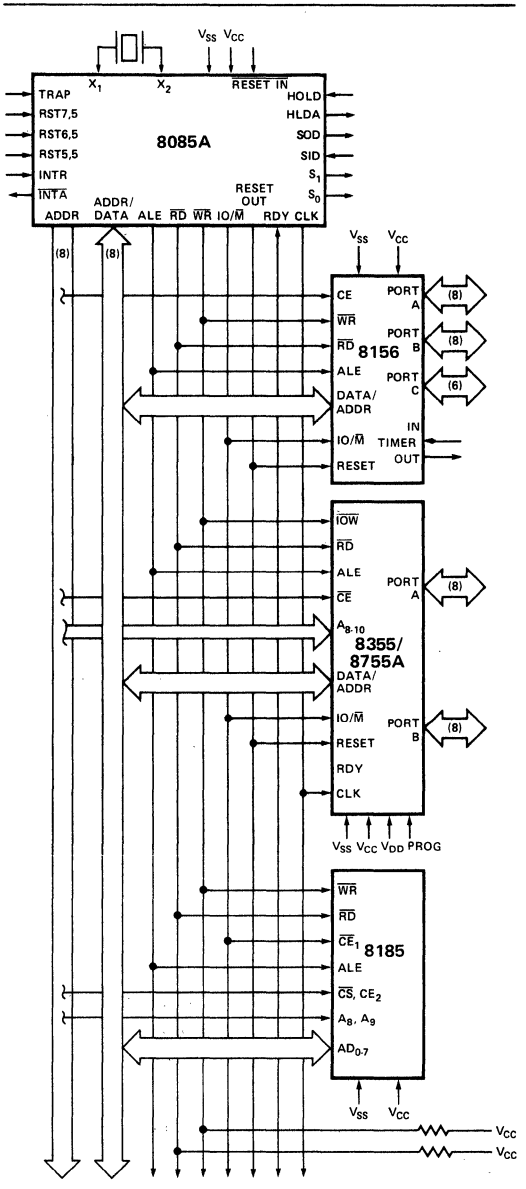


Figure 1. 8185 in an MCS-85 System.

4 Chips:
2K Bytes ROM
1.25K Bytes RAM
38 I/O Lines
1 Counter/Timer
2 Serial I/O Lines
5 Interrupt Inputs

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... 0°C to +70°C
Storage Temperature ..... -65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... -0.5V to +7V
Power Dissipation 1.5W

*COMMENT

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

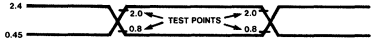
Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4			I_{OH} = – 400μA
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤ V_{OUT} ≤ V_{CC}
I_{CC}	V_{CC} Supply Current Powered Up		100	mA	
	Powered Down		35	mA	

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

Symbol	Parameter [1]	8185 Preliminary		8185-2 Preliminary		Units
		Min.	Max.	Min.	Max.	
t_{AL}	Address to Latch Set Up Time	50		30		ns
t_{LA}	Address Hold Time After Latch	80		30		ns
t_{LC}	Latch to READ/WRITE Control	100		40		ns
t_{RD}	Valid Data Out Delay from READ Control		170		140	ns
t_{LD}	ALE to Data Out Valid		300		200	ns
t_{LL}	Latch Enable Width	100		70		ns
t_{RDF}	Data Bus Float After READ	0	100	0	80	ns
t_{CL}	READ/WRITE Control to Latch Enable	20		10		ns
t_{CC}	READ/WRITE Control Width	250		200		ns
t_{DW}	Data In to WRITE Set Up Time	150		150		ns
t_{WD}	Data In Hold Time After WRITE	20		20		ns
t_{SC}	Chip Select Set Up to Control Line	10		10		ns
t_{CS}	Chip Select Hold Time After Control	10		10		ns
t_{ALCE}	Chip Enable Set Up to ALE Falling	30		10		ns
t_{LACE}	Chip Enable Hold Time After ALE	50		30		ns

Notes:
1. All AC parameters are referenced at
   a) 2.4V and .45V for inputs
   b) 2.0V and .8V for outputs.

Input Waveform for A.C. Tests:



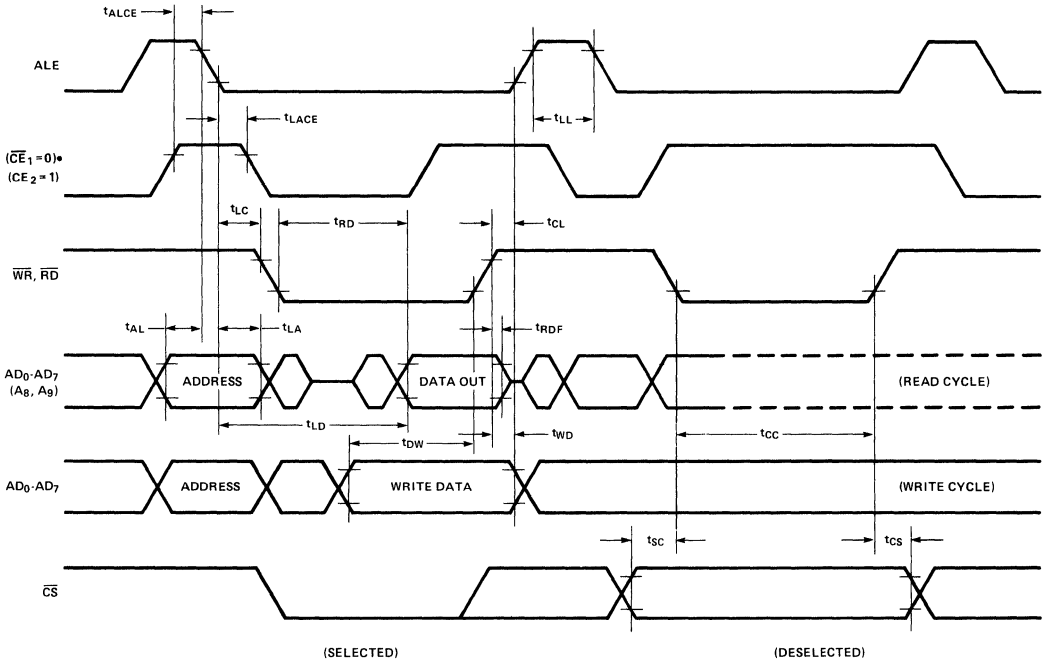


Figure 3. 8185 Timing.

MCS-51™ Component Specifications

7

PRELIMINARY

8031/8051/8751
SINGLE-COMPONENT 8-BIT MICROCOMPUTER

- 8031 - Control Oriented CPU With RAM and I/O
- 8051 - An 8031 With Factory Mask-Programmable ROM
- 8751 - An 8031 With User Programmable/Erasable EPROM

- 4K x 8 ROM/EPROM
- 128 x 8 RAM
- Four 8-Bit Ports, 32 I/O Lines
- Two 16-Bit Timer/Event Counters
- High-Performance Full-Duplex Serial Channel
- External Memory Expandable to 128K
- Compatible with MCS-80™/MCS-85™ Peripherals
- Boolean Processor
- MCS-48™ Architecture Enhanced with:
  - Non-Paged Jumps
  - Direct Addressing
  - Four 8-Register Banks
  - Stack Depth Up to 128-Bytes
  - Multiply, Divide, Subtract, Compare
- Most Instructions Execute in 1μs
- 4μs Multiply and Divide

The Intel® 8031/8051/8751 is a stand-alone, high-performance single-chip computer fabricated with Intel's highly-reliable +5 Volt, depletion-load, N-Channel, silicon-gate HMOS technology and packaged in a 40-pin DIP. It provides the hardware features, architectural enhancements and new instructions that are necessary to make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 8051/8751 contains a non-volatile 4K x 8 read only program memory; a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The 8031 is identical, except that it lacks the program memory. For systems that require extra capability, the 8051 can be expanded using standard TTL compatible memories and the byte oriented MCS-80 and MCS-85 peripherals.

The 8051 microcomputer, like its 8048 predecessor, is efficient both as a controller and as an arithmetic processor. The 8051 has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiply and divide require only 4μs. Among the many instructions added to the standard 8048 instruction set are multiply, divide, subtract and compare.

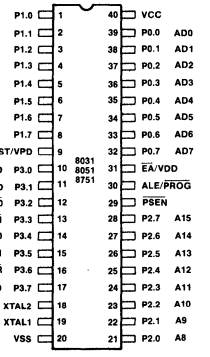


Figure 1. Pin Configuration

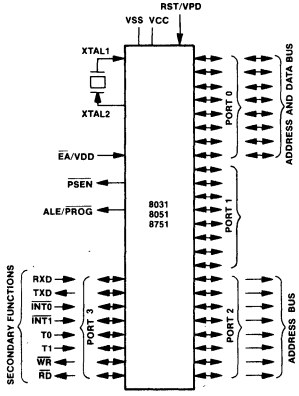


Figure 2. Logic Symbol

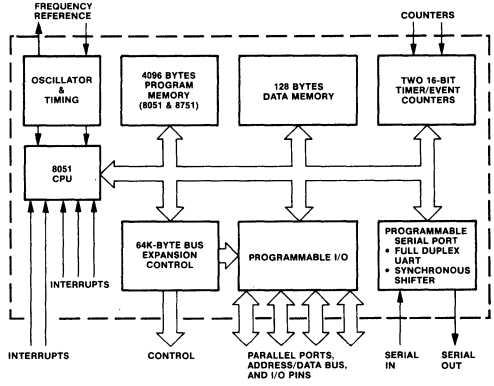


Figure 3. Block Diagram

AFN-01462A-01

1.0 INTRODUCTION

This data sheet provides an introduction to the 8051 family. A detailed description of the hardware required to expand the 8051 with more program memory, data memory, I/O, specialized peripherals and into multiprocessor configurations is described in the 8051 Family User's Manual.

1.1 THE 8051 FAMILY

The 8051 is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time applications such as instrumentation, industrial control and intelligent computer peripherals. It provides the hardware features, architectural enhancements and new instructions that make it a powerful and cost effective controller for applications requiring up to 64K-bytes of program memory and/or up to 64K-bytes of data storage. A Block Diagram is shown in Figure 3.

The 8031 is a control-oriented CPU without on-chip program memory. It can address 64K-bytes of external Program Memory in addition to 64K-bytes of External Data Memory. For systems requiring extra capability, each member of the 8051 family can be expanded using standard memories and the byte oriented MCS-80 and MCS-85 peripherals. The 8051 is an 8031 with the lower 4K-bytes of Program Memory filled with on-chip mask programmable ROM while the 8751 has 4K-bytes of UV-light-erasable/electrically-programmable ROM.

The three pin-compatible versions of this component reduce development problems to a minimum and provide maximum flexibility. The 8751 is well suited for development, prototyping, low-volume production and applications requiring field updates; the 8051 for low-cost, high volume production; and the 8031 for applications desiring the flexibility of external Program Memory which can be easily modified and updated in the field.

2.0 MACRO-VIEW OF THE 8051 ARCHITECTURE

On a single die the 8051 microcomputer combines CPU; non-volatile 4K x 8 read-only program memory; volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; serial I/O port for either multi-processor communciations, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. This section will provide an overview of the 8051 by providing a high-level description of its major elements: the CPU architecture and the on-chip functions peripheral to the CPU. The generic term "8051" is used to refer collectively to the 8031, 8051, and 8751.

2.1 8051 CPU ARCHITECTURE

The 8051 CPU manipulates operands in four memory spaces. These are the 64K-byte Program Memory, 64K-byte External Data Memory, 384-byte Internal Data Memory and 16-bit Program Counter spaces. The Internal Data Memory address space is further divided into the 256-byte Internal Data RAM and 128-byte Special Function Register (SFR) address spaces shown in Figure 2.1. Four Register Banks (each with eight registers), 128 addressable bits, and the stack reside in the Internal Data RAM. The stack depth is limited only by the available Internal Data RAM and its location is determined by the 8-bit stack pointer. All registers except the Program Counter and the four 8-Register Banks reside in the Special Function Register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. 128 bit locations in the SFR address space are addressable as bits. The 8051 contains 128 bytes of Internal Data RAM and 20 SFRs.

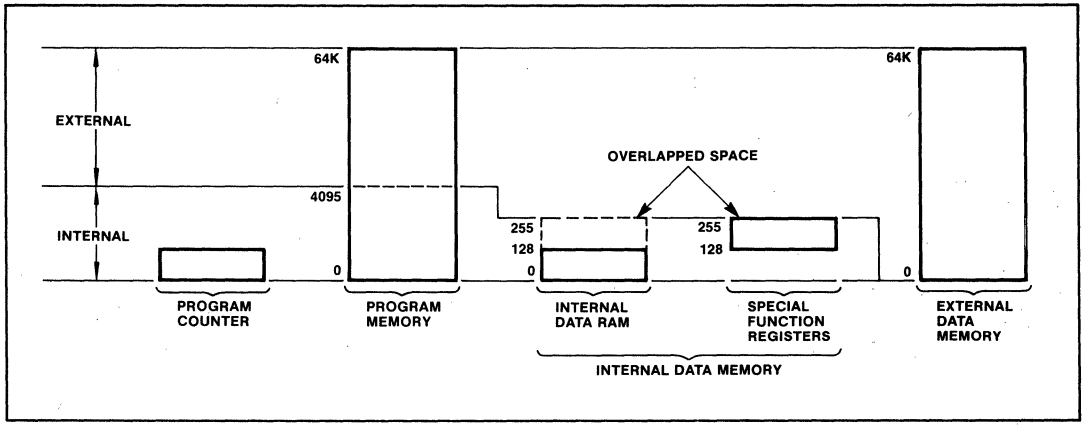


Figure 2.1. 8051 Family Memory Organization

The 8051 provides a non-paged Program Memory address space to accommodate relocatable code. Conditional branches are performed relative to the Program Counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. Sixteen-bit jumps and calls permit branching to any location in the contiguous 64K Program Memory address space.

The 8051 has five methods for addressing source operands: Register, Direct, Register-Indirect, Immediate and Base-Register- plus Index-Register-Indirect Addressing. The first three methods can be used for addressing destination operands. Most instructions have a "destination, source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Registers in the four 8-Register Banks can be accessed through Register, Direct, or Register-Indirect Addressing; the 128 bytes of Internal Data RAM through Direct or Register-Indirect Addressing; and the Special Function Registers through Direct Addressing. External Data Memory is accessed through Register-Indirect Addressing. Look-Up-Tables resident in Program Memory can be accessed through Base-Register- plus Index-Register- Indirect Addressing.

The 8051 is classified as an 8-bit machine since the internal ROM, RAM, Special Function Registers, Arithmetic/Logic Unit and external data bus are each 8-bits wide. The 8051 performs operations on bit, nibble, byte and double-byte data types.

The 8051 has extensive facilities for byte transfer, logic, and integer arithmetic operations. It excels at bit handling since data transfer, logic and conditional branch operations can be performed directly on Boolean variables.

The 8051's instruction set is an enhancement of the instruction set familiar to MCS-48 users. It is enhanced to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Op codes were reassigned to add new high-power operations and to permit new addressing modes which make the old operations more orthogonal. Efficient use of program memory results from an instruction set consisting of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1μs and 45 instructions execute in 2μs. The remaining instructions (multiply and divide) require only 4μs. The number of bytes in each instruction and the number of oscillator periods required for execution are listed in the appended 8051 Instruction Set Summary.

2.2 ON-CHIP PERIPHERAL FUNCTIONS

Thus far only the CPU and memory spaces of the 8051 have been described. In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The extensive I/O facilities include the I/O pins, parallel I/O ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit counters and the serial port. All of these work together to greatly boost system performance.

2.2.1 Interrupt System

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency ranges from 3μs to 7μs when using a 12 MHz crystal.

The 8051 acknowledges interrupt requests from five sources: Two from external sources via the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ pins, one from each of the two internal counters and one from the serial I/O port. Each interrupt vectors to a separate location in Program Memory for its service program. Each of the five sources can be assigned to either of two priority levels and can be independently enabled and disabled. Additionally all enabled sources can be globally disabled or enabled. Each external interrupt is programmable as either level- or transition-activated and is active-low to allow the "wire or-ing" of several interrupt sources to the input pin. The interrupt system is shown diagrammatically in Figure 2.2

2.2.2 I/O Facilities

The 8051 has instructions that treat its 32 I/O lines as 32 individually addressable bits and as four parallel 8-bit ports addressable as Ports 0, 1, 2 and 3. Ports 0, 2 and 3 can also assume other functions. Port 0 provides the multiplexed low-order address and data bus used for expanding the 8051 with standard memories and peripherals. Port 2 provides the high-order address bus when expanding the 8051 with external Program Memory or more than 256 bytes of External Data Memory. The pins of Port 3 can be configured individually to provide external interrupt request inputs, counter inputs, the serial port's receiver input and transmitter output, and to generate the control signals used for reading and writing External Data Memory. The generation or use of an alternate function on a Port 3 pin is done automatically by the 8051 as long as the pin

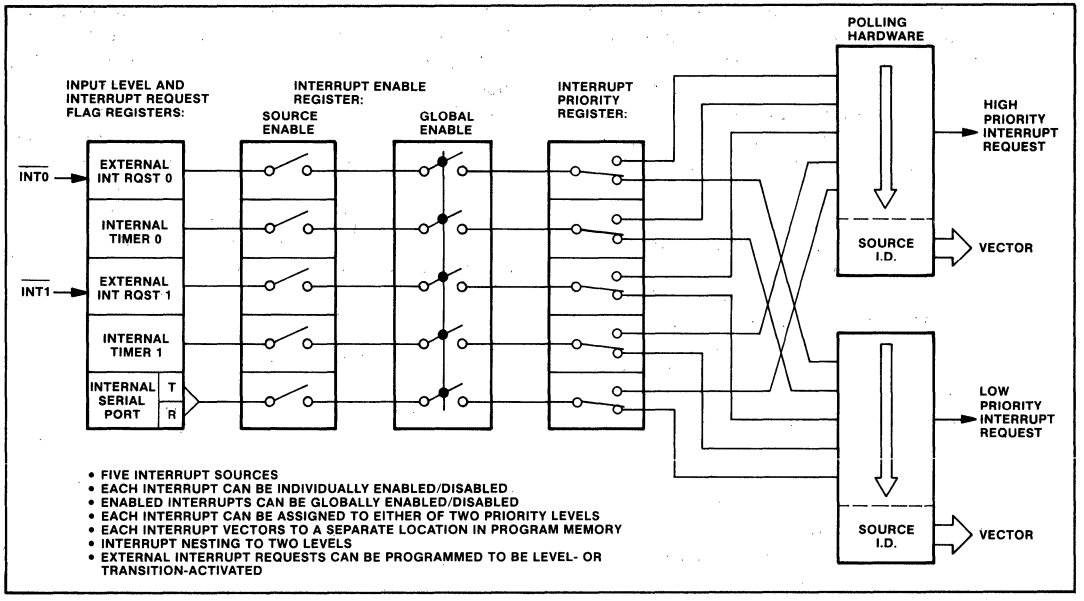


Figure 2.2. 8051 Interrupt System

is configured as an input. The configuration of the ports is shown on the 8051 Family Logic Symbol of Figure 2.

2.2.2.1 OPEN DRAIN I/O PINS

Each pin of Port 0 can be configured as an open drain output or as a high impedance input. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Re-writing the pin to a one (1) will place its output driver in a high-impedance state and configure the pin as an input. Each I/O pin of Port 0 can sink two TTL loads.

2.2.2.2 QUASI-BIDIRECTIONAL I/O PINS

Ports 1, 2 and 3 are quasi-bidirectional buffers. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Any pin that is configured as an output will be reconfigured as an input when a one (1) is written to the pin. Simultaneous to this reconfiguration the output driver of the quasi-bidirectional port will source current for two oscillator periods. Since current is sourced only when a bit previously written to a zero (0) is updated to a one (1), a pin programmed as an input will not source current into the TTL gate that is driving it if the pin is later written with another one (1). Since the quasi-bidirectional output driver sources current for only two oscillator periods, an internal pullup resistor of approximately 20K- to 40K-ohms is provided to hold the external driver's loading at a TTL high level. Ports 1, 2 and 3 can sink/source one TTL load.

2.2.2.3 MICROPROCESSOR BUS

A microprocessor bus is provided to permit the 8051 to solve a wide range of problems and to allow the upward growth of user products. This multiplexed address and data bus provides an interface compatible with standard memories, MCS-80 peripherals and the MCS-85 memories that include on-chip programmable I/O ports and timing functions. These are summarized in the 8051 Microcomputer Expansion Components chart of Figure 2.3.

When accessing external memory the high-order address is emitted on Port 2 and the low-order address on Port 0. The ALE signal is provided for strobing the address into an external latch. The program store enable ($\overline{PSEN}$) signal is provided for enabling an external memory device to Port 0 during a read from the Program Memory address space. When the MOVX instruction is executed Port 3 automatically generates the read ($\overline{RD}$) signal for enabling an External Data Memory device to Port 0 or generates the write ($\overline{WR}$) signal for strobing the external memory device with the data emitted by Port 0. Port 0 emits the address and data to the external memory through a push/pull driver that can sink/source two TTL loads. At the end of the read/write bus cycle Port 0 is automatically reprogrammed to its high

	Category	I.D.	Description	Comments	Program Or Data Memory	Crystal Frequency MHz (Max)
	I/O Expander		8 Line I/O Expander (Shift Register)	Low Cost I/O Expander		12
Compatible MCS-80/85 Components	Standard EPROMs	2758	1K x 8 450 ns Light Erasable	User programmable and erasable.	P	9
		2716-1	2K x 8 350 ns Light Erasable		P	11
		2732	4K x 8 450 ns Light Erasable		P	9
		2732A	4K x 8 250 ns Light Erasable		P	12
	Standard RAMs	2114A	1K x 4 100 ns RAM	Data memory can be easily expanded using standard NMOS RAMs.	D	12
		2148	1K x 4 70 ns RAM		D	12
		2142-2	1K x 4 200 ns RAM		D	12
	Multiplexed Address/ Data RAMs	8185A	1K x 8 300 ns RAM		D	12
	Standard I/O	8212	8-Bit I/O Port	Serves as Address Latch or I/O port.	D	12
		8282	8-Bit I/O Port		D	12
		8283	8-Bit I/O Port		D	12
		8255A	Programmable Peripheral Interface	Three 8-bit programmable I/O ports.	D	12
		8251A	Programmable Communications Interface	Serial Communications Receiver/Transmitter.	D	12
	Standard Peripherals	8205	1 of 8 Binary Decoder	MCS-80 and MCS-85 peripheral devices are compatible with the 8051 allowing easy addition of specialized interfaces. Future MCS-80/85 devices will also be compatible.	D	12
		8286	Bi-directional Bus Driver		D	12
		8287	Bi-directional Bus Driver (Inverting)		D	12
		8253A	Programmable Interval Timer		D	12
		8279	Programmable Keyboard/Display Interface (128 Keys)		D	12
		8291	GPIB Talker/Listener		D	12
		8292	GPIB Controller		D	11.7
	Universal Peripheral Interfaces	8041A	ROM Program Memory	User programmable to perform custom I/O and control functions.	D/P	12/11.7
		8741A	EPROM Program Memory		D/P	12/11.7
	Memories with on-chip I/O and Peripheral Functions.	8155-2	256 x 8 330 ns RAM		D	12
		8355-2	2K x 8 330 ns ROM		P	11.6
		8755-2	2K x 8 330 ns EPROM		P	11.6

Figure 2.3. 8051 Microcomputer Expansion Components

impedance state and Port 2 is returned to the state it had prior to the bus cycle. The 8051 generates the address, data and control signals needed by memory and I/O devices in a manner that minimizes the requirements placed on external program and data memories. At 12 MHz, the Program Memory cycle time is 500ns and the access times required from stable address and $\overline{PSEN}$ are approximately 320ns and 150ns respectively. The External Data Memory cycle time is 1μs and the access times required from stable address and from read ($\overline{RD}$) or write ($\overline{WR}$) command are approximately 600ns and 250ns respectively.

2.2.3 Timer/Event Counters

The 8051 contains two 16-bit counters for measuring time intervals, measuring pulse widths, counting events and generating precise, periodic interrupt requests. Each can be programmed independently to operate similar to an 8048 8-bit timer with divide by 32 prescaler or as an 8-bit counter with divide by 32 prescaler (Mode 0), as a 16-bit time-interval or event counter (Mode 1), or as an 8-bit time-interval or event counter with automatic reload upon overflow (Mode 2).

Additionally, counter 0 can be programmed to a mode that divides it into one 8-bit time-interval or

event counter and one 8-bit time-interval counter (Mode 3). When counter 0 is in Mode 3, counter 1 can be programmed to any of the three aforementioned modes, although it cannot set an interrupt request flag or generate an interrupt. This mode is useful because counter 1's overflow can be used to pulse the serial port's transmission-rate generator. Along with their multiple operating modes and 16-bit precision, the counters can also handle very high input frequencies. These range from 0.1 MHz to 1.0 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for an input that is a division by 12 of the oscillator frequency and from 0 Hz to an upper limit of 50 KHz to 0.5 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for external inputs. Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse widths.

The counters are started and stopped under software control. Each counter sets its interrupt request flag when it overflows from all ones to all zeros (or auto-reload value). The operating modes and input sources are summarized in Figures 2.4A and 2.4B. The effects of the configuration flags and the status flags are shown in Figures 2.5A and 2.5B.

2.2.4 Serial Communications

The 8051 has a serial I/O port that is useful for serially linking peripheral devices as well as multiple 8051s through standard asynchronous protocols with full-duplex operation. The serial port also has a synchronous mode for expansion of I/O lines using CMOS and TTL shift registers. This hardware serial communications interface saves ROM code and permits a much higher transmission rate than could be achieved through software. In response to a serial port interrupt request the CPU has only to read/write the serial port's buffer to service the serial link. A block diagram of the serial port is shown in Figure 2.6. Methods for linking UART (universal asynchronous receiver/transmitter) devices are shown in

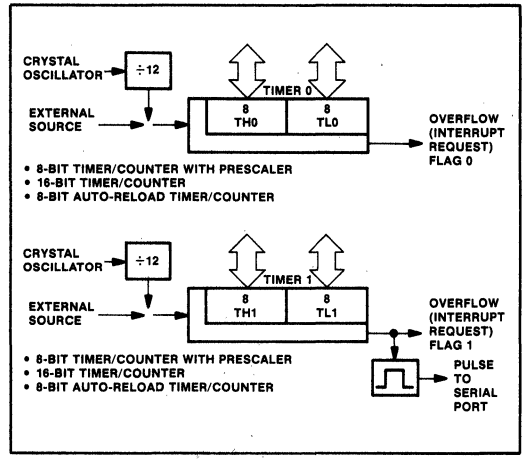


Figure 2.4.A Timer/Event Counter Modes 0, 1 and 2

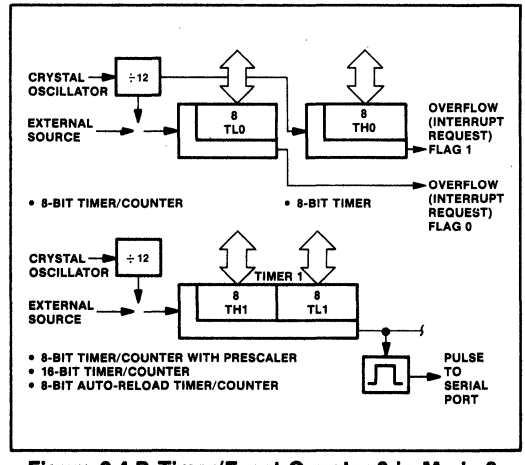


Figure 2.4.B Timer/Event Counter 0 In Mode 3

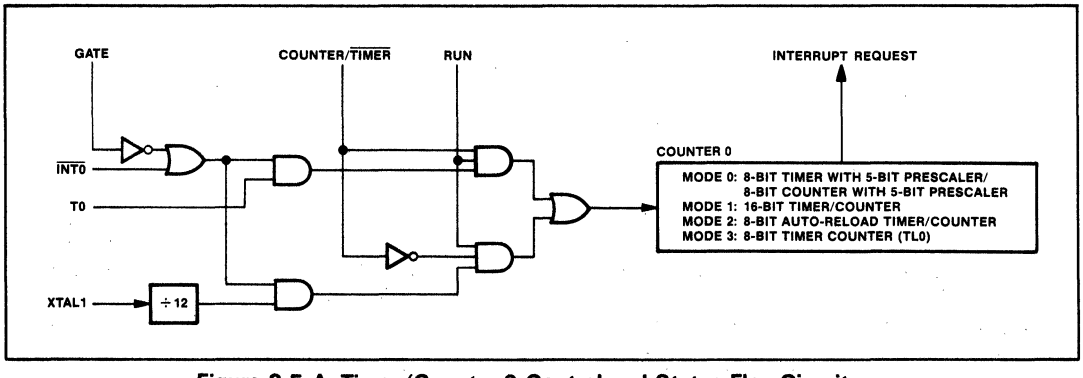


Figure 2.5.A Timer/Counter 0 Control and Status Flag Circuitry

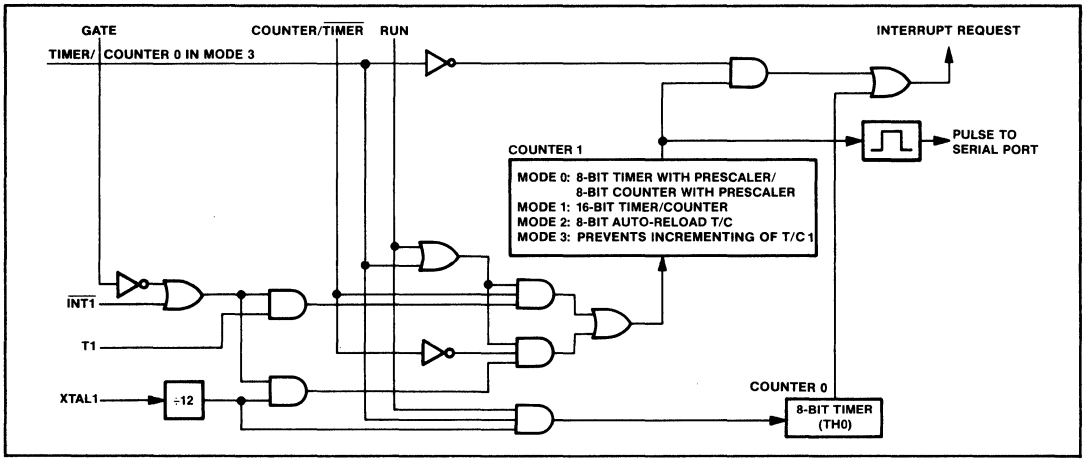


Figure 2.5.B Timer/Counter 1 Control and Status Flag Circuitry

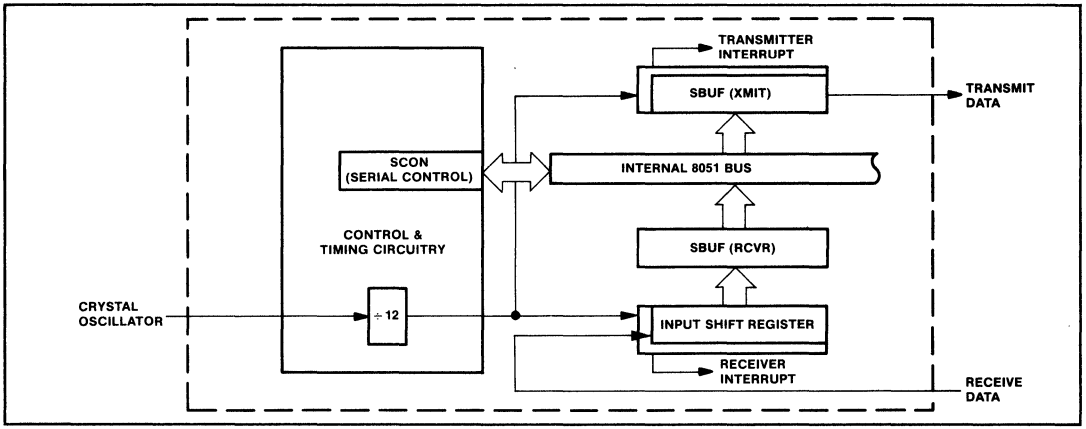


Figure 2.6.A Serial Port—Synchronous Mode 0

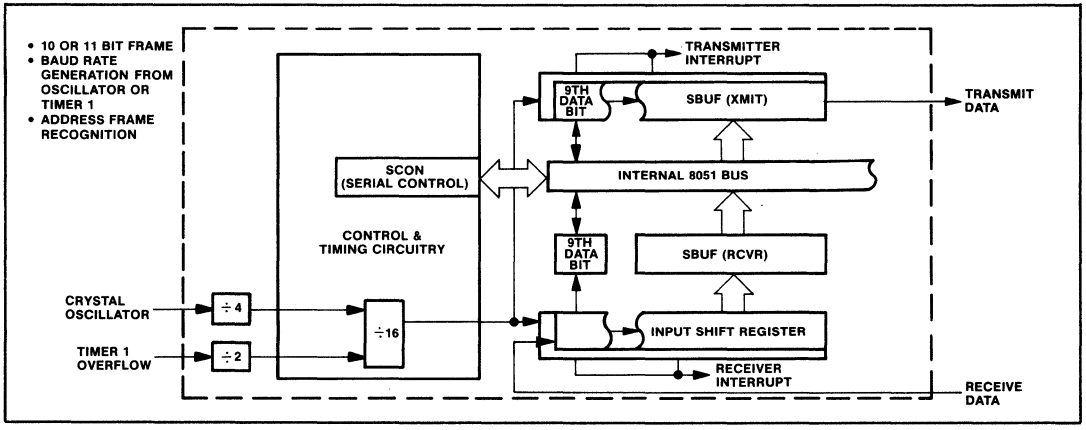


Figure 2.6.B Serial Port—UART Modes 1, 2, and 3

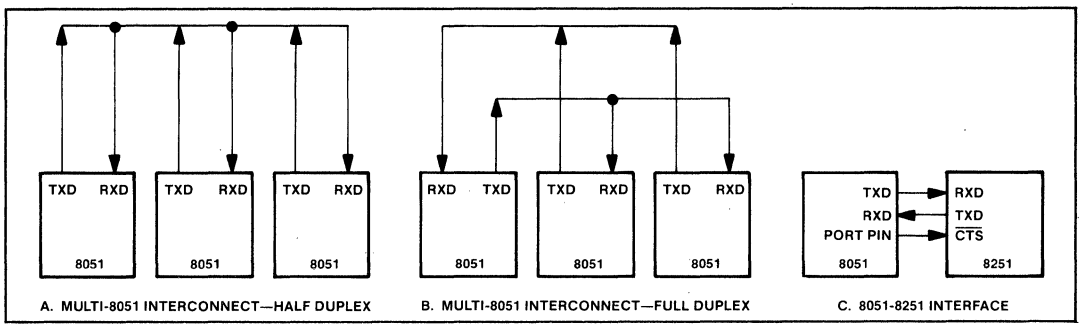


Figure 2.7. UART Interfacing Schemes

Figure 2.7 and a method for I/O expansion is shown in Figure 2.8.

The full-duplex serial I/O port provides asynchronous modes to facilitate communications with standard UART devices, such as printers and CRT terminals, or communications with other 8051s in multi-processor systems. The receiver is double buffered to eliminate the overrun that would occur if the CPU failed to respond to the receiver's interrupt before the beginning of the next frame. Double buffering of the transmitter is not needed since the 8051 can generally maintain the serial link at its maximum rate without it. A minor degradation in transmission rate can occur in rare events such as when the servicing of the transmitter has to wait for a lengthy interrupt service program to complete. In asynchronous modes, false start-bit rejection is provided on received frames. For noise rejection a best two-out-of-three vote is taken on three samples near the center of each received bit.

When interfacing with standard UART devices the serial channel can be programmed to a mode (Mode 1) that transmits/receives a ten-bit frame or programmed to a mode (Mode 2 or 3) that transmits/receives an eleven-bit frame as shown in Figure 2.9. The frame consists of a start bit, eight or nine data bits and a stop bit. In Modes 1 and 3, the transmission-rate timing circuitry receives a pulse from counter 1 each time the counter overflows. The input to counter 1 can be an external source or a division by 12 of the oscillator frequency. The auto-reload mode of the counter provides communication rates of 122 to 31,250 bits per second (including start and stop bits) for a 12 MHz crystal. In Mode 2 the communication rate is a division by 64 of the oscillator frequency yielding a transmission rate of 187,500 bits per second (including start and stop bits) for a 12 MHz crystal.

Distributed processing offers a faster, more powerful system than can be provided by a single CPU processor. This results from a hierarchy of interconnected processors, each with its own memories and

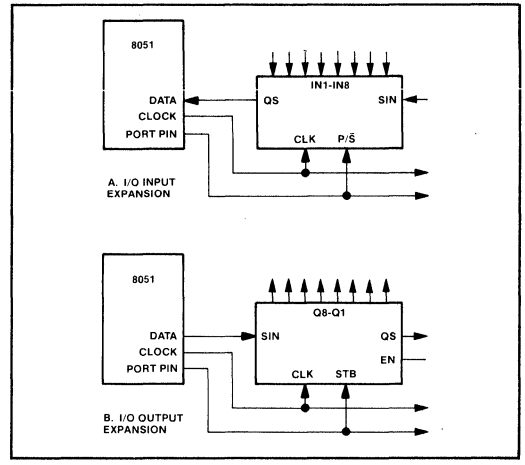


Figure 2.8. I/O Expansion Technique

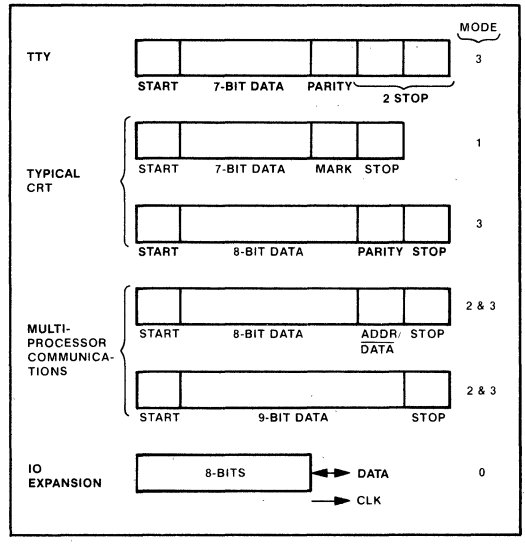


Figure 2.9. Typical Frame Formats

1. Slaves—Configure serial port to interrupt CPU if the received ninth data bit is a one (1).
2. Master—Transmit frame containing address in first 8 data bits and set ninth data bit (i.e. ninth data bit designates address frame).
3. Slaves—Serial port interrupts CPU when address frame is received. Interrupt service program compares received address to its address. The slave which has been addressed reconfigures its serial port to interrupt the CPU on all subsequent transmissions.
4. Master—Transmit control frames and data frames (these will be accepted only by the previously addressed slave).

Figure 2.10. Protocol for Multi-Processor Communications

I/O. In multiprocessing, a host 8051 microcomputer controls a multiplicity of 8051s configured to operate simultaneously on separate portions of the program, each controlling a portion of the overall process. The interconnected 8051s reduce the load on the host processor and result in a low-cost system of data transmission. This form of distributed processing is especially effective in systems where controls in a complex process are required at physically separated locations.

In Modes 2 and 3 the automatic wake-up of slave processors through interrupt driven address-frame recognition is provided to facilitate interprocessor communications. The protocol for interprocessor communications is shown in Figure 2.10.

In synchronous mode (Mode 0) the high speed serial port provides an efficient, low-cost method of expanding I/O lines using standard TTL and CMOS shift registers. The serial channel provides a clock output for synchronizing the shifting of bits to/from an external register. The data rate is a division by 12 of the oscillator frequency and is 1M bits per second at 12 MHz.

2.3 8051 FAMILY PIN DESCRIPTION

V_{SS}
Circuit ground potential.

V_{CC}
+5V power supply during operation, programming and verification.

Port 0
Port 0 is an 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data input and output during programming and verification. Port 0 can sink/source two TTL loads.

Port 1
Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during programming and verification. Port 1 can sink/source one TTL load.

Port 2
Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order 8 bits of address when accessing external memory. It is used for the high-order address and the control signals during programming and verification. Port 2 can sink/source one TTL load.

Port 3
Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a special function must be programmed to a one (1) for that function to operate. Port 3 can sink/source one TTL load. The special functions are assigned to the pins of Port 3, as follows:

—RXD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous).
— TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous).
— $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.
— $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.
— T0 (P3.4). Input to counter 0.
— T1 (P3.5). Input to counter 1.
— $\overline{WR}$ (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.
— $\overline{RD}$ (P3.7). The read control signal enables External Data Memory to Port 0.

RST/V_{PD}
A low to high transition on this pin (at approximately 3V) resets the 8051. If V_{PD} is held within its spec (approximately +5V), while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC}. A small internal resistor permits power-on reset using only a capacitor connected to V_{CC}.

ALE/$\overline{PROG}$
Provides Address Latch Enable output used for latching the address into external memory during normal operation. Receives the program pulse

input during EPROM programming.

$\overline{PSEN}$

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during normal fetch operations.

$\overline{EA}$/V_{DD}

When held at a TTL high level, the 8051 executes instructions from the internal ROM/EPROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instuctions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage.

XTAL1

Input to the oscillator's high gain amplifier. A crystal or external source can be used.

XTAL2

Output from the oscillator's amplifier. Required when a crystal is used.

8051 FAMILY DEVELOPMENT SYSTEM AND SOFTWARE SUPPORT

The 8051 is supported by a total range of Intel development tools. This broad range of support shortens the product development cycle and thus brings the product to market sooner.

- ASM51 Absolute macro assembler for the 8051.
- CONV51 8048 assembly language source code to 8051 assembly source code conversion program.
- EM-51 8051/8751 emulator board that uses a modified 8051 and an EPROM.
- ICE-51™ Real-time in-circuit emulator.
- UPP-8051 PROM programmer personality card.
- 8051 Workshop.

8051 Software Development Package (ASM51 and CONV51)

The 8051 software development package provides development system support for the powerful 8051 family of single chip microcomputers. The package contains a symbolic macro assembler and a 8048 to 8051 source code converter. This diskette-based software package runs under ISIS-II on any Intellec® Microcomputer Development System with 64K bytes of memory.

8051 Macro Assembler (ASM51)

The 8051 macro assembler translates symbolic 8051 assembly language instructions into machine executable object code. These assembly language mnemonics are easier to program and are more readable than binary or hexidecimal machine instructions. Also, by allowing the programmer to give symbolic names to memory locations rather than absolute addresses, software design and debug are performed more quickly and reliably.

ASM51 provides symbolic access for the many useful addressing methods in the 8051 architecture which reference bit, nibble and byte locations.

The assembler supports macro definitions and calls. This provides a convenient means of programming a frequently used code sequence only once. The assembler also provides conditional assembly capabilities. Cross referencing is provided in the symbol table listing, which shows the user the lines in which each symbol was defined and referenced.

If an 8051 program contains errors, the assembler provides a comprehensive set of error diagnostics, which are included in the assembly listing.

The object code generated may be used to program the 8751 EPROM version of the chip or sent to Intel for fabricating the 8051 ROM version. The assembler output can also be debugged using the ICE-51 in-circuit emulator.

8048 to 8051 Assembly Language Converter Utility Program (CONV51)

The 8048 to 8051 assembly language converter is a utility to help users of the MCS-48 family of microcomputers upgrade their designs to the high performance 8051 architecture. By converting 8048 source code to 8051 source code, the investment in software developed for the 8048 is maintained when the system is upgraded.

8051 Emulation Board (EM-51)

The EM-51 8051 emulation board is a small (2.85" x 5.25") board which emulates an 8031/8051/8751 microcomputer using standard PROMs or EPROMs in place of the 8051's on-chip program memory. The board includes a modified 8051 microcomputer, supporting circuits, and two sockets for program memory. The user may select two 2716 EPROMs, a 2732 EPROM, or two 3636 bipolar PROMs depending on crystal frequency and power requirements.

8051 In-Circuit Emulator (ICE-51™)

The 8051 In-Circuit Emulator resides in the Intellec development system. The development system interfaces with the user's 8051 system through an in-cable buffer box with the cable terminating in an 8051 pin-compatible plug. Together these replace the 8051 device in the system. With the emulator plug in place, the designer can exercise the system in real-time while collecting up to 255 instruction

cycles of real-time data. In addition, he can single step the system program.

Static RAM memory is available in the ICE-51 buffer box to emulate the 8051's internal and external program memories and external data memory. The designer can display and alter the contents of the replacement memory in the ICE-51 buffer box, internal 8051 registers, internal data RAM, and Special Function Registers. Symbolic reference capability allows the designer to use meaningful symbols provided by ASM51 rather than absolute values when examining and modifying these memory, register, flag, and I/O locations in his system.

Universal PROM Programmer Personality Card (UPP-851)

The UPP-851 is a personality card for the UPP-103 Universal PROM Programmer. The Universal PROM Programmer is an Intellec system peripheral capable of programming and verifying the 8751 when the UPP-851 is inserted. Programming and verification operations are initiated from the Intellec development system console and are controlled by the Universal PROM Mapper (UPM) program.

8051 Workshop

The workshop provides the design engineer or system designer hands-on experience with the 8051 microcomputers. The course includes explanation of the Intel 8051 architecture, system timing and input/output design. Lab sessions will allow the attendee to gain detailed familiarity with the 8051 family and support tools.

INSITE™ Library

The INSITE Library contains 8051 utilities and applications programs.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0° C to 70° C
Storage Temperature..... -65° C to +150° C
Voltage on Any Pin With
Respect to Ground (V_{SS}) ..... -0.5V to +7V
Power Dissipation ..... 2 Watts

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

D.C. CHARACTERISTICS

CONDITIONS: T_A = 0°C TO 70°C; V_{CC} = 5V ± 5%; V_{SS} = 0V

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V_{IL}	Input Low Voltage (All except XTAL1)	-0.5		0.8	V	
V_{IL1}	Input Low Voltage (XTAL1)	-0.5		TBD	V	
V_{IH}	Input High Voltage (All Except XTAL1, RST/V_{PD})	2.0		V_{CC}+0.5	V	
V_{IH1}	Input High Voltage (XTAL1)	TBD		V_{CC}+0.5	V	
V_{IH2}	Input High Voltage (RST)	3.0		V_{CC} + 0.5	V	
V_{IH3}	Input High Voltage (V_{PD})	4.5		5.5	V	Power Down Only (V_{CC} = 0)
V_{OL}	Output Low Voltage (All Outputs Except Port 0)			0.45	V	I_{OL}=2 mA
V_{OL1}	Output Low Voltage (Port 0)			0.45	V	I_{OL}=4 mA
V_{OH}	Output High Voltage (All Outputs Except Port 0, ALE and $\overline{PSEN}$)	2.4			V	I_{OH}=-100 μA
V_{OH1}	Output High Voltage (ALE and $\overline{PSEN}$, Port 0 In External Bus Mode)	2.4			V	I_{OH}=-400 μA
I_{LO}	Pullup Resistor Current (P1, P2, P3)			-250	μA	.45V≤ V_{IN} ≤V_{CC}
I_{LO1}	Output Leakage Current (P0)			±10	μA	.45V≤V_{IN}≤V_{CC}
I_{CC}	Power Supply Current			150	mA	T_A=25° C
I_{PD}	Power Down Supply Current			20	mA	T_A=25° C, V_{PD}=5V, V_{CC}=0V
C_{IO}	Capacitance Of I/O Buffer			10	pF	fc=1MHz

A.C. CHARACTERISTICS

CONDITIONS: T_A = 0°C TO 70°C; V_{CC}=5V±5% Port 0, ALE and $\overline{PSEN}$ Outputs - C_L = 150 PF; All Other Outputs - C_L = 80PF

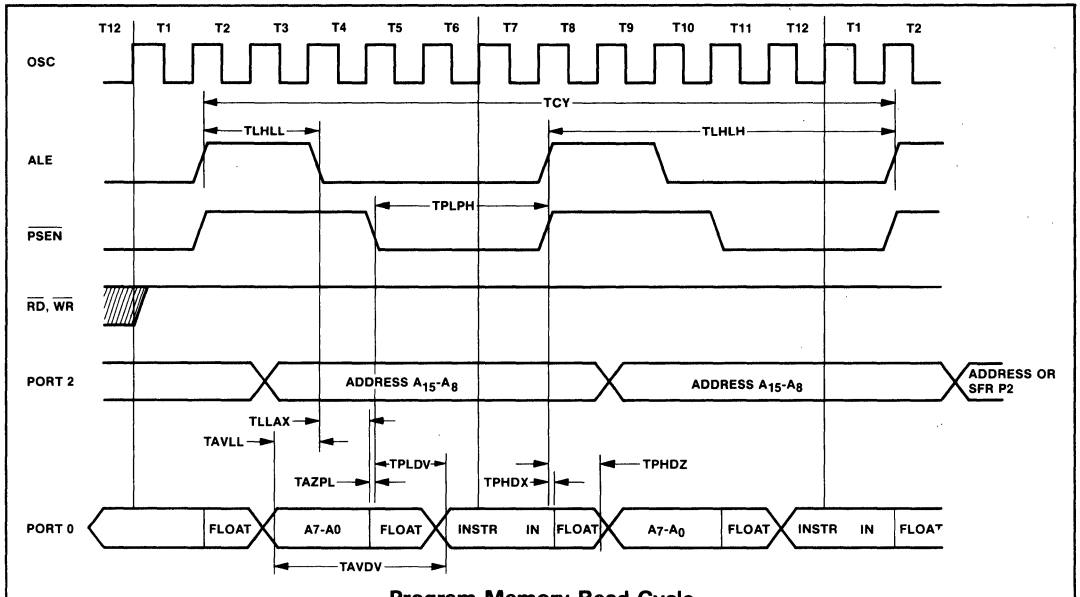
Program Memory Characteristics

Symbol	Parameter	12MHz Clock			Variable Clock 1/TCLCL=1.2 MHz to 12 MHz		
		Min.	Max.	Units	Min.	Max.	Units
TCLCL	Oscillator Period	83		ns			ns
TCY	Min Instruction Cycle Time	1.0		μs	12TCLCL	12TCLCL	ns
TLHLL	ALE Pulse Width	140		ns	2TCLCL-30		ns
TAVLL	Address Set Up To ALE	45		ns	TCLCL-40		ns
TLLAX	Address Hold After ALE	50		ns	TCLCL-35		ns
TPLPH	$\overline{PSEN}$ Width	230		ns	3TCLCL-20		ns
TLHLH	$\overline{PSEN}$, ALE Cycle Time	500		ns	6TCLCL		ns
TPLDV	$\overline{PSEN}$ To Valid Data In		150	ns		3TCLCL-100	ns
TPHDX	Input Data Hold After $\overline{PSEN}$	0		ns	0		ns
TPHDZ	Input Data Float After $\overline{PSEN}$		75	ns		TCLCL-10	ns
TAVDV	Address To Valid Data In		320	ns		5TCLCL-100	ns
TAZPL	Address Float To $\overline{PSEN}$	0		ns	0		ns

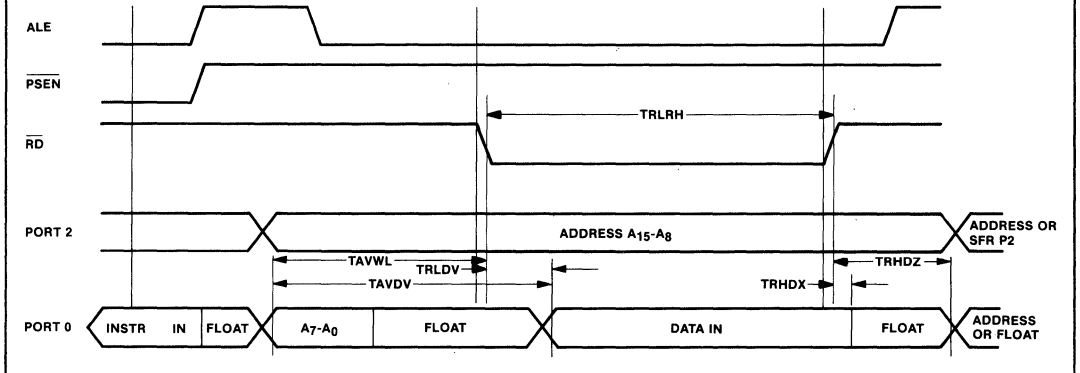
External Data Memory Characteristics

Symbol	Parameter	12MHz Clock			Variable Clock		
		Min.	Max.	Units	Min.	Max.	Units
TRLRH	$\overline{RD}$ Pulse Width	400		ns	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	400		ns	6TCLCL-100		ns
TRLDV	$\overline{RD}$ To Valid Data In		250	ns		5TCLCL-170	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns	0		ns
TRHDZ	Data Float After $\overline{RD}$		100	ns		2TCLCL-70	ns
TAVDV	Address To Valid Data In		600	ns		9TCLCL-150	ns
TAVWL	Address To $\overline{WR}$ or $\overline{RD}$	200		ns	4TCLCL-130		ns
TQVWH	Data Setup Before $\overline{WR}$	400		ns	7TCLCL-180		ns
TWHQX	Data Held After $\overline{WR}$	80		ns	2TCLCL-90		ns

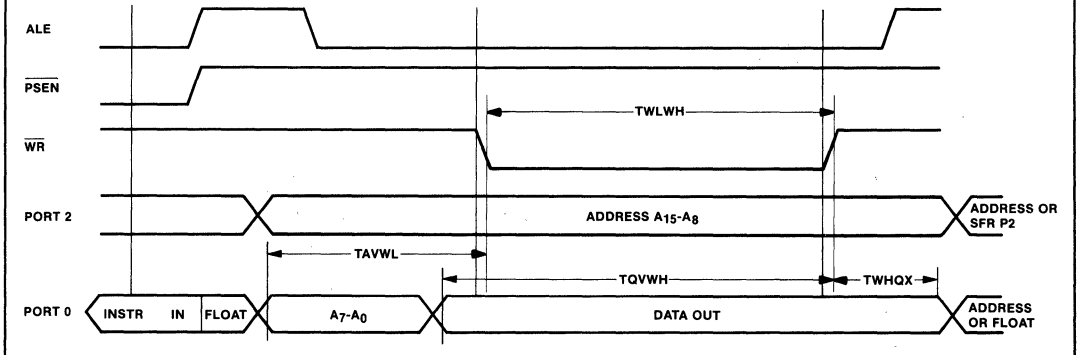
NOTE:
There are 2 to 8 ALE cycles per instruction. Clocks and state timing are shown on the timing diagram for reference purposes only. They are not accessible outside the package. TCY is the minimum instruction cycle time which consists of 12 oscillator clocks or two ALE cycles. Address setup and hold time from ALE are the same for data and program memory.



Program Memory Read Cycle



Data Memory Read Cycle



Data Memory Write Cycle

TABLE 2-1 8051 INSTRUCTION SET SUMMARY

Notes on instruction set and addressing modes:

- **Rn** —Register R7-R0 of the currently selected Register Bank.
- **data** —8-bit internal data location's address. This could be an Internal Data Ram location (0-127) or a SFR (i.e. I/O port, control register, status register, etc. (128-255).
- **@Ri** —8-bit Internal Data RAM location (0-255) addressed indirectly through register R1 or R0.
- **#data** —8-bit constant included in instruction.
- **#data16** —16-bit constant included in instruction.
- **addr16** —16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.
- **addr11** —11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
- **rel** —Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
- **bit** —Direct Addressed bit in Internal Data RAM or Special Function Register.
- ***** —New operation not provided by 8048/8049.

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request, push the PC and to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7 μs @12MHz).

INSTRUCTIONS THAT AFFECT FLAG SETTINGS[1]

INSTRUCTION	FLAG			INSTRUCTION	FLAG		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

[1]Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e. the PSW or bits in the PSW) will also affect flag settings.

Data Transfer

Mnemonic	Description	Bytes	Oscillator Periods
MOV A,Rn	Move register to A	1	12
*MOV A,data	Move direct byte to A	2	12
MOV A,@Ri	Move indirect RAM to A	1	12
MOV A,#data	Move immediate data to A	2	12
MOV Rn,A	Move A to register	1	12
*MOV Rn,data	Move direct byte to register	2	24
MOV Rn,#data	Move immediate data to register	2	12
*MOV data,A	Move A to direct byte	2	12
*MOV data,Rn	Move register to direct byte	2	24
*MOV data,data	Move direct byte to direct byte	3	24
*MOV data,@Ri	Move indirect RAM to direct byte	2	24
*MOV data,#data	Move immediate data to direct byte	3	24
MOV @Ri,A	Move A to indirect RAM	1	12
*MOV @Ri,data	Move direct byte to indirect RAM	2	24
MOV @Ri,#data	Move immediate data to indirect RAM	2	12
*MOV DPTR, #data16	Move 16-bit constant to Data Pointer	3	24
*MOV C,bit	Move direct bit to carry	2	12
*MOV bit,C	Move carry to direct bit	2	24
*MOVC A,@A+ DPTR	Move Program Memory byte addressed by A+DPTR to A	1	24
*MOVC A,@A+PC	Move Program Memory byte addressed by A+PC to A	1	24
MOVX A,@Ri	Move External Data (8-bit address) to A	1	24
*MOVX A,@DPTR	Move External Data (16-bit address) to A	1	24
MOVX @Ri,A	Move A to External Data (8-bit address)	1	24
*MOVX @DPTR,A	Move A to External Data (16-bit address)	1	24
*PUSH data	Move direct byte to stack and inc. SP	2	24
*POP data	Move direct byte from stack and dec. SP	2	24
XCH A,Rn	Exchange register with A	1	12
*XCH A,data	Exchange direct byte with A	2	12
XCH A,@Ri	Exchange indirect RAM with A	1	12
XCHD A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	12

Logic

Mnemonic	Description	Bytes	Oscillator Periods
ANL A,Rn	AND register to A	1	12
*ANL A,data	AND direct byte to A	2	12
ANL A,@Ri	AND indirect RAM to A	1	12
ANL A,#data	AND immediate data to A	2	12
*ANL data,A	AND A to direct byte	2	12
*ANL data,#data	AND immediate data to direct byte	3	24
*ANL C,bit	AND direct bit to carry	2	24
*ANL C,/bit	AND complement of direct bit to carry	2	24
ORL A,Rn	OR register to A	1	12
*ORL A,data	OR direct byte to A	2	12
ORL A,@Ri	OR indirect RAM to A	1	12
ORL A,#data	OR immediate data to A	2	12
*ORL data,A	OR A to direct byte	2	12
*ORL data,#data	OR immediate data to direct byte	3	24
*ORL C,bit	OR direct bit to carry	2	24
*ORL C,/bit	OR complement of direct bit to carry	2	24
XRL A,Rn	Exclusive-OR register to A	1	12
*XRL A,data	Exclusive-OR direct byte to A	2	12
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	12
XRL A,#data	Exclusive-OR immediate data to A	2	12
*XRL data,A	Exclusive-OR A to direct byte	2	12
*XRL data,#data	Exclusive-OR immediate data to direct byte	3	24
*SETB C	Set carry	1	12
*SETB bit	Set direct bit	2	12
CLR A	Clear A	1	12
CLR C	Clear carry	1	12
*CLR bit	Clear direct bit	2	12
CPL A	Complement A	1	12
CPL C	Complement carry	1	12
*CPL bit	Complement direct bit	2	12
RL A	Rotate A Left	1	12
RLC A	Rotate A Left through carry	1	12
RR A	Rotate A Right	1	12
RRC A	Rotate A Right through carry	1	12
SWAP A	Rotate A left four (exchange nibbles within A)	1	12

Arithmetic

Mnemonic	Description	Bytes	Oscillator Periods
ADD A,Rn	Add register to A	1	12
*ADD A,data	Add direct byte to A	2	12
ADD A,@Ri	Add indirect RAM to A	1	12
ADD A,#data	Add immediate data to A	2	12
ADDC A,Rn	Add register and carry flag to A	1	12
*ADDC A,data	Add direct byte and carry flag to A	2	12
ADDC A,@Ri	Add indirect RAM and carry flag to A	1	12
ADDC A,#data	Add immediate data and carry flag to A	2	12
*SUBB A,Rn	Subtract register and carry flag from A	1	12
*SUBB A,data	Subtract direct byte and carry flag from A	2	12
*SUBB A,@Ri	Subtract indirect RAM and carry flag from A	1	12
*SUBB A,#data	Subtract immediate data and carry flag from A	2	12
INC A	Increment A	1	12
INC Rn	Increment register	1	12
*INC data	Increment direct byte	2	12
INC @Ri	Increment indirect RAM	1	12
DEC A	Decrement A	1	12
DEC Rn	Decrement register	1	12
*DEC data	Decrement direct byte	2	12
*DEC @Ri	Decrement indirect RAM	1	12
*INC DPTR	Increment Data Pointer	1	24
*MUL AB	Multiply A times B	1	48
*DIV AB	Divide A by B	1	48
DA A	Decimal add Adjust of A	1	12

Other

Mnemonic	Description	Bytes	Oscillator Periods
NOP	No Operation	1	12

Control Transfer (Branch)

Mnemonic	Description	Bytes	Oscillator Periods
AJMP addr 11	Absolute Jump	2	24
*LJMP addr16	Long Jump	3	24
*SJMP rel	Short Jump	2	24
*JMP @A+DPTR	Jump indirect relative to the DPTR	1	24
JZ rel	Jump if A is zero	2	24
JNZ rel	Jump if A is not zero	2	24
JC rel	Jump if carry is set	2	24
JNC rel	Jump if carry is not set	2	24
*JB bit,rel	Jump relative if direct bit is set	3	24
*JNB bit,rel	Jump relative if direct bit is not set	3	24
*JBC bit,rel	Jump relative if direct bit is set, then clear bit	3	24
*CJNE A,data,rel	Compare direct byte to A & Jump if not Eq. See Note a.	3	24
*CJNE A,#data,rel	Compare immed. to A & Jump if not Eq. See Note a.	3	24
*CJNE Rn,#data, rel	Compare immed. to reg & Jump if not Eq. See Note a.	3	24
*CNJE @Ri, #data,rel	Compare immed. to indirect RAM & Jump if not Eq. See Note a.	3	24
DJNZ Rn,rel	Decrement register & Jump if not zero	3	24
*DJNZ data,rel	Decrement direct byte & Jump if not zero	3	24

Note a) Set C if the first operand is less than the second operand; else clear

Control Transfer (Subroutine)

Mnemonic	Description	Bytes	Oscillator Periods
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine Call	1	24
RETI	Return from Interrupt Call	1	24

Compatible MCS-48™ Components

8

2114A
1024 X 4 BIT STATIC RAM

	2114AL-1	2114AL-2	2114AL-3	2114AL-4	2114A-4	2114A-5
Max. Access Time (ns)	100	120	150	200	200	250
Max. Current (mA)	40	40	40	40	70	70

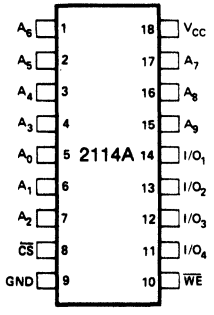
- **HMOS Technology**
- **Low Power, High Speed**
- **Identical Cycle and Access Times**
- **Single +5V Supply ±10%**
- **High Density 18 Pin Package**
- **Completely Static Memory - No Clock or Timing Strobe Required**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Common Data Input and Output Using Three-State Outputs**
- **2114 Upgrade**

The Intel® 2114A is a 4096-bit static Random Access Memory organized as 1024 words by 4-bits using HMOS, a high performance MOS technology. It uses fully DC stable (static) circuitry throughout, in both the array and the decoding, therefore it requires no clocks or refreshing to operate. Data access is particularly simple since address setup times are not required. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

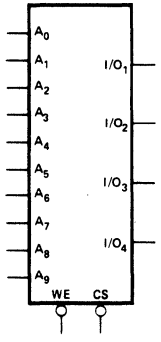
The 2114A is designed for memory applications where the high performance and high reliability of HMOS, low cost, large bit storage, and simple interfacing are important design objectives. The 2114A is placed in an 18-pin package for the highest possible density.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. A separate Chip Select ($\overline{CS}$) lead allows easy selection of an individual package when outputs are or-tied.

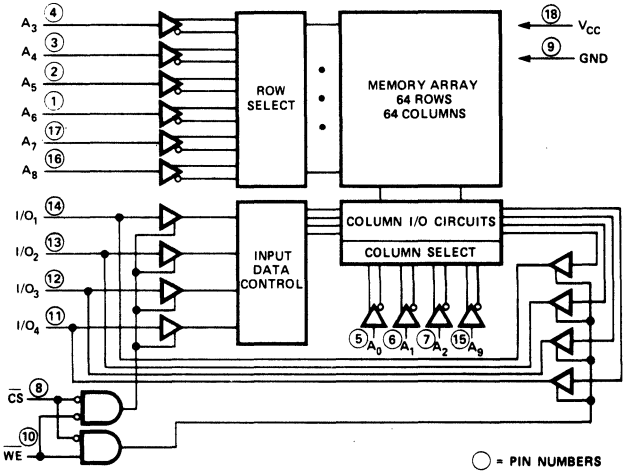
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN NAMES

A_0–A_9	ADDRESS INPUTS	V_{CC} POWER (+5V)
$\overline{WE}$	WRITE ENABLE	GND GROUND
$\overline{CS}$	CHIP SELECT	
I/O_1–I/O_4	DATA INPUT/OUTPUT	

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias ..... -10°C to 80°C
Storage Temperature ..... -65°C to 150°C
Voltage on any Pin
With Respect to Ground ..... -3.5V to +7V
Power Dissipation 1.0W
D.C. Output Current 5mA

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

SYMBOL	PARAMETER	2114AL-1/L-2/L-3/L-4 Min.	Typ.[1]	Max.	2114A-4/-5 Min.	Typ.[1]	Max.	UNIT	CONDITIONS
I_{LI}	Input Load Current (All Input Pins)			10			10	μA	V_{IN} = 0 to 5.5V
$\|I_{LO}\|$	I/O Leakage Current			10			10	μA	$\overline{CS}$ = V_{IH} $V_{I/O}$ = GND to VCC
I_{CC}	Power Supply Current		25	40		50	70	mA	V_{CC} = max, $I_{I/O}$ = 0 mA, T_A = 0°C
V_{IL}	Input Low Voltage	-3.0		0.8	-3.0		0.8	V	
V_{IH}	Input High Voltage	2.0		6.0	2.0		6.0	V	
I_{OL}	Output Low Current	2.1	9.0		2.1	9.0		mA	V_{OL} = 0.4V
I_{OH}	Output High Current	-1.0	-2.5		-1.0	-2.5		mA	V_{OH} = 2.4V
I_{OS}[2]	Output Short Circuit Current			40			40	mA	

NOTE: 1. Typical values are for T_A = 25°C and V_{CC} = 5.0V.
2. Duration not to exceed 30 seconds.

CAPACITANCE

T_A = 25°C, f = 1.0 MHz

SYMBOL	TEST	MAX	UNIT	CONDITIONS
$C_{I/O}$	Input/Output Capacitance	5	pF	$V_{I/O}$ = OV
C_{IN}	Input Capacitance	5	pF	V_{IN} = OV

NOTE: This parameter is periodically sampled and not 100% tested.

A.C. CONDITIONS OF TEST

Input Pulse Levels ..... 0.8 Volt to 2.0 Volt
Input Rise and Fall Times ..... 10 nsec
Input and Output Timing Levels ..... 1.5 Volts
Output Load ..... 1 TTL Gate and C_L = 100 pF

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ± 10%, unless otherwise noted.

READ CYCLE [1]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{RC}	Read Cycle Time	100		120		150		200		250		ns
t_A	Access Time		100		120		150		200		250	ns
t_{CO}	Chip Selection to Output Valid		70		70		70		70		85	ns
t_{CX}	Chip Selection to Output Active	10		10		10		10		10		ns
t_{OTD}	Output 3-state from Deselection		30		35		40		50		60	ns
t_{OHA}	Output Hold from Address Change	15		15		15		15		15		ns

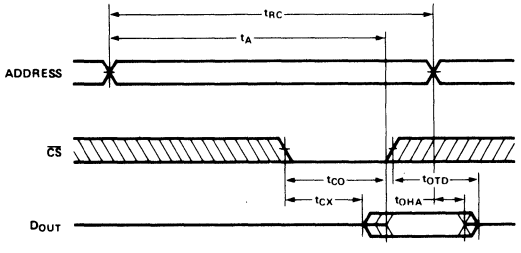
WRITE CYCLE [2]

SYMBOL	PARAMETER	2114AL-1 Min.	2114AL-1 Max.	2114AL-2 Min.	2114AL-2 Max.	2114AL-3 Min.	2114AL-3 Max.	2114A-4/L-4 Min.	2114A-4/L-4 Max.	2114A-5 Min.	2114A-5 Max.	UNIT
t_{WC}	Write Cycle Time	100		120		150		200		250		ns
t_W	Write Time	75		75		90		120		135		ns
t_{WR}	Write Release Time	0		0		0		0		0		ns
t_{OTW}	Output 3-state from Write		30		35		40		50		60	ns
t_{DW}	Data to Write Time Overlap	70		70		90		120		135		ns
t_{DH}	Data Hold from Write Time	0		0		0		0		0		ns

NOTES:
1. A Read occurs during the overlap of a low $\overline{CS}$ and a high $\overline{WE}$.
2. A Write occurs during the overlap of a low $\overline{CS}$ and a low $\overline{WE}$. t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to the earlier of $\overline{CS}$ or $\overline{WE}$ going high.

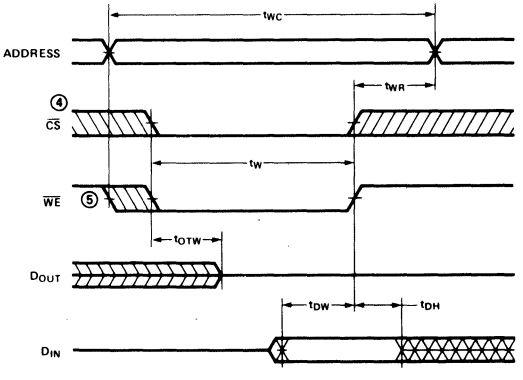
WAVEFORMS

READ CYCLE ③



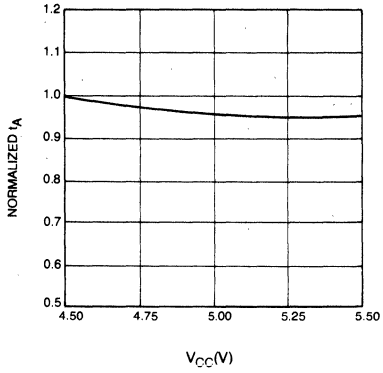
NOTES:
3. $\overline{WE}$ is high for a Read Cycle.
4. If the $\overline{CS}$ low transition occurs simultaneously with the $\overline{WE}$ low transition, the output buffers remain in a high impedance state.
5. $\overline{WE}$ must be high during all address transitions.

WRITE CYCLE

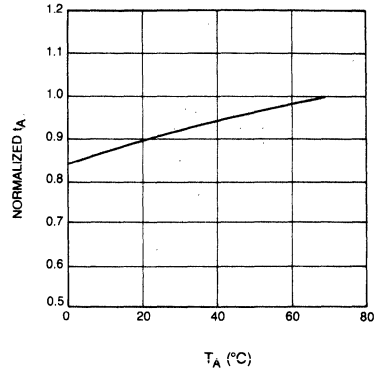


TYPICAL D.C. AND A.C. CHARACTERISTICS

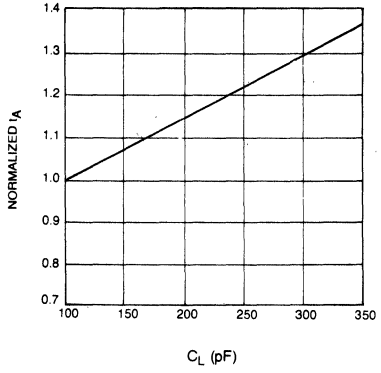
NORMALIZED ACCESS TIME VS. SUPPLY VOLTAGE



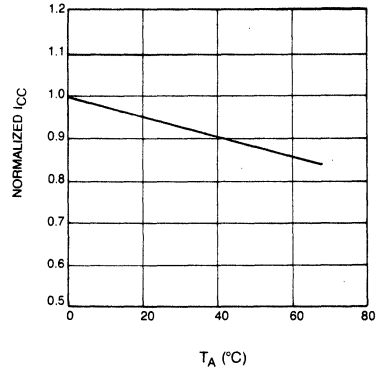
NORMALIZED ACCESS TIME VS. AMBIENT TEMPERATURE



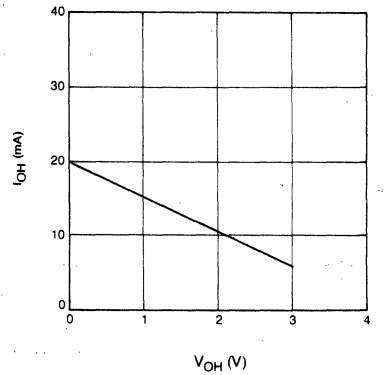
NORMALIZED ACCESS TIME VS. OUTPUT LOAD CAPACITANCE



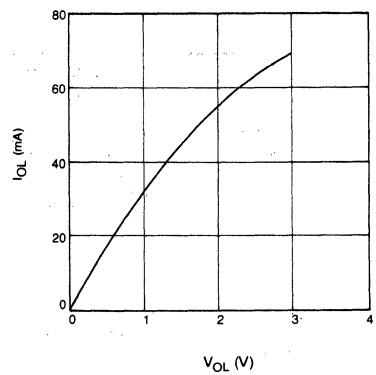
NORMALIZED POWER SUPPLY CURRENT VS. AMBIENT TEMPERATURE



OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



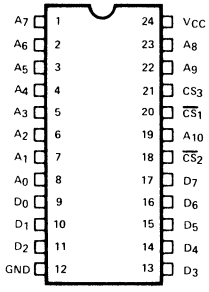
2316E
16K (2K x 8) ROM

- **Fast Access Time—450 ns Max.**
- **Single +5V ± 10% Power Supply**
- **Intel MCS 80 and 85 Compatible**
- **Three Programmable Chip Selects for Simple Memory Expansion and System Interface**
- **EPROM/ROM Pin Compatible for Cost-Effective System Development**
- **Completly Static Operation**
- **Inputs and Outputs TTL Compatible**
- **Three-State Output for Direct Bus Interface**

The Intel® 2316E is a 16,384-bit static, N-channel MOS read only memory (ROM) organized as 2048 words by 8 bits. Its high bit density is ideal for large, non-volatile data storage applications such as program storage. The three-state outputs and TTL input/output levels allow for direct interface with common system bus structures. The 2316E single +5V power supply and 450 ns access time are both ideal for usage with high performance microcomputers such as the Intel MCS™-80 and MCS™-85 devices.

A cost-effective system development program may be implemented by using the pin compatible Intel 2716 16K UV EPROM for prototyping and the lower cost 2616 PROM and 2316E ROM for production. The three 2316E programmable chip selects may be defined by the user and are fixed during the masking process. To simplify the conversion from 2716 prototyping to 2316E production, it is recommended that the 2316E programmable chip select logic levels be defined the same as that shown in the below data sheet pin configuration. This pin configuration and these chip select logic levels are the same as the 2716.

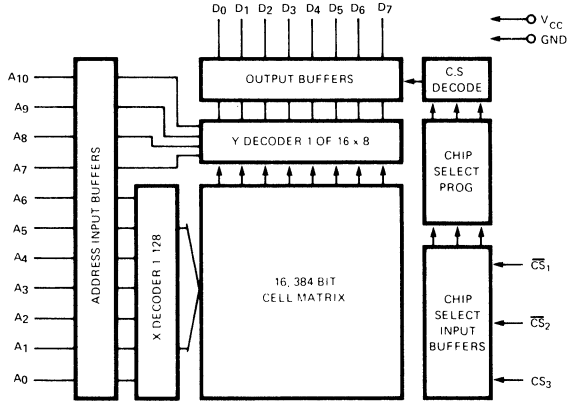
PIN CONFIGURATION



PIN NAMES

A_0–A_{10}	ADDRESS INPUTS
O_7–O_0	DATA OUTPUTS
CS_1–CS_3	CHIP SELECT INPUTS

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . .–10°C to 80°C
Storage Temperature . . . . . .–65°C to +150°C
Voltage On Any Pin With Respect to Ground . . . . . –0.5V to +7V
Power Dissipation . . . . . .1.0 Watt

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±10%, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.(1)	MAX.		
I_{LI}	Input Load Current (All Input Pins)			10	µA	V_{IN} = 0 to 5.25V
I_{LOH}	Output Leakage Current			10	µA	Chip Deselected, V_{OUT} = 4.0V
I_{LOL}	Output Leakage Current			–20	µA	Chip Deselected, V_{OUT} = 0.4V
I_{CC}	Power Supply Current		70	120	mA	All Inputs 5.25V Data Out Open
V_{IL}	Input "Low" Voltage	–0.5		0.8	V	
V_{IH}	Input "High" Voltage	2.4		V_{CC}+1.0V	V	
V_{OL}	Output "Low" Voltage			0.4	V	I_{OL} = 2.1 mA
V_{OH}	Output "High" Voltage	2.4			V	I_{OH} =– 400 µA

NOTE: 1. Typical values for T_A = 25°C and nominal supply voltage.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN.	MAX.	
t_A	Address to Output Delay Time		450	ns
t_{CO}	Chip Select to Output Enable Delay Time		120	ns
t_{DF}	Chip Deselect to Output Data Float Delay Time	10	100	ns

CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

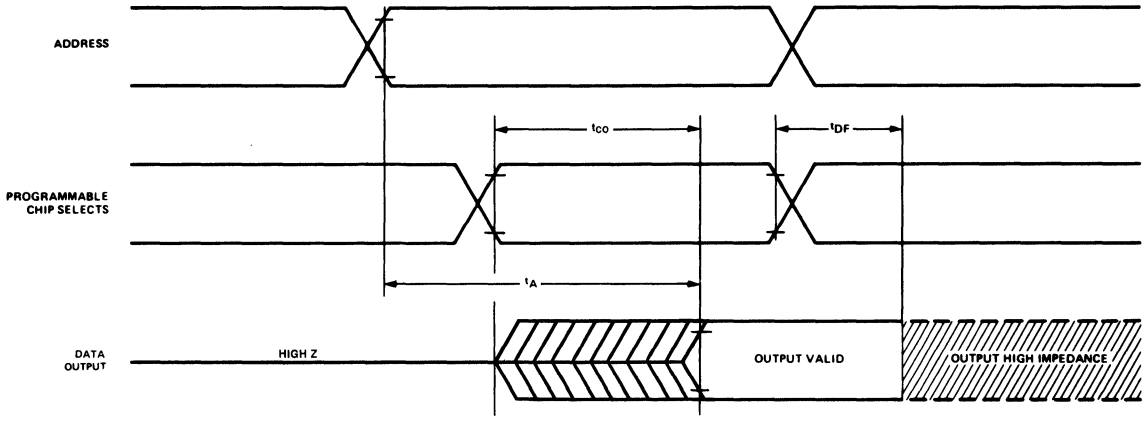
Output Load . . . . . 1 TTL Gate and C_L = 100 pF
Input Pulse Levels . . . . . .0.8 to 2.4V
Input Pulse Rise and Fall Times (10% to 90%) . . . . . 20 ns
Timing Measurement Reference Level
Input . . . . . 1V and 2.2V
Output . . . . . .0.8V and 2.0V

CAPACITANCE(2) T_A = 25°C, f = 1 MHz

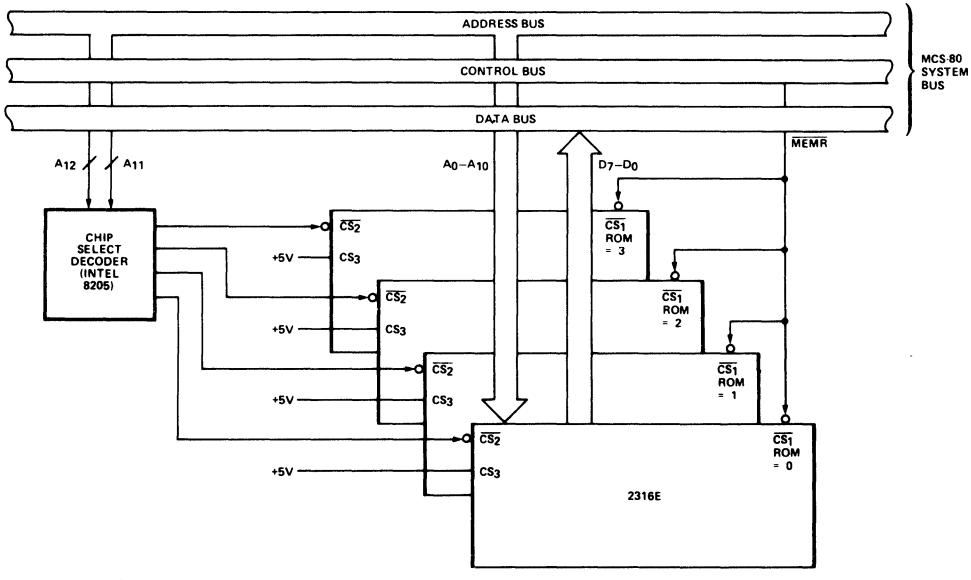
SYMBOL	TEST	LIMITS	
		TYP.	MAX.
C_{IN}	All Pins Except Pin Under Test Tied to AC Ground	5 pF	10 pF
C_{OUT}	All Pins Except Pin Under Test Tied to AC Ground	10 pF	15 pF

NOTE: 2. This parameter is periodically sampled and is not 100% tested.

A.C. Waveforms



Typical System Application (8K × 8 ROM Memory)



2708
8K (1K x 8) UV ERASABLE PROM

	Max. Power	Max. Access
2708	800mW	450ns
2708L	425mW	450ns
2708-1	800mW	350ns
2708-6	800mW	550ns

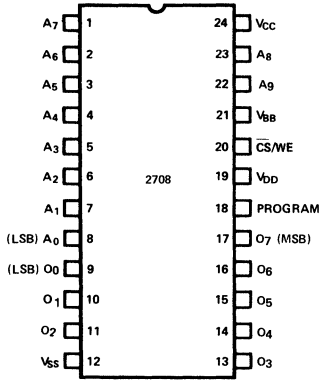
- **Low Power Dissipation — 425 mW Max. (2708L)**
- **Fast Access Time — 350 ns Max. (2708-1)**
- **Static — No Clocks Required**
- **Data Inputs and Outputs TTL Compatible during both Read and Program Modes**
- **Three-State Outputs — OR-Tie Capability**

The Intel® 2708 is an 8192-bit ultraviolet light erasable and electrically reprogrammable EPROM, ideally suited where fast turnaround and pattern experimentation are important requirements. All data inputs and outputs are TTL compatible during both the read and program modes. The outputs are three-state, allowing direct interface with common system bus structures.

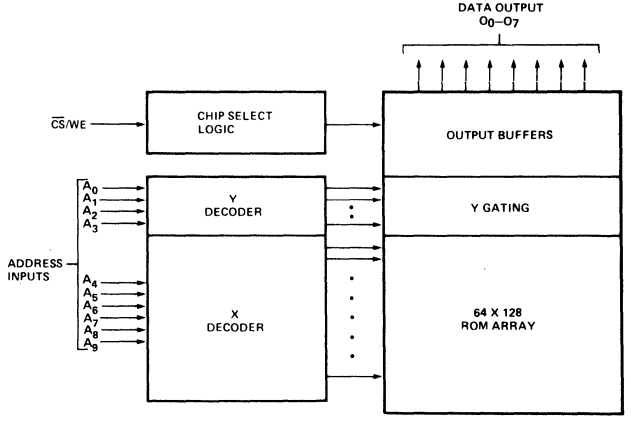
The 2708L at 425mW is available for systems requiring lower power dissipation than from the 2708. A power dissipation savings of over 50% without any sacrifice in speed is obtained with the 2708L. The 2708L has high input noise immunity and is specified at 10% power supply tolerance. A high-speed 2708-1 is also available at 350ns for microprocessors requiring fast access times.

The 2708 family is fabricated with the N-channel silicon gate FAMOS technology and is available in a 24-pin dual in-line package.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

A_0-A_9	ADDRESS INPUTS
O_1-O_8	DATA OUTPUTS/INPUTS
$\overline{CS}$/WE	CHIP SELECT/WRITE ENABLE INPUT

PIN CONNECTION DURING READ OR PROGRAM

	PIN NUMBER							
MODE	DATA I/O 9-11, 13-17	ADDRESS INPUTS 1-8, 22, 23	V_{SS} 12	PROGRAM 18	V_{DD} 19	$\overline{CS}$/WE 20	V_{BB} 21	V_{CC} 24
READ	D_{OUT}	A_{IN}	GND	GND	+12	V_{IL}	-5	+5
DESELECT	HIGH IMPEDANCE	DON'T CARE	GND	GND	+12	V_{IH}	-5	+5
PROGRAM	D_{IN}	A_{IN}	GND	PULSED 26V	+12	V_{IHW}	-5	+5

PROGRAMMING

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions Section.

Absolute Maximum Ratings*

Temperature Under Bias	−25°C to +85°C
Storage Temperature	−65°C to +125°C
V_{DD} With Respect to V_{BB}	+20V to −0.3V
V_{CC} and V_{SS} With Respect to V_{BB}	+15V to −0.3V
All Input or Output Voltages With Respect to V_{BB} During Read	+15V to −0.3V
$\overline{CS}$/WE Input With Respect to V_{BB} During Programming	+20V to −0.3V
Program Input With Respect to V_{BB}	+35V to −0 3V
Power Dissipation	1.5W

*COMMENT

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

D.C. AND A.C. OPERATING CONDITIONS DURING READ

	2708	2708-1	2708-6	2708L
Temperature Range	0°C–70°C	0°C–70°C	0°C–70°C	0°C–70°C
V_{CC} Power Supply	5V ± 5%	5V ± 5%	5V ± 5%	5V ± 10%
V_{DD} Power Supply	12V ± 5%	12V ± 5%	12V ± 5%	12V ± 10%
V_{BB} Power Supply	−5V ± 5%	−5V ± 5%	−5V ± 5%	−5V ± 10%

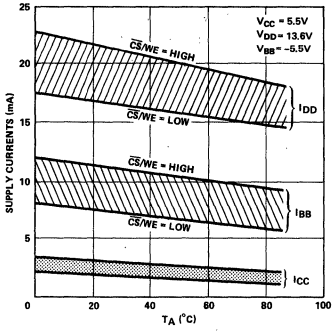
READ OPERATION

D.C. AND OPERATING CHARACTERISTICS

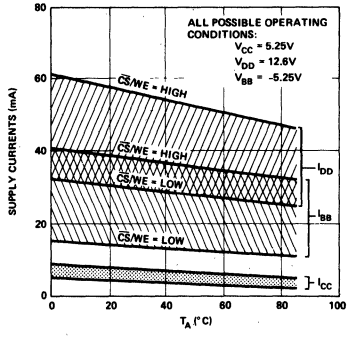
Symbol	Parameter	2708, 2708-1, 2708-6 Limits			2708L Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.	Min.	Typ.[2]	Max.		
I_{LI}	Address and Chip Select Input Sink Current		1	10		1	10	μA	$V_{IN} = 5.25V$ or $V_{IN} = V_{IL}$
I_{LO}	Output Leakage Current		1	10		1	10	μA	$V_{OUT} = 5.5V$, CS/WE = 5V
I_{DD}[3]	V_{DD} Supply Current		50	65		21	28	mA	Worst Case Supply Currents[4]
I_{CC}[3]	V_{CC} Supply Current		6	10		2	4	mA	All Inputs High;
I_{BB}[3]	V_{BB} Supply Current		30	45		10	14	mA	CSWE = 5V; $T_A = 0°C$
V_{IL}	Input Low Voltage	V_{SS}		0.65	V_{SS}		0.65	V	
V_{IH}	Input High Voltage	3.0		$V_{CC}+1$	2.2		$V_{CC}+1$	V	
V_{OL}	Output Low Voltage			0.45			0.4	V	$I_{OL} = 1.6mA$ (2708, 2708-1, 2708-6)
									$I_{OL} = 2mA$ (2708L)
V_{OH1}	Output High Voltage	3.7			3.7			V	$I_{OH} = -100\mu A$
V_{OH2}	Output High Voltage	2.4			2.4			V	$I_{OH} = -1mA$
PD	Power Dissipation			800			325	mW	$T_A = 70°C$
							425	mW	$T_A = 0°C$

NOTES:
1. V_{BB} must be applied prior to V_{CC} and V_{DD}. V_{BB} must also be the last power supply switched off.
2. Typical values are for $T_A = 25°C$ and nominal supply voltages.
3. The total power dissipation is not calculated by summing the various currents (I_{DD}, I_{CC}, and I_{BB}) multiplied by their respective voltages since current paths exist between the various power supplies and V_{SS}. The I_{DD}, I_{CC}, and I_{BB} currents should be used to determine power supply capacity only.
4. I_{BB} for the 2708L is specified in the programmed state and is 18mA maximum in the unprogrammed state.

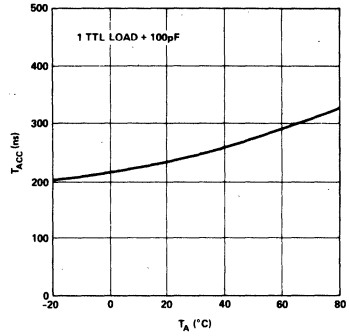
2708L
RANGE OF SUPPLY CURRENTS VS. TEMPERATURE



2708, 2708-1, AND 2708-6
RANGE OF SUPPLY CURRENTS VS. TEMPERATURE



ACCESS TIME VS. TEMPERATURE



A.C. CHARACTERISTICS

Symbol	Parameter	2708, 2708L Limits		2708-1 Limits		2708-6 Limits		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Output Delay		450		350		550	ns
t_{CO}	Chip Select to Output Delay		120		120		160	ns
t_{DF}	Chip Deselect to Output Float	0	120	0	120	0	160	ns
t_{OH}	Address to Output Hold	0		0		0		ns

CAPACITANCE [1] $T_A = 25°C$, f = 1 MHz

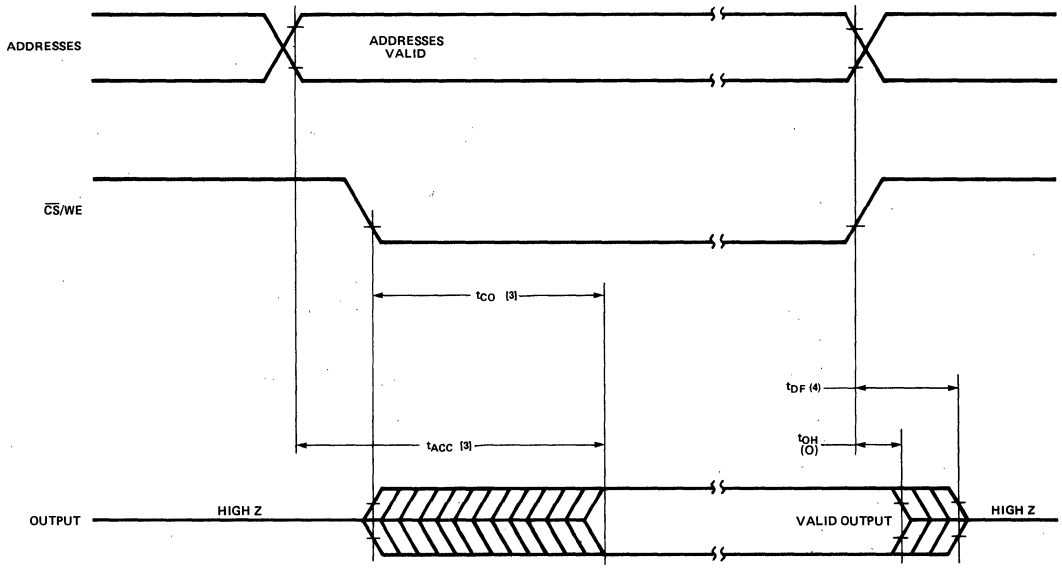
Symbol	Parameter	Typ.	Max.	Unit.	Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

NOTE: 1. This parameter is periodically sampled and is not 100% tested.

A.C. TEST CONDITIONS:

Output Load: 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times: ≤20 ns
Timing Measurement Reference Levels: 0.8V and 2.8V for inputs; 0.8V and 2.4V for outputs.
Input Pulse Levels: 0.65V to 3.0V

A.C. WAVEFORMS [2]



NOTES:

2. ALL TIMES SHOWN IN PARENTHESES ARE MINIMUM AND ARE NSEC UNLESS OTHERWISE SPECIFIED.
3. $\overline{CS}$ MAY BE DELAYED UP TO t_{ACC}-t_{CO} AFTER ADDRESSES ARE VALID WITHOUT IMPACT ON t_{ACC}.
4. t_{DF} IS SPECIFIED FROM $\overline{CS}$ OR ADDRESS CHANGE, WHICHEVER OCCURS FIRST.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2708 family are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical device in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2708 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2708 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog PROM/ROM Programming Instructions Section) for the 2708 family is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The device should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

2716
16K (2K x 8) UV ERASABLE PROM

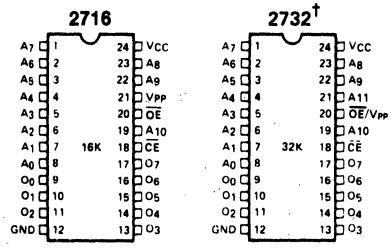
- **Fast Access Time**
  - **— 350 ns Max. 2716-1**
  - **— 390 ns Max. 2716-2**
  - **— 450 ns Max. 2716**
  - **— 490 ns Max. 2716-5**
  - **— 650 ns Max. 2716-6**
- **Single +5V Power Supply**
- **Low Power Dissipation**
  - **— 525 mW Max. Active Power**
  - **— 132 mW Max. Standby Power**
- **Pin Compatible to Intel® 2732 EPROM**
- **Simple Programming Requirements**
  - **— Single Location Programming**
  - **— Programs with One 50 ms Pulse**
- **Inputs and Outputs TTL Compatible during Read and Program**
- **Completely Static**

The intel® 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features fast single address location programming. It makes designing with EPROMs faster, easier and more economical.

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with the newer high performance +5V microprocessors such as Intel's 8085 and 8086. A selected 2716-5 and 2716-6 is available for slower speed applications. The 2716 is also the first EPROM with a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 has the simplest and fastest method yet devised for programming EPROMs – single pulse TTL level programming. No need for high voltage pulsing because all programming controls are handled by TTL signals. Program any location at any time—either individually, sequentially or at random, with the 2716's single address location programming. Total programming time for all 16,384 bits is only 100 seconds.

PIN CONFIGURATION



†Refer to 2732 data sheet for specifications

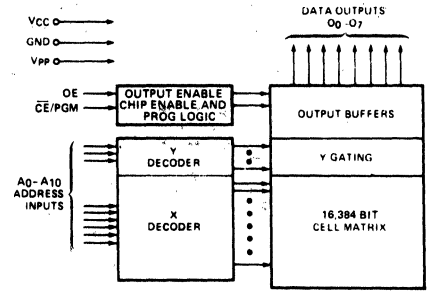
PIN NAMES

A_0–A_{10}	ADDRESSES
$\overline{CE}$/PGM	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
O_0–O_7	OUTPUTS

MODE SELECTION

MODE \ PINS	$\overline{CE}$/PGM (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

BLOCK DIAGRAM



PROGRAMMING

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions Section.

Absolute Maximum Ratings*

Temperature Under Bias . . . . . -10°C to +80°C
Storage Temperature . . . . . -65°C to +125°C
All Input or Output Voltages with
Respect to Ground . . . . . +6V to -0.3V
V_{PP} Supply Voltage with Respect
to Ground During Program . . . . . +26.5V to -0.3V

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

DC and AC Operating Conditions During Read

	2716	2716-1	2716-2	2716-5	2716-6
Temperature Range	0°C – 70°C	0°C – 70°C	0°C – 70°C	0°C – 70°C	0°C – 70°C
V_{CC} Power Supply[1,2]	5V ±5%	5V ±10%	5V ±5%	5V ±5%	5V ±5%
V_{PP} Power Supply[2]	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}

READ OPERATION

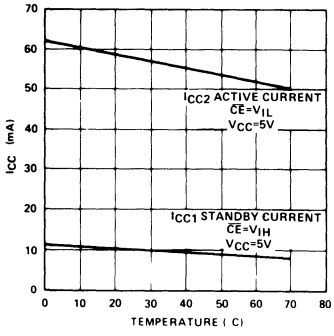
D.C. and Operating Characteristics

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{PP1}[2]	V_{PP} Current			5	mA	V_{PP} = 5.25V
I_{CC1}[2]	V_{CC} Current (Standby)		10	25	mA	$\overline{CE}$ = V_{IH}, $\overline{OE}$ = V_{IL}
I_{CC2}[2]	V_{CC} Current (Active)		57	100	mA	$\overline{OE}$ = $\overline{CE}$ = V_{IL}
V_{IL}	Input Low Voltage	-0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -400 μA

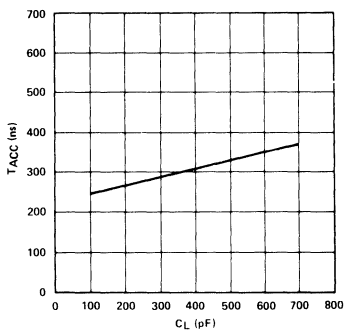
NOTES: 1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for T_A = 25°C and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested.

Typical Characteristics

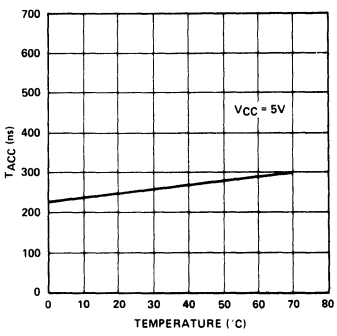
I_{CC} CURRENT vs. TEMPERATURE



ACCESS TIME vs. CAPACITANCE



ACCESS TIME vs. TEMPERATURE



A.C. Characteristics

Symbol	Parameter	Limits (ns)										Test Conditions
		2716		2716-1		2716-2		2716-5		2716-6		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Output Delay		450		350		390		450		450	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	$\overline{CE}$ to Output Delay		450		350		390		490		650	$\overline{OE} = V_{IL}$
t_{OE}	Output Enable to Output Delay		120		120		120		160		200	$\overline{CE} = V_{IL}$
t_{DF}	Output Enable High to Output Float	0	100	0	100	0	100	0	100	0	100	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Addresses, $\overline{CE}$ or $\overline{OE}$ Whichever Occurred First	0		0		0		0		0		$\overline{CE} = \overline{OE} = V_{IL}$

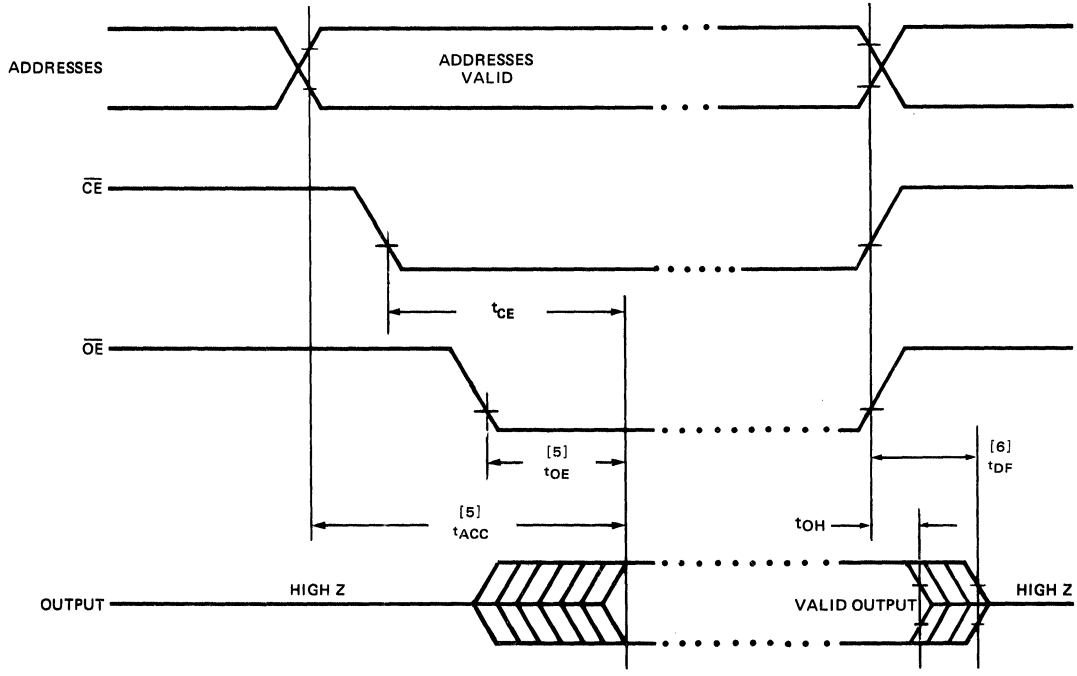
Capacitance [4] $T_A = 25°C$, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

A.C. Test Conditions:

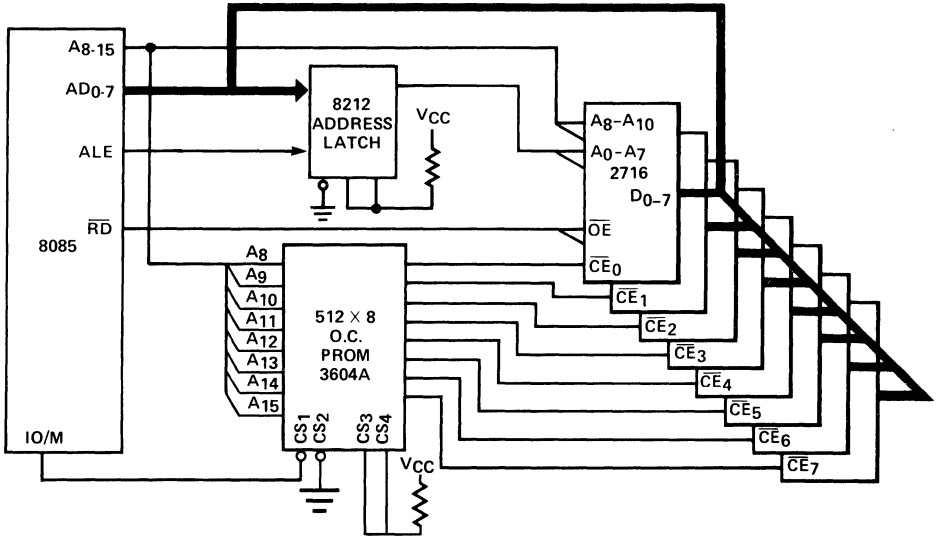
Output Load: 1 TTL gate and $C_L = 100$ pF
Input Rise and Fall Times: $\leqslant$20 ns
Input Pulse Levels: 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs 1V and 2V
Outputs 0.8V and 2V

A. C. Waveforms [1]



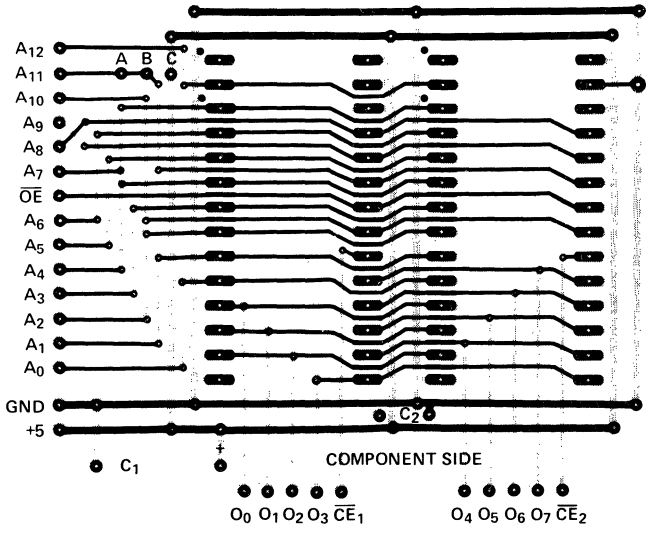
NOTE:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. Typical values are for $T_A = 25°C$ and nominal supply voltages.
4. This parameter is only sampled and is not 100% tested.
5. OE may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of CE without impact on t_{ACC}.
6. t_{DF} is specified from OE or CE, whichever occurs first.

TYPICAL 16K EPROM SYSTEM



- This scheme accomplished by using $\overline{CE}$ (PD) as the primary decode. $\overline{OE}$ (CS) is now controlled by previously unused signal. RD now controls data on and off the bus by way of $\overline{OE}$.
- A selected 2716 is available for systems which require $\overline{CE}$ access of less than 450 ns for decode network operation.
- The use of a PROM as a decoder allows for:
  a) Compatibility with upward (and downward) memory expansion.
  b) Easy assignment of ROM memory modules, compatible with PL/M modular software concepts.

8K, 16K, 32K, 64K 5V EPROM/ROM FAMILY PRINTED CIRCUIT BOARD LAYOUT



ERASURE CHARACTERISTICS

The erasure characteristics of the 2716 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximatley 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog PROM/ROM Programming Instruction Section) for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

DEVICE OPERATION

The five modes of operation of the 2716 are listed in Table I. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other two modes.

TABLE I. MODE SELECTION

MODE \ PINS	$\overline{CE}$/PGM (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

READ MODE

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs 120 ns (t_{OE}) after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

STANDBY MODE

The 2716 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedence state, independent of the $\overline{OE}$ input.

OUTPUT OR-TIEING

Because 2716's are usually used in larger memory arrays, Intel has provided a 2 line control function that accomodates this use of multiple memory connections. The two line control function allows for:

a) the lowest possible memory power dissipation, and
b) complete assurance that output bus contention will not occur.

To most efficiently use these two control lines, it is recommended that $\overline{CE}$ (pin 18) be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) be made a common connection to all devices in the array and connected to the READ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are only active when data is desired from a particular memory device.

PROGRAMMING

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{OE}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active high, TTL program pulse is applied to the $\overline{CE}$/PGM input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the $\overline{CE}$/PGM input.

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the $\overline{CE}$/PGM input programs the paralleled 2716s.

PROGRAM INHIBIT

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for $\overline{CE}$/PGM, all like inputs (including $\overline{OE}$) of the parallel 2716s may be common. A TTL level program pulse applied to a 2716's $\overline{CE}$/PGM input with V_{PP} at 25V will program that 2716. A low level $\overline{CE}$/PGM input inhibits the other 2716 from being programmed.

PROGRAM VERIFY

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed wth V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

2732A
32K (4K x 8) UV ERASABLE PROM

- **200ns (2732A-2) Maximum Access Time . . . HMOS*-E Technology**
- **Compatible to High Speed 8mHz 8086-2 MPU . . .Zero WAIT State**
- **Two Line Control**
- **Pin Compatible to 2764 EPROM**
- **Industry Standard Pinout . . . JEDEC Approved**
- **Low Standby Current . . . 35mA Max.**

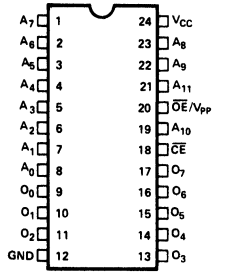
The Intel 2732A is a 5V only, 32,384 bit ultraviolet erasable and electrically programmable read-only memory (EPROM). It is pin compatible to Intel's 450ns 2732. The standard 2732A's access time is 250ns with speed selection (2732A-2) available at 200ns. The access time is compatible to high performance microprocessors, such as the 8mHz 8086-2. In these systems, the 2732A allows the microprocessor to operate without the addition of WAIT states.

An important 2732A feature is the separate output control, Output Enable ($\overline{OE}$), from the Chip Enable control ($\overline{CE}$). The $\overline{OE}$ control eliminates bus contention in multiple bus microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's EPROMs. AP-72 is available from Intel's Literature Department.

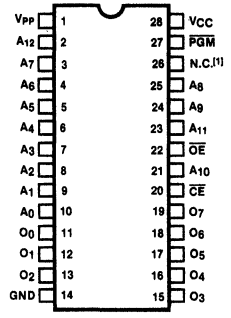
The 2732A has a standby mode which reduces the power dissipation without increasing access time. The maximum active current is 150mA, while the maximum standby current is only 35mA, a 75% saving. The standby mode is achieved by applying a TTL-high signal to the $\overline{CE}$ input.

The 2732A is fabricated with HMOS*-E technology, Intel's high speed N-channel MOS Silicon Gate Technology.

2732A PIN CONFIGURATION



2764 PIN CONFIGURATION



[1]For total compatibility from 2732A provide a trace to pin 26

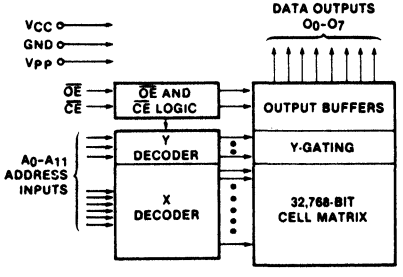
MODE SELECTION

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	High Z
Program	V_{IL}	V_{PP}	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+5	D_{OUT}
Program Inhibit	V_{IH}	V_{PP}	+5	High Z

PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS

BLOCK DIAGRAM



*HMOS is a patented process of Intel Corporation.

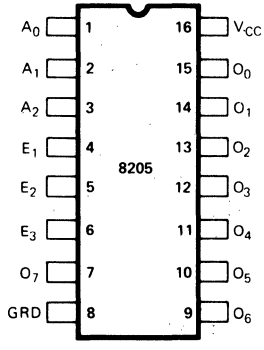
8205
HIGH SPEED 1 OUT OF 8 BINARY DECODER

- **I/O Port or Memory Selector**
- **Simple Expansion — Enable Inputs**
- **High Speed Schottky Bipolar Technology — 18 ns Max Delay**
- **Directly Compatible with TTL Logic Circuits**
- **Low Input Load Current — 0.25 mA Max, 1/6 Standard TTL Input Load**
- **Minimum Line Reflection — Low Voltage Diode Input Clamp**
- **Outputs Sink 10 mA Min**
- **16-Pin Dual In-Line Ceramic or Plastic Package**

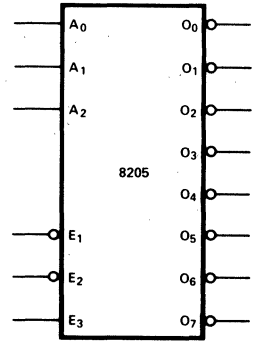
The Intel® 8205 decoder can be used for expansion of systems which utilize input ports, output ports, and memory components with active low chip select input. When the 8205 is enabled, one of its 8 outputs goes "low", thus a single row of a memory system is selected. The 3-chip enable inputs on the 8205 allow easy system expansion. For very large systems, 8205 decoders can be cascaded such that each decoder can drive 8 other decoders for arbitrary memory expansions.

The 8205 is packaged in a standard 16-pin dual in-line package, and its performance is specified over the temperature range of 0°C to +75°C, ambient. The use of Schottky barrier diode clamped transistors to obtain fast switching speeds results in higher performance than equivalent devices made with a gold diffusion process.

PIN CONFIGURATION



LOGIC SYMBOL



PIN NAMES

A_0-A_2	ADDRESS INPUTS
$\overline{E_1}$-$\overline{E_3}$	ENABLE INPUTS
$\overline{O_0}$-$\overline{O_7}$	DECODED OUTPUTS

ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

FUNCTIONAL DESCRIPTION

Decoder

The 8205 contains a one out of eight binary decoder. It accepts a three bit binary code and by gating this input, creates an exclusive output that represents the value of the input code.

For example, if a binary code of 101 was present on the A0, A1 and A2 address input lines, and the device was enabled, an active low signal would appear on the $\overline{O5}$ output line. Note that all of the other output pins are sitting at a logic high, thus the decoded output is said to be exclusive. The decoders outputs will follow the truth table shown below in the same manner for all other input variations.

Enable Gate

When using a decoder it is often necessary to gate the outputs with timing or enabling signals so that the exclusive output of the decoded value is synchronous with the overall system.

The 8205 has a built-in function for such gating. The three enable inputs ($\overline{E1}$, $\overline{E2}$, E3) are ANDed together and create a single enable signal for the decoder. The combination of both active "high" and active "low" device enable inputs provides the designer with a powerfully flexible gating function to help reduce package count in his system.

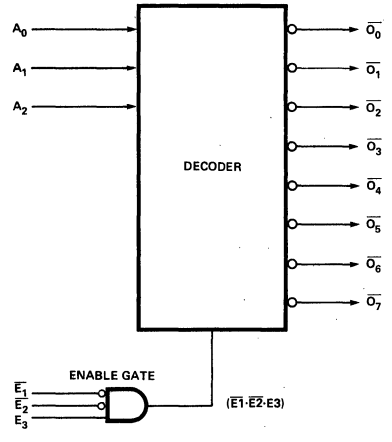


Figure 1. Enable Gate

ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

APPLICATIONS OF THE 8205

The 8205 can be used in a wide variety of applications in microcomputer systems. I/O ports can be decoded from the address bus, chip select signals can be generated to select memory devices and the type of machine state such as in 8008 systems can be derived from a simple decoding of the state lines (S0, S1, S2) of the 8008 CPU.

I/O Port Decoder

Shown in the figure below is a typical application of the 8205. Address input lines are decoded by a group of 8205s (3). Each input has a binary weight. For example, A0 is assigned a value of 1 and is the LSB; A4 is assigned a value of 16 and is the MSB. By connecting them to the decoders as shown, an active low signal that is exclusive in nature and represents the value of the input address lines, is available at the outputs of the 8205s.

This circuit can be used to generate enable signals for I/O ports or any other decoder related application.

Note that no external gating is required to decode up to 24 exclusive devices and that a simple addition of an inverter or two will allow expansion to even larger decoder networks.

Chip Select Decoder

Using a very similar circuit to the I/O port decoder, an array of 8205s can be used to create a simple interface to a 24K memory system.

The memory devices used can be either ROM or RAM and are 1K in storage capacity. 8308s and 8102s are the devices typically used for this application. This type of memory device has ten (10) address inputs and an active "low" chip select ($\overline{CS}$). The lower order address bits A0-A9 which come from the microprocessor are "bussed" to all memory elements and the chip select to enable a specific device or group of devices comes from the array of 8205s. The output of the 8205 is active low so it is directly compatible with the memory components.

Basic operation is that the CPU issues an address to identify a specific memory location in which it wishes to "write" or "read" data. The most significant address bits A10-A14 are decoded by the array of 8205s and an exclusive, active low, chip select is generated that enables a specific memory device. The least significant address bits A0-A9 identify a specific location within the selected device. Thus, all addresses throughout the entire memory array are exclusive in nature and are non-redundant.

This technique can be expanded almost indefinitely to support even larger systems with the addition of a few inverters and an extra decoder (8205).

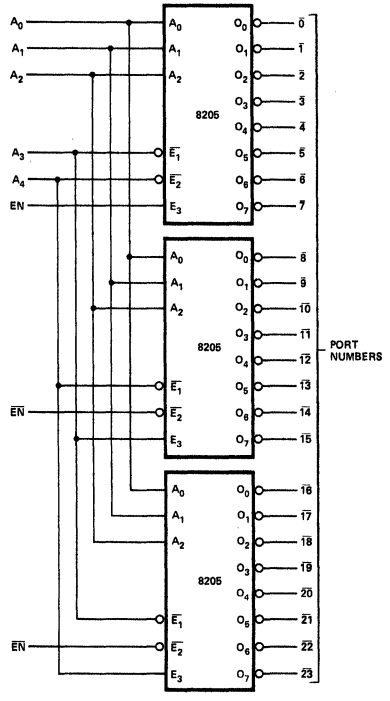


Figure 2. I/O Port Decoder

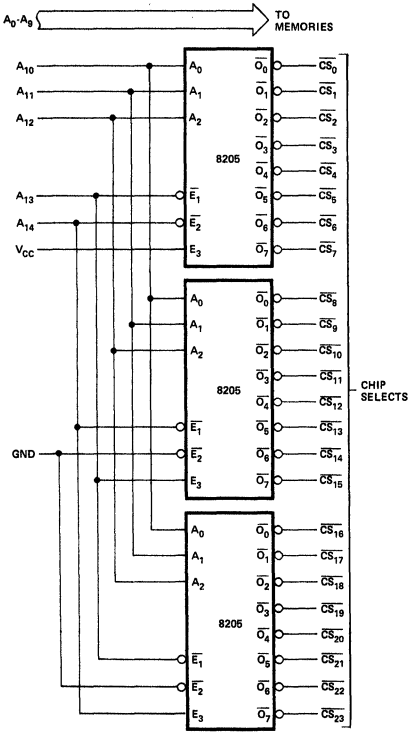


Figure 3. 32K Memory Interface

Logic Element Example

Probably the most overlooked application of the 8205 is that of a general purpose logic element. Using the "on-chip" enabling gate, the 8205 can be configured to gate its decoded outputs with system timing signals and generate strobes that can be directly connected to latches, flip-flops and one-shots that are used throughout the system.

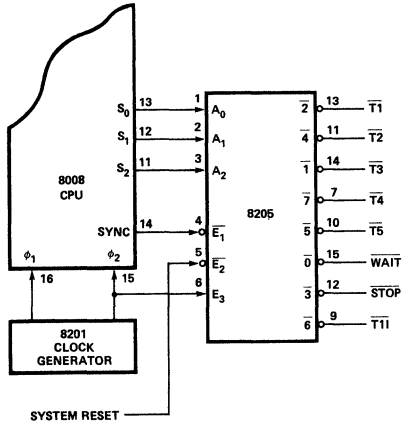
An excellent example of such an application is the "state decoder" in an 8008 CPU based system. The 8008 CPU issues three bits of information (S0, S1, S2) that indicate the nature of the data on the Data Bus during each machine state. Decoding of these signals is vital to generate strobes that can load the address latches, control bus discipline and general machine functions.

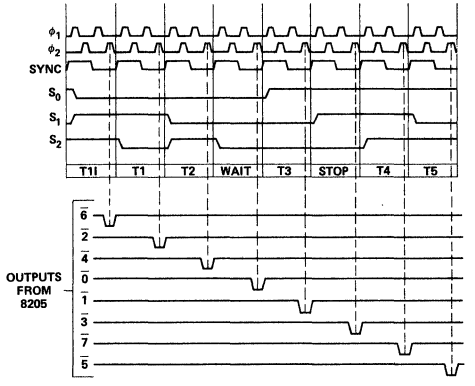
In the figure below a circuit is shown using the 8205 as the "state decoder" for an 8008 CPU that not only decodes the S0, S1, S2 outputs but gates these signals with the clock (phase 2) and the SYNC output of the 8008 CPU. The $\overline{T1}$ and $\overline{T2}$ decoded strobes can connect directly to devices like 8212s for latching the address information. The other decoded strobes can be used to generate signals to control the system data bus, memory timing functions and interrupt structure. RESET is connected to the enable gate so that strobes are not generated during system reset, eliminating accidental loading.

The power of such a circuit becomes evident when a single decoded strobe is logically broken down. Consider $\overline{T1}$ output, the boolean equation for it would be:

$$\overline{T1} = (\overline{S0}\cdot S1\cdot\overline{S2})\cdot(\overline{SYNC}\cdot \text{Phase 2}\cdot\overline{\text{Reset}})$$

A six input NAND gate plus a few inverters would be needed to implement this function. The seven remaining outputs would need a similar circuit to duplicate their function, obviously a substantial savings in components can be achieved when using such a technique.





State Control Coding

S_0	S_1	S_2	STATE
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	WAIT
1	0	0	T3
1	1	0	STOP
1	1	1	T4
1	0	1	T5

Figure 4. 8205 State Decoder Circuit

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias:	Ceramic	−65°C to +125°C
	Plastic	−65°C to +75°C
Storage Temperature		−65°C to +160°C
All Output or Supply Voltages		−0.5 to +7 Volts
All Input Voltages		−1.0 to +5.5 Volts
Output Currents		125 mA

*COMMENT

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

D.C. CHARACTERISTICS

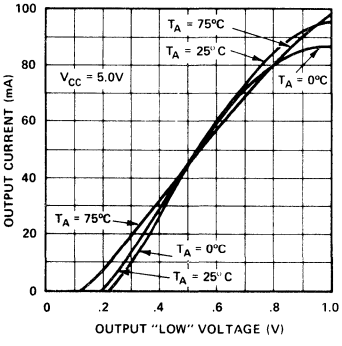
T_A = 0°C to +75°C, V_{CC} = 5V ±5%

8205

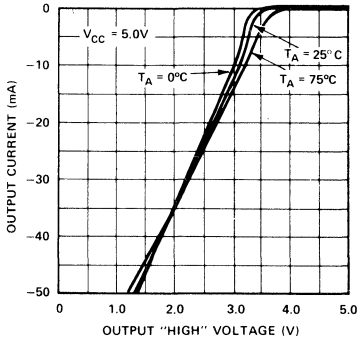
SYMBOL	PARAMETER	LIMIT MIN.	LIMIT MAX.	UNIT	TEST CONDITIONS
I_F	INPUT LOAD CURRENT		−0.25	mA	V_{CC} = 5.25V, V_F = 0.45V
I_R	INPUT LEAKAGE CURRENT		10	µA	V_{CC} = 5.25V, V_R = 5.25V
V_C	INPUT FORWARD CLAMP VOLTAGE		−1.0	V	V_{CC} = 4.75V, I_C = −5.0 mA
V_{OL}	OUTPUT "LOW" VOLTAGE		0.45	V	V_{CC} = 4.75V, I_{OL} = 10.0 mA
V_{OH}	OUTPUT HIGH VOLTAGE	2.4		V	V_{CC} = 4.75V, I_{OH} = −1.5 mA
V_{IL}	INPUT "LOW" VOLTAGE		0.85	V	V_{CC} = 5.0V
V_{IH}	INPUT "HIGH" VOLTAGE	2.0		V	V_{CC} = 5.0V
I_{SC}	OUTPUT HIGH SHORT CIRCUIT CURRENT	−40	−120	mA	V_{CC} = 5.0V, V_{OUT} = 0V
V_{OX}	OUTPUT "LOW" VOLTAGE @ HIGH CURRENT		0.8	V	V_{CC} = 5.0V, I_{OX} = 40 mA
I_{CC}	POWER SUPPLY CURRENT		70	mA	V_{CC} = 5.25V

TYPICAL CHARACTERISTICS

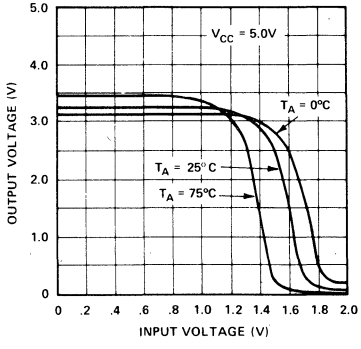
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



DATA TRANSFER FUNCTION



SWITCHING CHARACTERISTICS

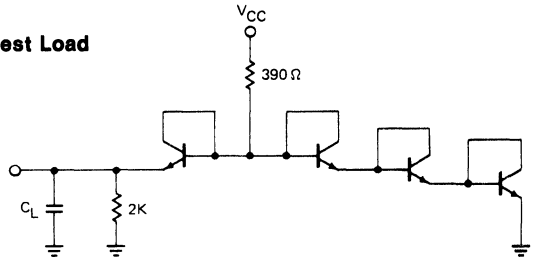
Conditions of Test:

Input pulse amplitudes: 2.5V

Input rise and fall times: 5 nsec between 1V and 2V

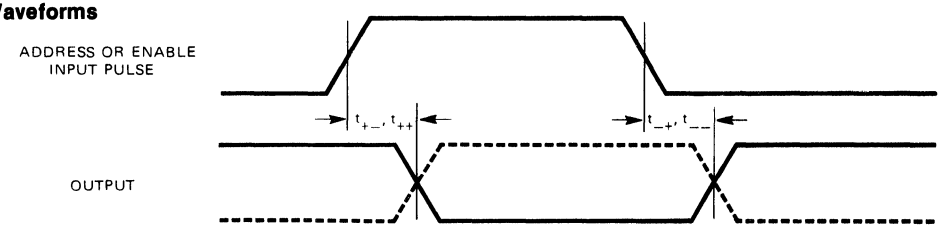
Measurements are made at 1.5V

Test Load



All Transistors 2N2369 or Equivalent. C_L = 30 pF

Test Waveforms



A.C. CHARACTRISTICS

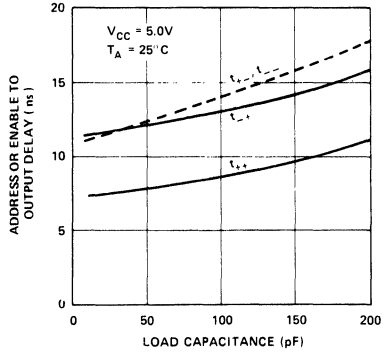
T_A = 0°C to +75°C, V_{CC} = 5V ±5% unless otherwise specified.

SYMBOL	PARAMETER		MAX. LIMIT	UNIT	TEST CONDITIONS
t_{++}	ADDRESS OR ENABLE TO OUTPUT DELAY		18	ns	
t_{-+}			18	ns	
t_{+-}			18	ns	
t_{--}			18	ns	
C_{IN} (1)	INPUT CAPACITANCE	P8205	4(typ.)	pF	f = 1 MHz, V_{CC} = 0V
		C8205	5(typ.)	pF	V_{BIAS} = 2.0V, T_A = 25°C

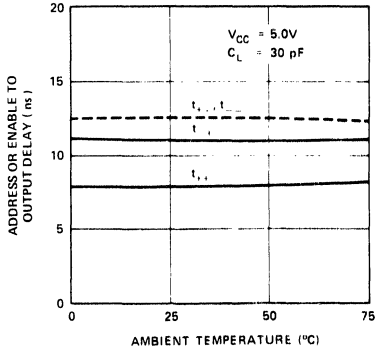
1. This parameter is periodically sampled and is not 100% tested.

TYPICAL CHARACTERISTICS

ADDRESS OR ENABLE TO OUTPUT DELAY VS. LOAD CAPACITANCE



ADDRESS OR ENABLE TO OUTPUT DELAY VS. AMBIENT TEMPERATURE



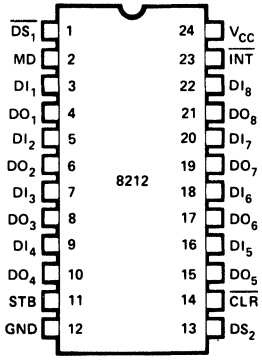
8212
8-BIT INPUT/OUTPUT PORT

- **Fully Parallel 8-Bit Data Register and Buffer**
- **Service Request Flip-Flop for Interrupt Generation**
- **Low Input Load Current — .25mA Max.**
- **Three State Outputs**
- **Outputs Sink 15mA**
- **3.65V Output High Voltage for Direct Interface to 8008, 8080A, or 8085A CPU**
- **Asynchronous Register Clear**
- **Replaces Buffers, Latches and Multiplexers in Microcomputer Systems**
- **Reduces System Package Count**

The 8212 input/output port consists of an 8-bit latch with 3-state output buffers along with control and device selection logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

The device is multimode in nature. It can be used to implement latches, gated buffers or multiplexers. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with this device.

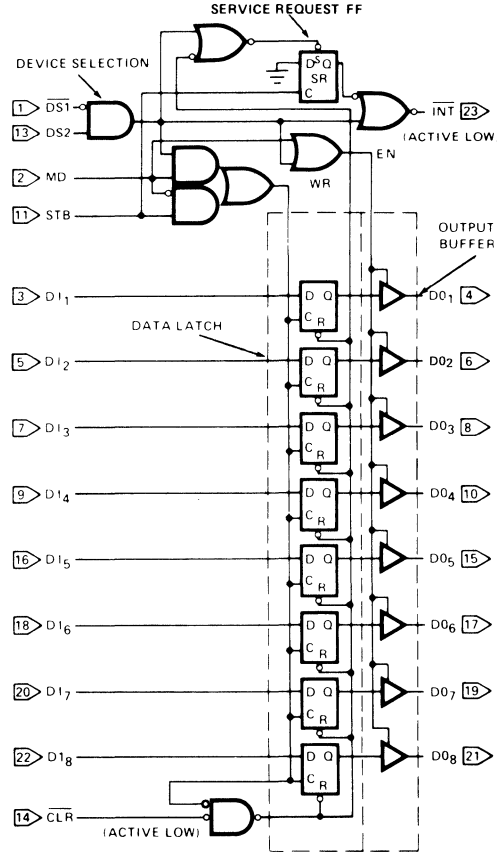
PIN CONFIGURATION



PIN NAMES

DI_1-DI_8	DATA IN
DO_1-DO_8	DATA OUT
$\overline{DS_1}$-DS_2	DEVICE SELECT
MD	MODE
STB	STROBE
$\overline{INT}$	INTERRUPT (ACTIVE LOW)
$\overline{CLR}$	CLEAR (ACTIVE LOW)

LOGIC DIAGRAM



FUNCTIONAL DESCRIPTION

Data Latch

The 8 flip-flops that make up the data latch are of a "D" type design. The output (Q) of the flip-flop will follow the data input (D) while the clock input (C) is high. Latching will occur when the clock (C) returns low.

The latched data is cleared by an asynchronous reset input ($\overline{CLR}$). (Note: Clock (C) Overrides Reset ($\overline{CLR}$).)

Output Buffer

The outputs of the data latch (Q) are connected to 3-state, non-inverting output buffers. These buffers have a common control line (EN); this control line either enables the buffer to transmit the data from the outputs of the data latch (Q) or disables the buffer, forcing the output into a high impedance state. (3-state)

The high-impedance state allows the designer to connect the 8212 directly onto the microprocessor bi-directional data bus.

Control Logic

The 8212 has control inputs $\overline{DS1}$, DS2, MD and STB. These inputs are used to control device selection, data latching, output buffer state and service request flip-flop.

$\overline{DS1}$, DS2 (Device Select)

These 2 inputs are used for device selection. When $\overline{DS1}$ is low and DS2 is high ($\overline{DS1} \cdot DS2$) the device is selected. In the selected state the output buffer is enabled and the service request flip-flop (SR) is asynchronously set.

MD (Mode)

This input is used to control the state of the output buffer and to determine the source of the clock input (C) to the data latch.

When MD is high (output mode) the output buffers are enabled and the source of clock (C) to the data latch is from the device selection logic ($\overline{DS1} \cdot DS2$).

When MD is low (input mode) the output buffer state is determined by the device selection logic ($\overline{DS1} \cdot DS2$) and the source of clock (C) to the data latch is the STB (Strobe) input.

STB (Strobe)

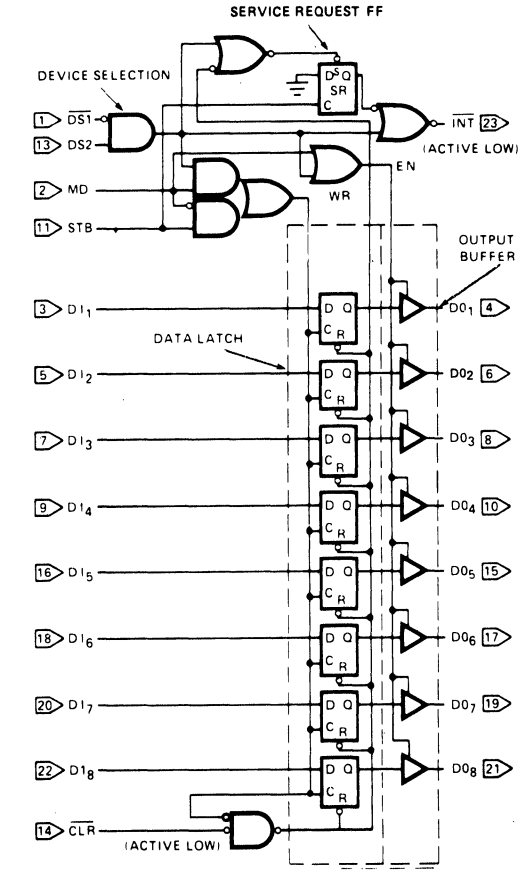
This input is used as the clock (C) to the data latch for the input mode MD = 0) and to synchronously reset the service request flip-flop (SR).

Note that the SR flip-flop is negative edge triggered.

Service Request Flip-Flop

The (SR) flip-flop is used to generate and control interrupts in microcomputer systems. It is asynchronously set by the $\overline{CLR}$ input (active low). When the (SR) flip-flop is set it is in the non-interrupting state.

The output of the (SR) flip-flop (Q) is connected to an inverting input of a "NOR" gate. The other input to the "NOR" gate is non-inverting and is connected to the device selection logic ($\overline{DS1} \cdot DS2$). The output of the "NOR" gate ($\overline{INT}$) is active low (interrupting state) for connection to active low input priority generating circuits.



STB	MD	($DS_1 \cdot DS_2$)	DATA OUT EQUALS
0	0	0	3-STATE
1	0	0	3-STATE
0	1	0	DATA LATCH
1	1	0	DATA LATCH
0	0	1	DATA LATCH
1	0	1	DATA IN
0	1	1	DATA IN
1	1	1	DATA IN

$\overline{CLR}$ – RESETS DATA LATCH
SETS SR FLIP-FLOP
(NO EFFECT ON OUTPUT BUFFER)

CLR	($\overline{DS_1} \cdot DS_2$)	STB	*SR	INT
0	0	0	1	1
0	1	0	1	0
1	1	↓	0	0
1	1	0	1	0
1	0	0	1	1
1	1	↓	1	0

*INTERNAL SR FLIP-FLOP

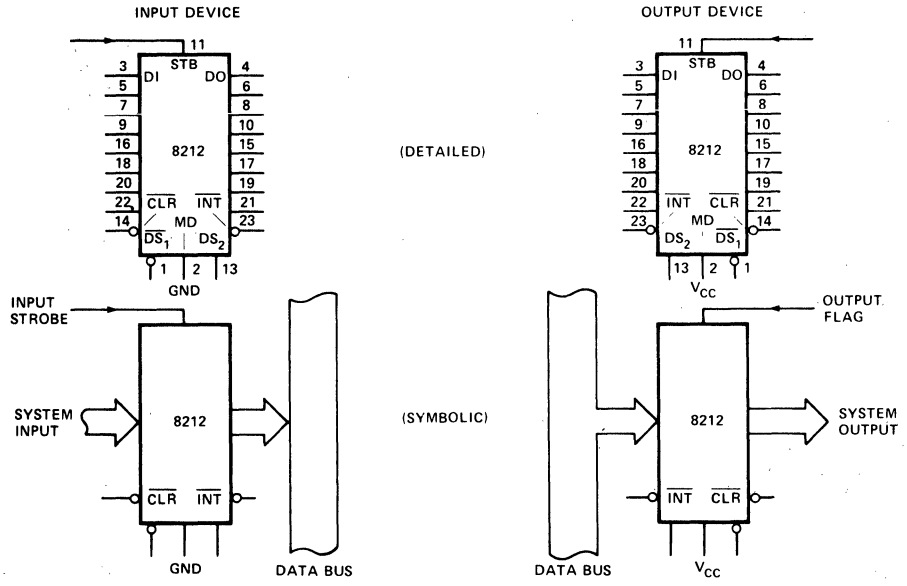
Applications of the 8212 — For Microcomputer Systems

I Basic Schematic Symbol
II Gated Buffer
III Bi-Directional Bus Driver
IV Interrupting Input Port
V Interrupt Instruction Port
VI Output Port
VII 8080A Status Latch
VIII 8085A Address Latch

1. Basic Schematic Symbols

Two examples of ways to draw the 8212 on system schematics — (1) the top being the detailed view showing pin numbers, and (2) the bottom being the symbolic view showing the system input or output as a system bus (bus containing 8 parallel lines). The output to the data bus is symbolic in referencing 8 parallel lines.

BASIC SCHEMATIC SYMBOLS



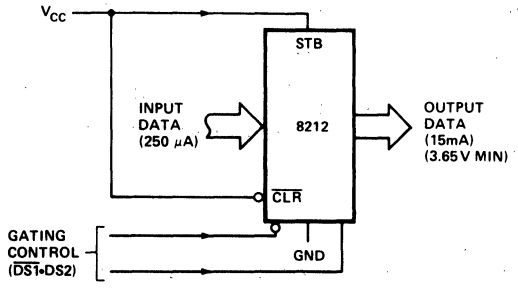
II. Gated Buffer (3-State)

The simplest use of the 8212 is that of a gated buffer. By tying the mode signal low and the strobe input high, the data latch is acting as a straight through gate. The output buffers are then enabled from the device selection logic $\overline{DS1}$ and DS2.

When the device selection logic is false, the outputs are 3-state.

When the device selection logic is true, the input data from the system is directly transferred to the output. The input data load is 250 micro amps. The output data can sink 15 milli amps. The minimum high output is 3.65 volts.

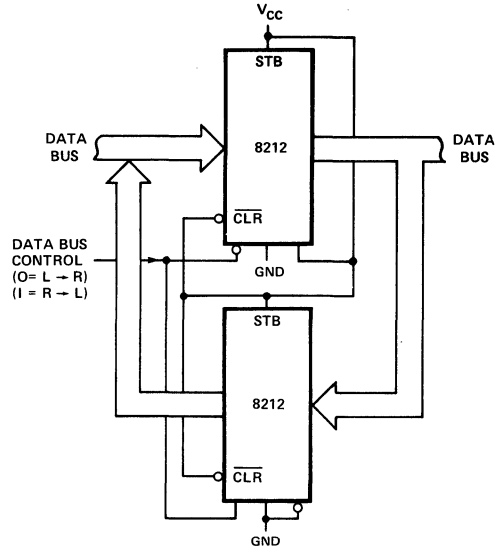
GATED BUFFER



III. Bi-Directional Bus Driver

A pair of 8212's wired (back-to-back) can be used as a symmetrical drive, bi-directional bus driver. The devices are controlled by the data bus input control which is connected to $\overline{DS1}$ on the first 8212 and to DS2 on the second. One device is active, and acting as a straight through buffer the other is in 3-state mode. This is a very useful circuit in small system design.

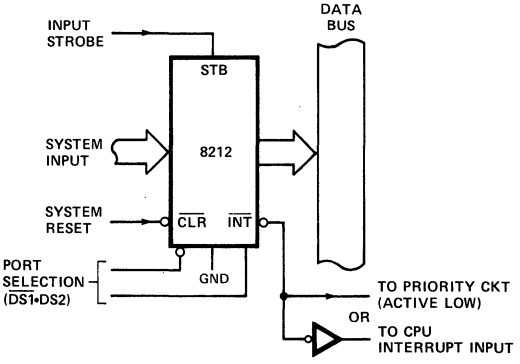
BI-DIRECTIONAL BUS DRIVER



IV. Interrupting Input Port

This use of an 8212 is that of a system input port that accepts a strobe from the system input source, which in turn clears the service request flip-flop and interrupts the processor. The processor then goes through a service routine, identifies the port, and causes the device selection logic to go true — enabling the system input data onto the data bus.

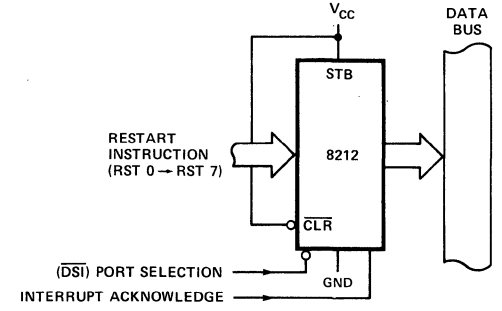
INTERRUPTING INPUT PORT



V. Interrupt Instruction Port

The 8212 can be used to gate the interrupt instruction, normally RESTART instructions, onto the data bus. The device is enabled from the interrupt acknowledge signal from the microprocessor and from a port selection signal. This signal is normally tied to ground. ($\overline{DS1}$ could be used to multiplex a variety of interrupt instruction ports onto a common bus).

INTERRUPT INSTRUCTION PORT

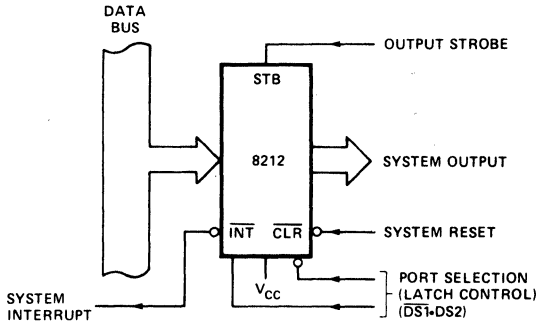


AFN-00731A-04

VI. Output Port (With Hand-Shaking)

The 8212 can be used to transmit data from the data bus to a system output. The output strobe could be a hand-shaking signal such as "reception of data" from the device that the system is outputting to. It in turn, can interrupt the system signifying the reception of data. The selection of the port comes from the device selection logic.($\overline{DS1}$ · DS2)

OUTPUT PORT (WITH HAND-SHAKING)

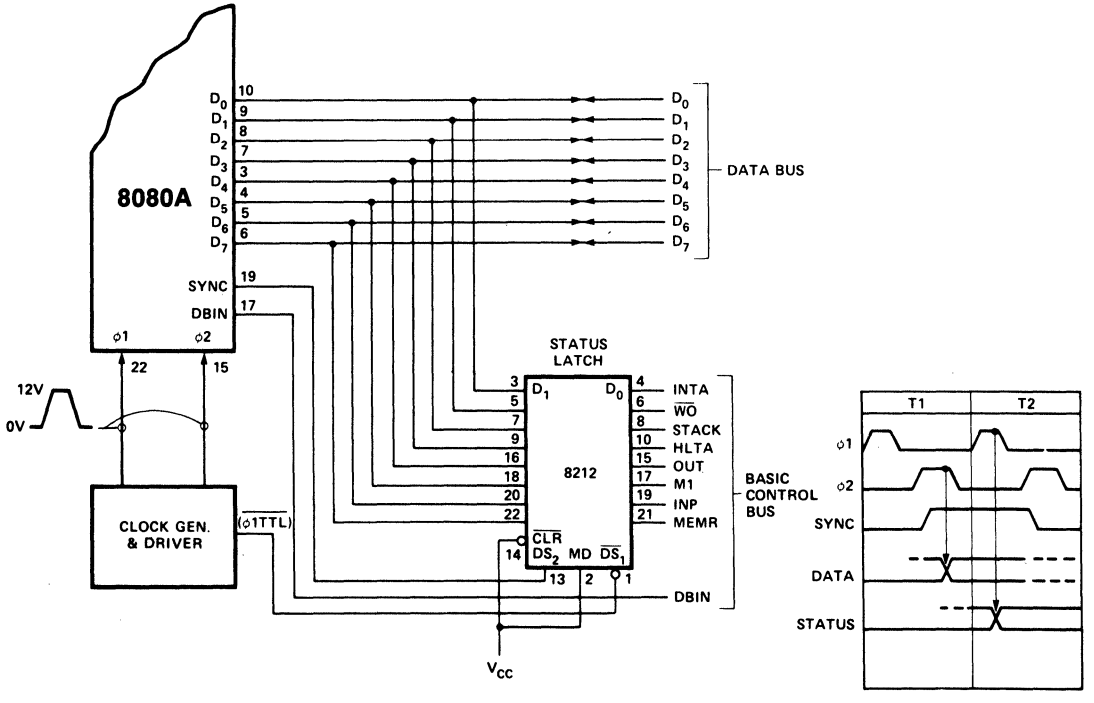


VII. 8080A Status Latch

Here the 8212 is used as the status latch for an 8080A microcomputer system. The input to the 8212 latch is directly from the 8080A data bus. Timing shows that when the SYNC signal is true, which is connected to the DS2 input and the phase 1 signal is true, which is a TTL level coming from the clock generator; then, the status data will be latched into the 8212.

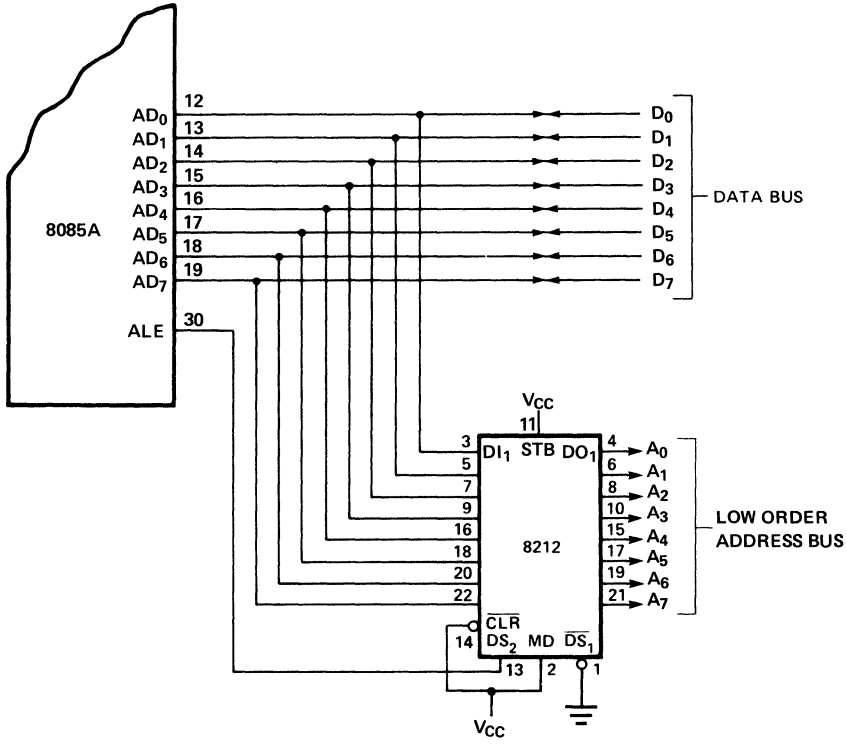
Note: The mode signal is tied high so that the output on the latch is active and enabled all the time.

It is shown that the two areas of concern are the bi-directional data bus of the microprocessor and the control bus.



VIII. 8085A Low-Order Address Latch

The 8085A microprocessor uses a multiplexed address/data bus that contains the low order 8-bits of address information during the first part of a machine cycle. The same bus contains data at a later time in the cycle. An address latch enable (ALE) signal is provided by the 8085A to be used by the 8212 to latch the address so that it may be available through the whole machine cycle. Note: In this configuration, the MODE input is tied high, keeping the 8212's output buffers turned on at all times.



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias Plastic ..... 0°C to +70°C
Storage Temperature ..... -65°C to +160°C
All Output or Supply Voltages ..... -0.5 to +7 Volts
All Input Voltages ..... -1.0 to 5.5 Volts
Output Currents 100mA

*COMMENT

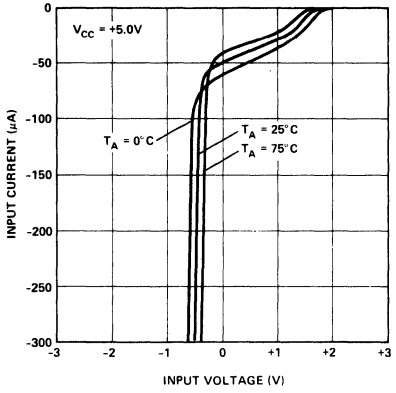
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure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

D.C. CHARACTERISTICS $T_A = 0°C$ to $+75°C$, $V_{CC} = +5V \pm 5\%$

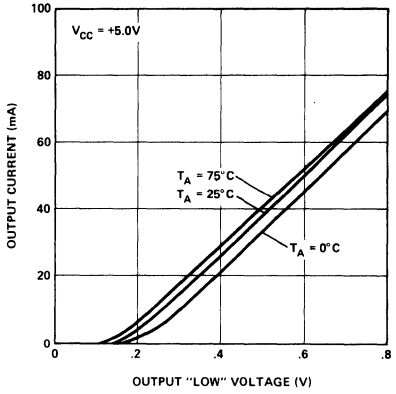
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_F	Input Load Current, ACK, DS_2, CR, DI_1-DI_8 Inputs			-.25	mA	$V_F = .45V$
I_F	Input Load Current MD Input			-.75	mA	$V_F = .45V$
I_F	Input Load Current DS_1 Input			-1.0	mA	$V_F = .45V$
I_R	Input Leakage Current, ACK, DS, CR, DI_1-DI_8 Inputs			10	μA	$V_R \leq V_{CC}$
I_R	Input Leakage Current MO Input			30	μA	$V_R \leq V_{CC}$
I_R	Input Leakage Current DS_1 Input			40	μA	$V_R \leq V_{CC}$
V_C	Input Forward Voltage Clamp			-1	V	$I_C = -5mA$
V_{IL}	Input "Low" Voltage			.85	V	
V_{IH}	Input "High" Voltage	2.0			V	
V_{OL}	Output "Low" Voltage			.45	V	$I_{OL} = 15mA$
V_{OH}	Output "High" Voltage	3.65	4.0		V	$I_{OH} = -1mA$
I_{SC}	Short Circuit Output Current	-15		-75	mA	$V_O = 0V$, $V_{CC} = 5V$
$\|I_O\|$	Output Leakage Current High Impedance State			20	μA	$V_O = .45V/5.25V$
I_{CC}	Power Supply Current		90	130	mA	

TYPICAL CHARACTERISTICS

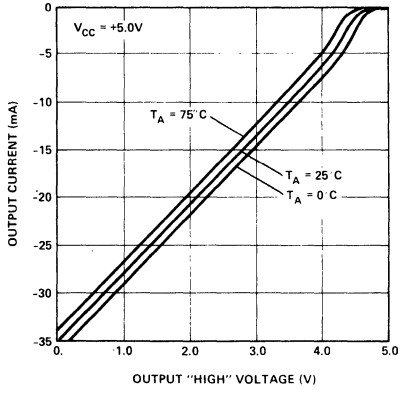
INPUT CURRENT VS. INPUT VOLTAGE



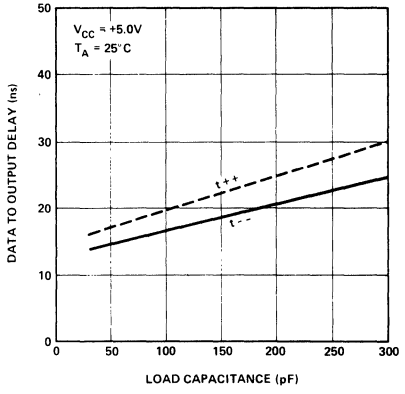
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



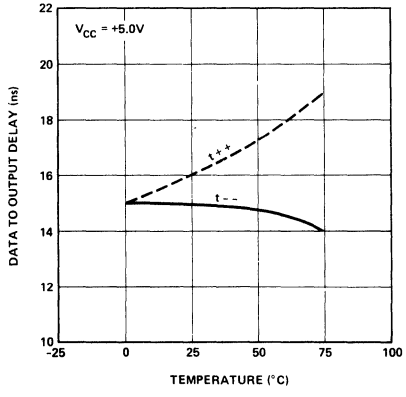
OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



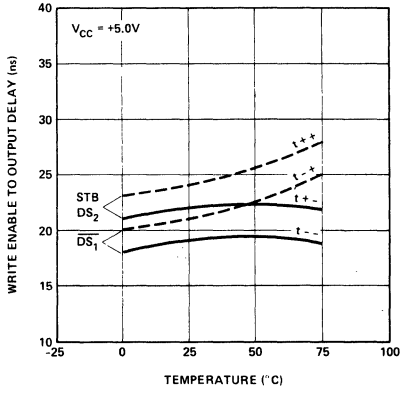
DATA TO OUTPUT DELAY VS. LOAD CAPACITANCE



DATA TO OUTPUT DELAY VS. TEMPERATURE



WRITE ENABLE TO OUTPUT DELAY VS. TEMPERATURE



A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
t_{PW}	Pulse Width	30			ns	
t_{PD}	Data to Output Delay			30	ns	Note 1
t_{WE}	Write Enable to Output Delay			40	ns	Note 1
t_{SET}	Data Set Up Time	15			ns	
t_H	Data Hold Time	20			ns	
t_R	Reset to Output Delay			40	ns	Note 1
t_S	Set to Output Delay			30	ns	Note 1
t_E	Output Enable/Disable Time			45	ns	Note 1
t_C	Clear to Output Delay			55	ns	Note 1

CAPACITANCE* $F = 1MHz$, $V_{BIAS} = 2.5V$, $V_{CC} = +5V$, $T_A = 25°C$

Symbol	Test	Limits	
		Typ.	Max.
C_{IN}	DS_1 MD Input Capacitance	9pF	12pF
C_{IN}	DS_2, CK, ACK, DI_1-DI_8 Input Capacitance	5pF	9pF
C_{OUT}	DO_1-DO_8 Output Capacitance	8pF	12pF

*This parameter is sampled and not 100% tested.

SWITCHING CHARACTERISTICS

Conditions of Test

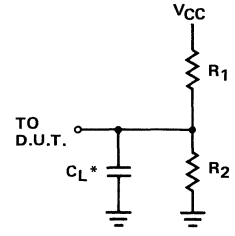
Input Pulse Amplitude = 2.5V
Input Rise and Fall Times 5ns
Between 1V and 2V Measurements made at 1.5V
with 15mA and 30pF Test Load

Note 1:

Test	C_L*	R_1	R_2
t_{PD}, t_{WE}, t_R, t_S, t_C	30pF	300Ω	600Ω
t_E, ENABLE↑	30pF	10KΩ	1KΩ
t_E, ENABLE↓	30pF	300Ω	600Ω
t_E, DISABLE↑	5pF	300Ω	600Ω
t_E, DISABLE↓	5pF	10KΩ	1KΩ

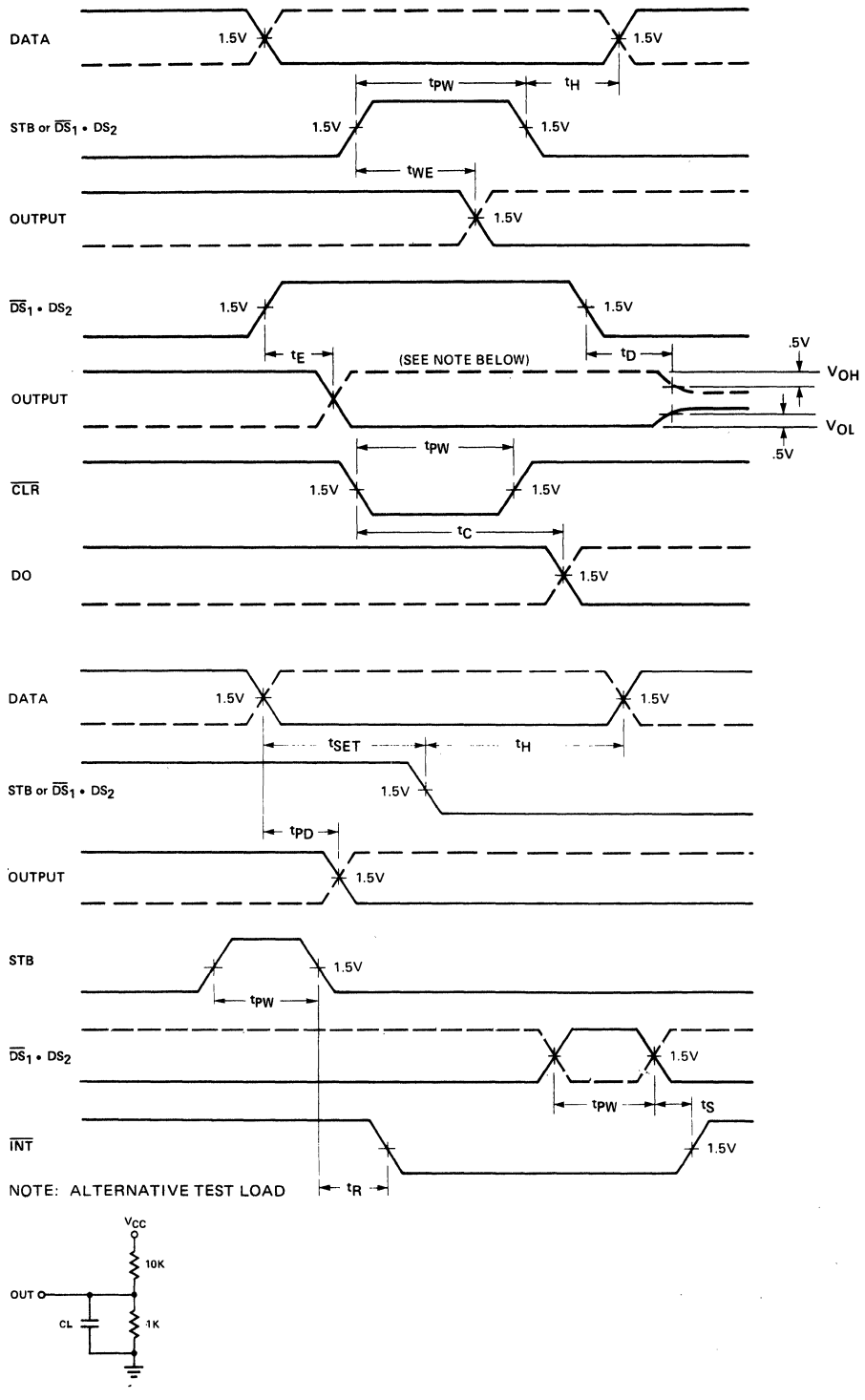
*Includes probe and jig capacitance.

Test Load
15mA & 30pF



*INCLUDING JIG & PROBE CAPACITANCE

TIMING DIAGRAM



8214/3214*
PRIORITY INTERRUPT CONTROL UNIT

- **8 Priority Levels**
- **Fully Expandable**
- **Current Status Register**
- **High Performance (50 ns)**
- **Priority Comparator**
- **24-Pin Dual In-Line Package**

The Intel® 8214 is an 8-level priority interrupt control unit (PICU) designed to simplify interrupt-driven microcomputer systems.

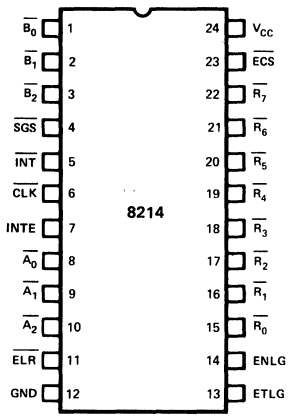
The PICU can accept 8 requesting levels; determine the highest priority, compare this priority to a software controlled current status register and issue an interrupt to the system along with vector information to identify the service routine.

The 8214 is fully expandable by the use of open collector interrupt output vector information. Control signals are also provided to simplify this function.

The PICU is designed to support a wide variety of vectored interrupt structures and reduce package count in interrupt-driven microcomputer systems.

*Note: The specifications for the 3214 are identical with those for the 8214.

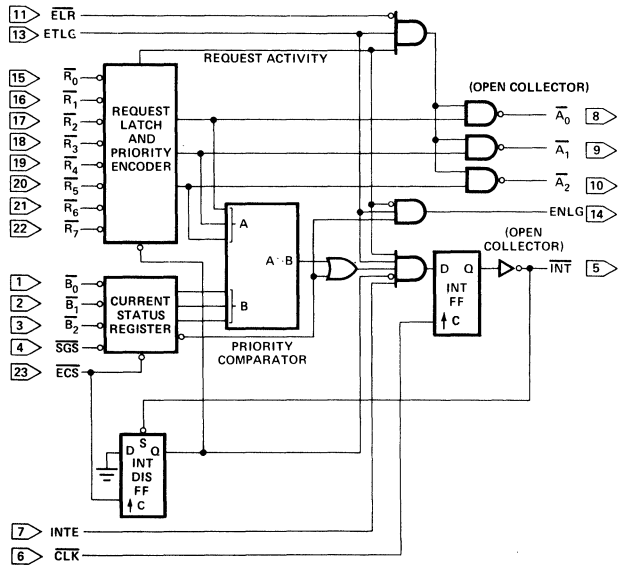
PIN CONFIGURATION



PIN NAMES

INPUTS	
$\overline{R_0}$-$\overline{R_7}$	REQUEST LEVELS (R_7 HIGHEST PRIORITY)
$\overline{B_0}$-$\overline{B_2}$	CURRENT STATUS
$\overline{SGS}$	STATUS GROUP SELECT
$\overline{ECS}$	ENABLE CURRENT STATUS
INTE	INTERRUPT ENABLE
$\overline{CLK}$	CLOCK (INT F-F)
$\overline{ELR}$	ENABLE LEVEL READ
ETLG	ENABLE THIS LEVEL GROUP
OUTPUTS:	
$\overline{A_0}$-$\overline{A_2}$	REQUEST LEVELS (OPEN COLLECTOR)
$\overline{INT}$	INTERRUPT (ACT. LOW) (OPEN COLLECTOR)
ENLG	ENABLE NEXT LEVEL GROUP

LOGIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . 0°C to 75°C
Storage Temperature . . . . . −65°C to +160°C
All Output and Supply Voltages . . . . . -0.5V to +7V
All Input Voltages . . . . . -1.0V to +5.5V
Output Currents . . . . . 100 mA

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±5%.

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ.[1]	Max.		
V_C	Input Clamp Voltage (all inputs)			-1.0	V	I_C=-5mA
I_F	Input Forward Current: ETLG input		-.15	-0.5	mA	V_F=0.45V
	all other inputs		-.08	-0.25	mA	
I_R	Input Reverse Current: ETLG input			80	μA	V_R=5.25V
	all other inputs			40	μA	
V_{IL}	Input LOW Voltage: all inputs			0.8	V	V_{CC}=5.0V
V_{IH}	Input HIGH Voltage: all inputs	2.0			V	V_{CC}=5.0V
I_{CC}	Power Supply Current		90	130	mA	See Note 2.
V_{OL}	Output LOW Voltage: all outputs		.3	.45	V	I_{OL}=15mA
V_{OH}	Output HIGH Voltage: ENLG output	2.4	3.0		V	I_{OH}=-1mA
I_{OS}	Short Circuit Output Current: ENLG output	-20	-35	-55	mA	V_{OS}=0V, V_{CC}=5.0V
I_{CEX}	Output Leakage Current: $\overline{INT}$ and $\overline{A_0}$-$\overline{A_2}$			100	μA	V_{CEX}=5.25V

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. B_0-B_2, $\overline{SGS}$, CLK, $\overline{R_0}$-$\overline{R_4}$ grounded, all other inputs and all outputs open.

A.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Limits			Unit
		Min.	Typ.[1]	Max.	
t_{CY}	$\overline{CLK}$ Cycle Time	80	50		ns
t_{PW}	$\overline{CLK}$, $\overline{ECS}$, $\overline{INT}$ Pulse Width	25	15		ns
t_{ISS}	INTE Setup Time to $\overline{CLK}$	16	12		ns
t_{ISH}	INTE Hold Time after $\overline{CLK}$	20	10		ns
t_{ETCS}[2]	ETLG Setup Time to $\overline{CLK}$	25	12		ns
t_{ETCH}[2]	ETLG Hold Time After $\overline{CLK}$	20	10		ns
t_{ECCS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	80	25		ns
t_{ECCH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{ECRS}[3]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	110	70		ns
t_{ECRH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			
t_{ECSS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	75	70		ns
t_{ECSH}[2]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{DCS}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{CLK}$	70	50		ns
t_{DCH}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{CLK}$	0			ns
t_{RCS}[3]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{CLK}$	90	55		ns
t_{RCH}[3]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{CLK}$	0			ns
t_{ICS}	$\overline{INT}$ Setup Time to $\overline{CLK}$	55	35		ns
t_{CI}	$\overline{CLK}$ to $\overline{INT}$ Propagation Delay		15	25	ns
t_{RIS}[4]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{INT}$	10	0		ns
t_{RIH}[4]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{INT}$	35	20		ns
t_{RA}	$\overline{R_0}$-$\overline{R_7}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		80	100	ns
t_{ELA}	$\overline{ELR}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		40	55	ns
t_{ECA}	$\overline{ECS}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		100	120	ns
t_{ETA}	ETLG to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		35	70	ns
t_{DECS}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{ECS}$	15	10		ns
t_{DECH}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{ECS}$	15	10		ns
t_{REN}	$\overline{R_0}$-$\overline{R_7}$ to ENLG Propagation Delay		45	70	ns
t_{ETEN}	ETLG to ENLG Propagation Delay		20	25	ns
t_{ECRN}	$\overline{ECS}$ to ENLG Propagation Delay		85	90	ns
t_{ECSN}	$\overline{ECS}$ to ENLG Propagation Delay		35	55	ns

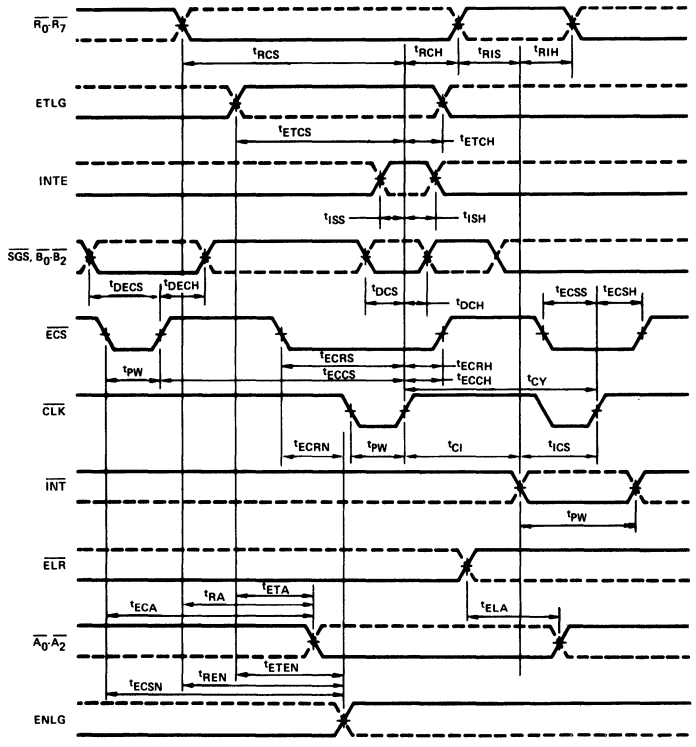
CAPACITANCE[5]

Symbol	Parameter	Limits			Unit
		Min.	Typ.[1]	Max	
C_{IN}	Input Capacitance		5	10	pF
C_{OUT}	Output Capacitance		7	12	pF

Test Conditions: $V_{BIAS} = 2.5V$, $V_{CC} = 5V$, $T_A = 25°C$, f = 1 MHz

NOTE 5. This parameter is periodically sampled and not 100% tested.

WAVEFORMS



NOTES:

(1) Typical values are for $T_A = 25°C$, $V_{CC} = 5.0V$.

(2) Required for proper operation if ISE is enabled during next clock pulse.

(3) These times are not required for proper operation but for desired change in interrupt flip-flop.

(4) Required for new request or status to be properly loaded.

Test Conditions

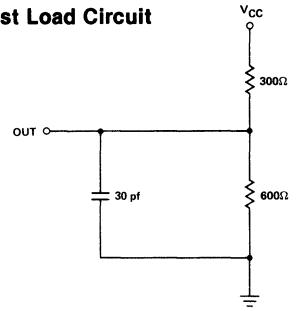
Input pulse amplitude: 2.5 volts.

Input rise and fall times: 5 ns between 1 and 2 volts.

Output loading of 15 mA and 30 pf.

Speed measurements taken at the 1.5V levels.

Test Load Circuit



AFN-00210A-04

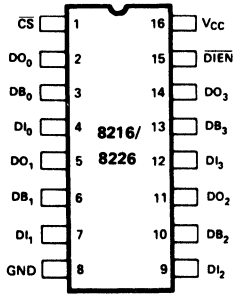
8216/8226
4-BIT PARALLEL BIDIRECTIONAL BUS DRIVER

- **Data Bus Buffer Driver for 8080 CPU**
- **Low Input Load Current — 0.25 mA Maximum**
- **High Output Drive Capability for Driving System Bus**
- **3.65V Output High Voltage for Direct Interface to 8080 CPU**
- **3-State Outputs**
- **Reduces System Package Count**

The 8216/8226 is a 4-bit bidirectional bus driver/receiver. All inputs are low power TTL compatible. For driving MOS, the DO outputs provide a high 3.65V V_{OH}, and for high capacitance terminated bus structures, the DB outputs provide a high 50 mA I_{OL} capability. A non-inverting (8216) and an inverting (8226) are available to meet a wide variety of applications for buffering in microcomputer systems.

*Note: The specifications for the 3216/3226 are identical with those for the 8216/8226.

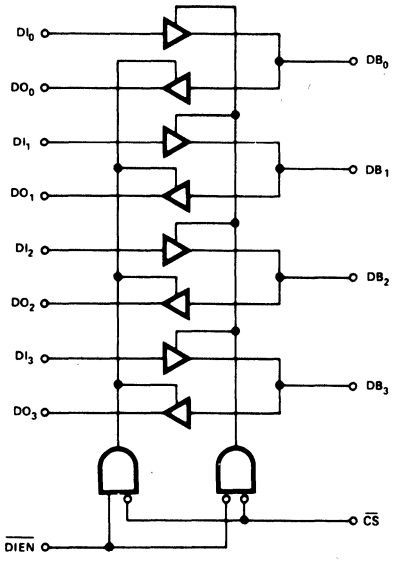
PIN CONFIGURATION



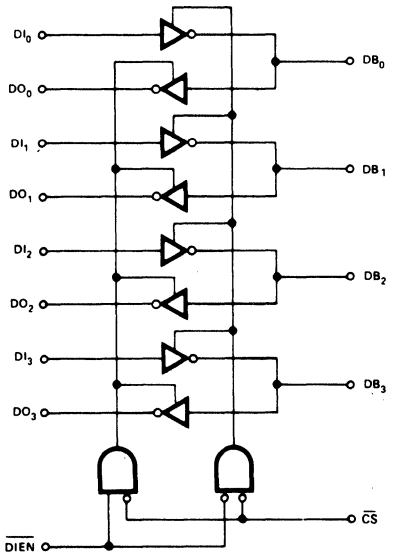
PIN NAMES

DB_0-DB_3	DATA BUS BI-DIRECTIONAL
DI_0-DI_3	DATA INPUT
DO_0-DO_3	DATA OUTPUT
$\overline{DIEN}$	DATA IN ENABLE DIRECTION CONTROL
$\overline{CS}$	CHIP SELECT

LOGIC DIAGRAM
8216



LOGIC DIAGRAM
8226



FUNCTIONAL DESCRIPTION

Microprocessors like the 8080 are MOS devices and are generally capable of driving a single TTL load. The same is true for MOS memory devices. While this type of drive is sufficient in small systems with few components, quite often it is necessary to buffer the microprocessor and memories when adding components or expanding to a multi-board system.

The 8216/8226 is a four bit bi-directional bus driver specifically designed to buffer microcomputer system components.

Bidirectional Driver

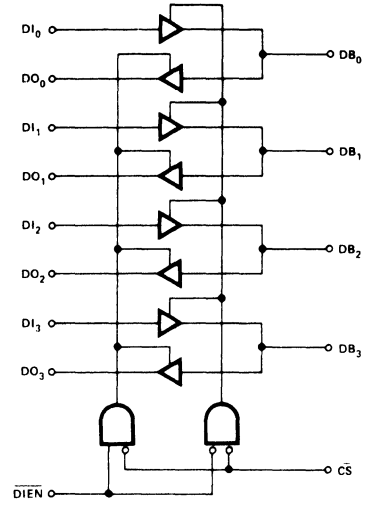
Each buffered line of the four bit driver consists of two separate buffers that are tri-state in nature to achieve direct bus interface and bi-directional capability. On one side of the driver the output of one buffer and the input of another are tied together (DB), this side is used to interface to the system side components such as memories, I/O, etc., because its interface is direct TTL compatible and it has high drive (50mA). On the other side of the driver the inputs and outputs are separated to provide maximum flexibility. Of course, they can be tied together so that the driver can be used to buffer a true bi-directional bus such as the 8080 Data Bus. The DO outputs on this side of the driver have a special high voltage output drive capability (3.65V) so that direct interface to the 8080 and 8008 CPUs is achieved with an adequate amount of noise immunity (350mV worst case).

Control Gating $\overline{DIEN}$, $\overline{CS}$

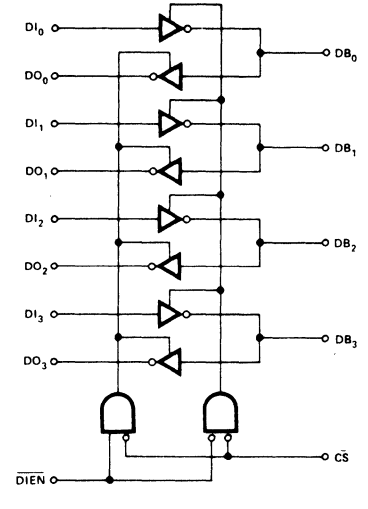
The $\overline{CS}$ input is actually a device select. When it is "high" the output drivers are all forced to their high-impedance state. When it is at "zero" the device is selected (enabled) and the direction of the data flow is determined by the $\overline{DIEN}$ input.

The $\overline{DIEN}$ input controls the direction of data flow (see Figure 1) for complete truth table. This direction control is accomplished by forcing one of the pair of buffers into its high impedance state and allowing the other to transmit its data. A simple two gate circuit is used for this function.

The 8216/8226 is a device that will reduce component count in microcomputer systems and at the same time enhance noise immunity to assure reliable, high performance operation.



(a) 8216

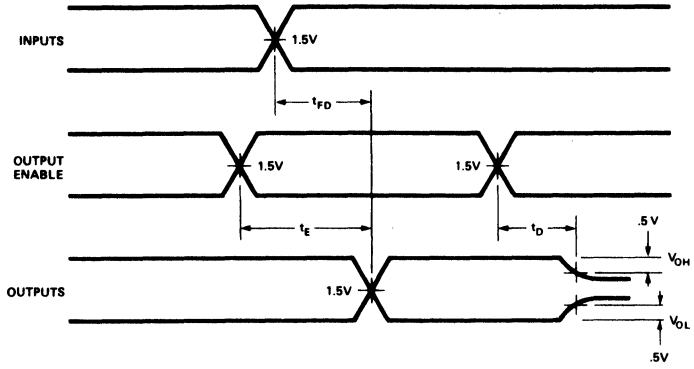


(b) 8226

$\overline{DIEN}$	$\overline{CS}$	
0	0	DI → DB
1	0	DB → DO
0	1	HIGH IMPEDANCE
1	1	

Figure 1. 8216/8226 Logic Diagrams

WAVEFORMS



A.C. CHARACTERISTICS

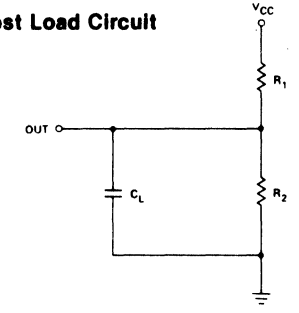
T_A = 0°C to +70°C. V_{CC}=+5V ±5%

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit	Conditions
T_{PD1}	Input to Output Delay DO Outputs		15	25	ns	C_L=30pF, R_1=300Ω R_2=600Ω
T_{PD2}	Input to Output Delay DB Outputs 8216		19	30	ns	C_L=300pF, R_1=90Ω
	8226		16	25	ns	R_2 = 180Ω
T_E	Output Enable Time 8216		42	65	ns	(Note 2)
	8226		36	54	ns	(Note 3)
T_D	Output Disable Time		16	35	ns	(Note 4)

Test Conditions:

Input pulse amplitude of 2.5V.
Input rise and fall times of 5 ns between 1 and 2 volts.
Output loading is 5 mA and 10 pF.
Speed measurements are made at 1.5 volt levels.

Test Load Circuit



CAPACITANCE[5]

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit
C_{IN}	Input Capacitance		4	8	pF
C_{OUT1}	Output Capacitance		6	10	pF
C_{OUT2}	Output Capacitance		13	18	pF

Test Conditions V_{BIAS} = 2.5V, V_{CC} = 5.0V, T_A = 25°C, f = 1 MHz.

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 180/1KΩ; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
3. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 600/1K; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
4. DO Outputs, C_L = 5pF, R_1 = 300/10 KΩ, R_2 = 600/1 KΩ; DB Outputs, C_L = 5pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
5. This parameter is periodically sampled and not 100% tested.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . -65°C to +150°C
All Output and Supply Voltages . . . . . -0.5V to +7V
All Input Voltages . . . . . -1.0V to +5.5V
Output Currents . . . . . 125 mA

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

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = +5V ±5%

Symbol	Parameter		Limits Min.	Typ.	Max.	Unit	Conditions
I_{F1}	Input Load Current $\overline{DIEN}$, $\overline{CS}$			-0.15	-.5	mA	V_F = 0.45
I_{F2}	Input Load Current All Other Inputs			-0.08	-.25	mA	V_F = 0.45
I_{R1}	Input Leakage Current $\overline{DIEN}$, $\overline{CS}$				80	μA	V_R = 5.25V
I_{R2}	Input Leakage Current DI Inputs				40	μA	V_R = 5.25V
V_C	Input Forward Voltage Clamp				-1	V	I_C = -5mA
V_{IL}	Input "Low" Voltage				.95	V	
V_{IH}	Input "High" Voltage		2.0			V	
$\|I_O\|$	Output Leakage Current (3-State)	DO DB			20 100	μA	V_O = 0.45V/5.25V
I_{CC}	Power Supply Current	8216		95	130	mA	
		8226		85	120	mA	
V_{OL1}	Output "Low" Voltage			0.3	.45	V	DO Outputs I_{OL}=15mA DB Outputs I_{OL}=25mA
V_{OL2}	Output "Low" Voltage	8216		0.5	.6	V	DB Outputs I_{OL}=55mA
		8226		0.5	.6	V	DB Outputs I_{OL}=50mA
V_{OH1}	Output "High" Voltage		3.65	4.0		V	DO Outputs I_{OH} = -1mA
V_{OH2}	Output "High" Voltage		2.4	3.0		V	DB Outputs I_{OH} = -10mA
I_{OS}	Output Short Circuit Current		-15 -30	-35 -75	-65 -120	mA mA	DO Outputs $V_O \cong 0V$, DB Outputs V_{CC}=5.0V

NOTE: Typical values are for T_A = 25°C, V_{CC} = 5.0V.

APPLICATIONS OF THE 8216/8226

8080 Data Bus Buffer

The 8080 CPU Data Bus is capable of driving a single TTL load and is more than adequate for small, single board systems. When expanding such a system to more than one board to increase I/O or Memory size, it is necessary to provide a buffer. The 8216/8226 is a device that is exactly fitted to this application.

Shown in Figure 2 are a pair of 8216/8226 connected directly to the 8080 Data Bus and associated control signals. The buffer is bi-directional in nature and serves to isolate the CPU data bus.

On the system side, the DB lines interface with standard semiconductor I/O and Memory components and are completely TTL compatible. The DB lines also provide a high drive capability (50mA) so that an extremely large system can be dirven along with possible bus termination networks.

On the 8080 side the DI and DO lines are tied together and are directly connected to the 8080 Data Bus for bi-directional operation. The DO outputs of the 8216/8226 have a high voltage output capability of 3.65 volts which allows direct connection to the 8080 whose minimum input voltage is 3.3 volts. It also gives a very adequate noise margin of 350mV (worst case).

The $\overline{DIEN}$ inputs to 8216/8226 is connected directly to the 8080. $\overline{DIEN}$ is tied to DBIN so that proper bus flow is maintained, and $\overline{CS}$ is tied to $\overline{BUSEN}$ so that the system side Data Bus will be 3-stated when a Hold request has been acknowledged during a DMA activity.

Memory and I/O Interface to a Bidirectional Bus

In large microcomputer systems it is often necessary to provide Memory and I/O with their own buffers and at the same time maintain a direct, common interface to a bi-directional Data Bus. The 8216/8226 has separated data in and data out lines on one side and a common bi-directional set on the other to accomodate such a function.

Shown in Figure 3 is an example of how the 8216/8226 is used in this type of application.

The interface to Memory is simple and direct. The memories used are typically Intel® 8102, 8102A, 8101 or 8107B-4 and have separate data inputs and outputs. The DI and DO lines of the 8216/8226 tie to them directly and under control of the $\overline{MEMR}$ signal, which is connected to the $\overline{DIEN}$ input, an interface to the bi-directional Data Bus is maintained.

The interface to I/O is similar to Memory. The I/O devices used are typically Intel® 8255s, and can be used for both input and output ports. The $\overline{I/O\ R}$ signal is connected directly to the $\overline{DIEN}$ input so that proper data flow from the I/O device to the Data Bus is maintained.

The 8216/8226 can be used in a wide variety of other buffering functions in microcomputer systems such as Address Bus Drivers, Drivers to peripheral devices such as printers, and as Drivers for long length cables to other peripherals or systems.

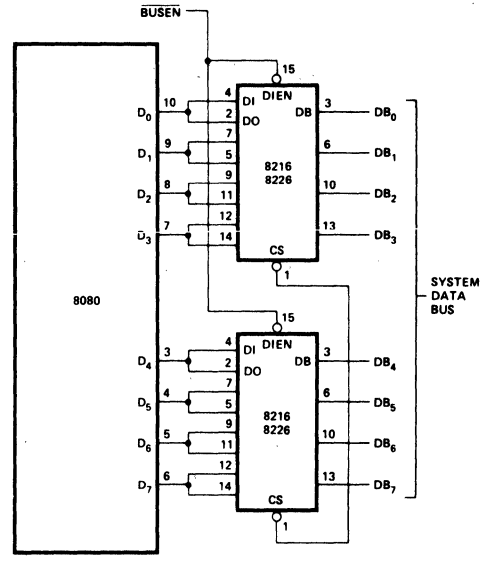


Figure 2. 8080 Data Bus Buffer

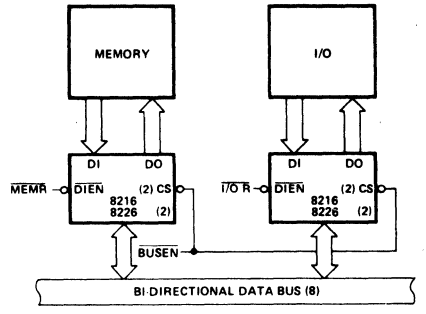


Figure 3. Memory and I/O Interface to a Bidirectional Bus

8282/8283
OCTAL LATCH

- **Address Latch for iAPX 86,88, MCS-80™, MCS-85™, MCS-48™ Families**
- **High Output Drive Capability for Driving System Data Bus**
- **Fully Parallel 8-Bit Data Register and Buffer**
- **Transparent during Active Strobe**
- **3-State Outputs**
- **20-Pin Package with 0.3" Center**
- **No Output Low Noise when Entering or Leaving High Impedance State**

The 8282 and 8283 are 8-bit bipolar latches with 3-state output buffers. They can be used to implement latches, buffers, or multiplexers. The 8283 inverts the input data at its outputs while the 8282 does not. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with these devices.

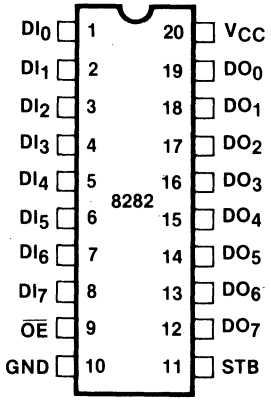


Figure 1. 8282 Pin Configuration

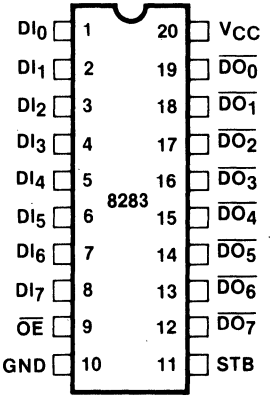


Figure 2. 8283 Pin Configuration

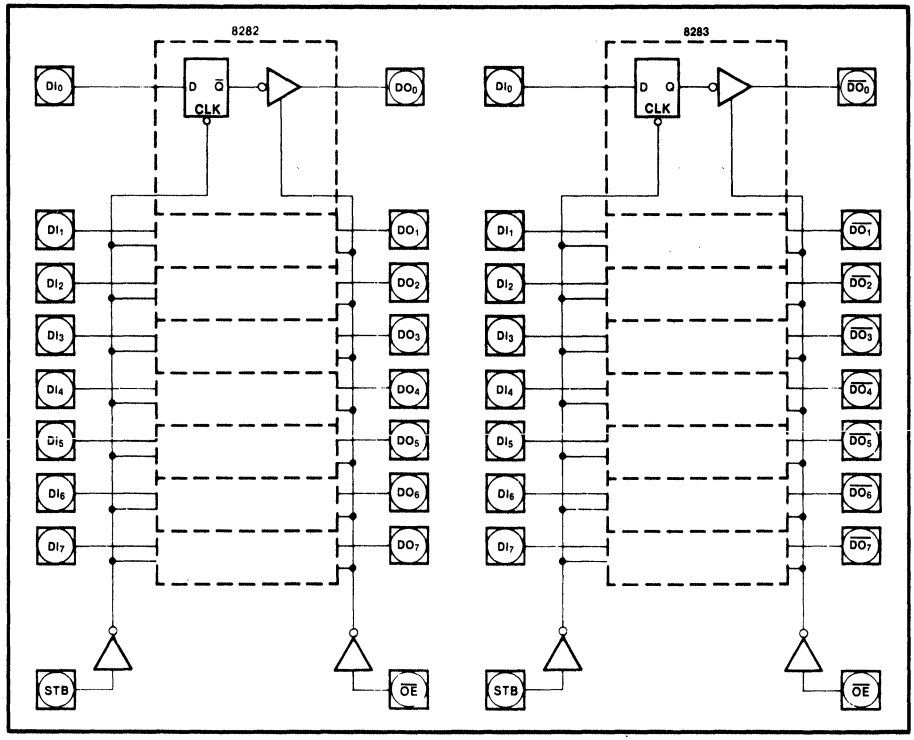


Figure 3. Logic Diagrams

PIN DEFINITIONS

Pin	Description
STB	STROBE (Input). STB is an input control pulse used to strobe data at the data input pins (A_0-A_7) into the data latches. This signal is active HIGH to admit input data. The data is latched at the HIGH to LOW transition of STB.
$\overline{OE}$	OUTPUT ENABLE (Input). $\overline{OE}$ is an input control signal which when active LOW enables the contents of the data latches onto the data output pin (B_0-B_7). OE being inactive HIGH forces the output buffers to their high impedance state.
DI_0-DI_7	DATA INPUT PINS (Input). Data presented at these pins satisfying setup time requirements when STB is strobed and latched into the data input latches.
DO_0-DO_7 (8282) $\overline{DO_0}$-$\overline{DO_7}$ (8283)	DATA OUTPUT PINS (Output). When $\overline{OE}$ is true, the data in the data latches is presented as inverted (8283) or non-inverted (8282) data onto the data output pins.

OPERATIONAL DESCRIPTION

The 8282 and 8283 octal latches are 8-bit latches with 3-state output buffers. Data having satisfied the setup time requirements is latched into the data latches by strobing the STB line HIGH to LOW. Holding the STB line in its active HIGH state makes the latches appear transparent. Data is presented to the data output pins by activating the $\overline{OE}$ input line. When $\overline{OE}$ is inactive HIGH the output buffers are in their high impedance state. Enabling or disabling the output buffers will not cause negative-going transients to appear on the data output bus.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias.....0°C to 70°C
Storage Temperature.....−65°C to +150°C
All Output and Supply Voltages.....−0.5V to +7V
All Input Voltages.....−1.0V to +5.5V
Power Dissipation.....1 Watt

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

D.C. CHARACTERISTICS

Conditions: $V_{CC} = 5V \pm 10\%$, $T_A = 0°C$ to $70°C$

Symbol	Parameter	Min	Max	Units	Test Conditions
V_C	Input Clamp Voltage		−1	V	$I_C = -5$ mA
I_{CC}	Power Supply Current		160	mA	
I_F	Forward Input Current		−0.2	mA	$V_F = 0.45V$
I_R	Reverse Input Current		50	μA	$V_R = 5.25V$
V_{OL}	Output Low Voltage		.45	V	$I_{OL} = 32$ mA
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -5$ mA
I_{OFF}	Output Off Current		±50	μA	$V_{OFF} = 0.45$ to $5.25V$
V_{IL}	Input Low Voltage		0.8	V	$V_{CC} = 5.0V$ See Note 1
V_{IH}	Input High Voltage	2.0		V	$V_{CC} = 5.0V$ See Note 1
C_{IN}	Input Capacitance		12	pF	F = 1 MHz $V_{BIAS} = 2.5V$, $V_{CC} = 5V$ $T_A = 25°C$

NOTE: 1. Output Loading $I_{OL} = 32$mA, $I_{OH} = -5$mA, $C_L = 300$pF.

A.C. CHARACTERISTICS

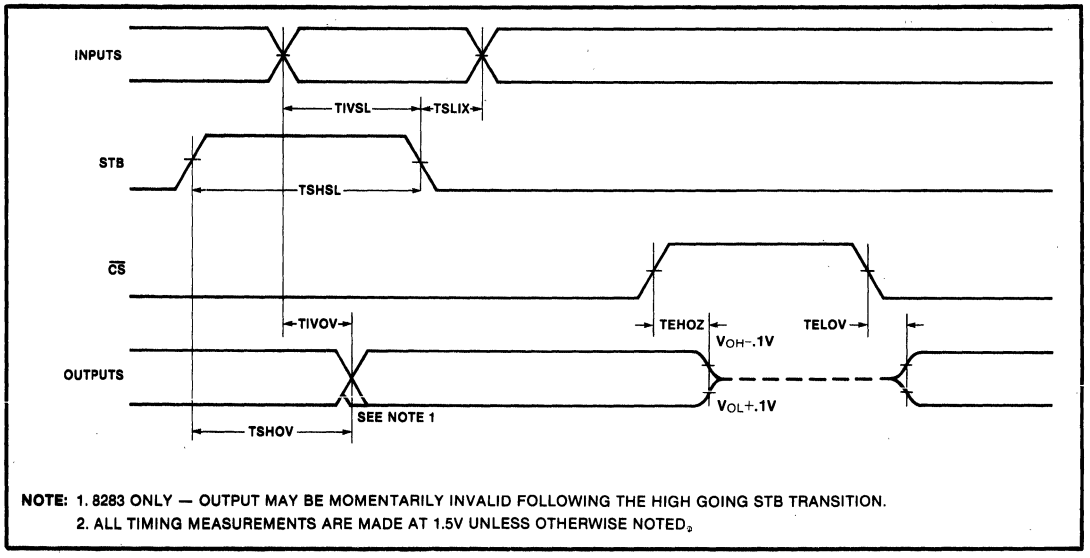
Conditions: $V_{CC} = 5V \pm 10\%$, $T_A = 0°C$ to $70°C$

Loading: Outputs — $I_{OL} = 32$ mA, $I_{OH} = -5$ mA, $C_L = 300$ pF

Symbol	Parameter	Min	Max	Units	Test Conditions
TIVOV	Input to Output Delay				(See Note 1)
	—Inverting	5	22	ns	
	—Non-Inverting	5	30	ns	
TSHOV	STB to Output Delay				
	—Inverting	10	40	ns	
	—Non-Inverting	10	45	ns	
TEHOZ	Output Disable Time	5	18	ns	
TELOV	Output Enable Time	10	30	ns	
TIVSL	Input to STB Setup Time	0		ns	
TSLIX	Input to STB Hold Time	25		ns	
TSHSL	STB High Time	15		ns	

NOTE: 1. See waveforms and test load circuit on following page.

WAVEFORMS



NOTE: 1. 8283 ONLY — OUTPUT MAY BE MOMENTARILY INVALID FOLLOWING THE HIGH GOING STB TRANSITION.
2. ALL TIMING MEASUREMENTS ARE MADE AT 1.5V UNLESS OTHERWISE NOTED.

Figure 4. Timing Diagram

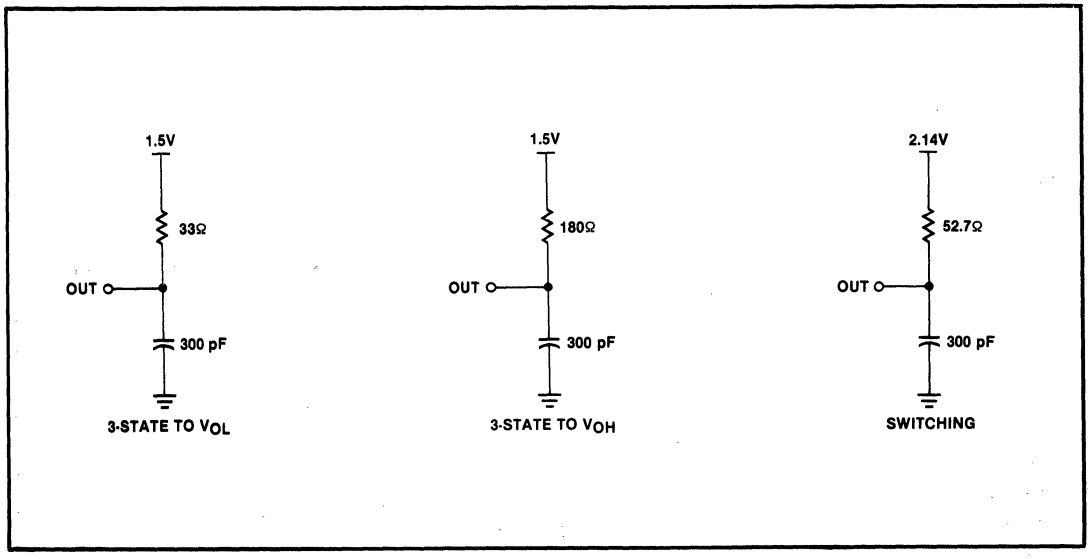


Figure 5. Output Test Load Circuits

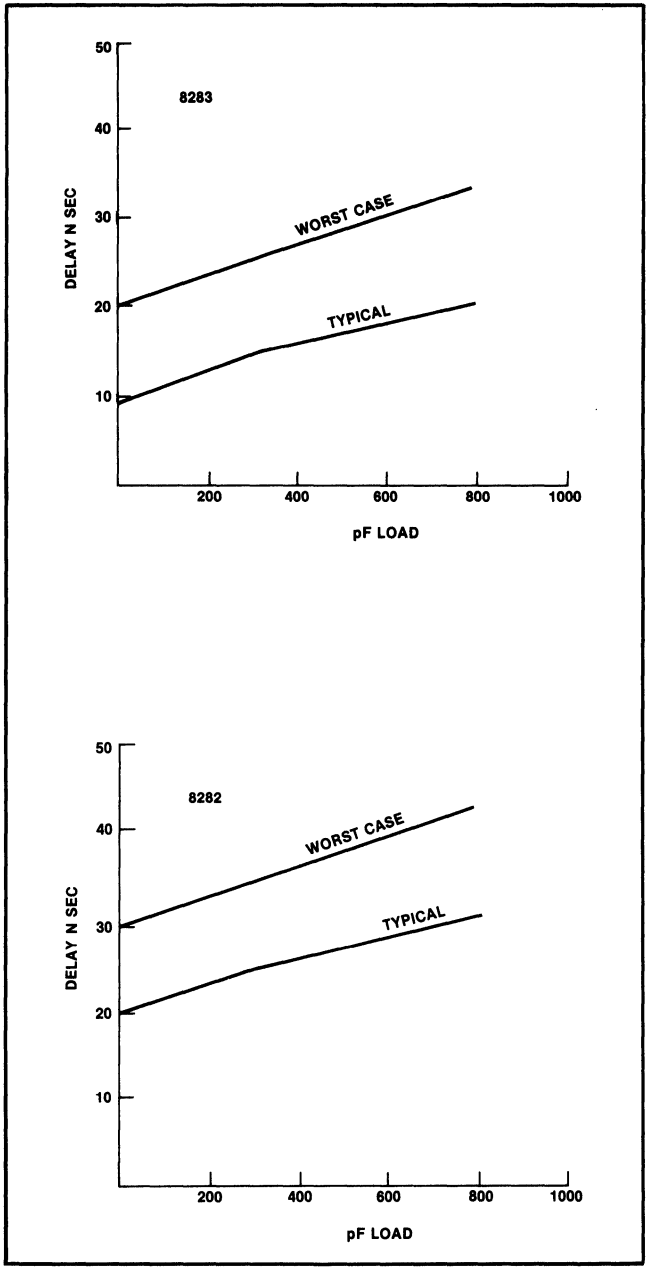


Figure 6. Output Delay vs. Capacitance

8286/8287
OCTAL BUS TRANSCEIVER

- **Data Bus Buffer Driver for iAPX 86,88, MCS-80™, MCS-85™, and MCS-48™ Families**
- **High Output Drive Capability for Driving System Data Bus**
- **Fully Parallel 8-Bit Transceivers**
- **3-State Outputs**
- **20-Pin Package with 0.3" Center**
- **No Output Low Noise when Entering or Leaving High Impedance State**

The 8286 and 8287 are 8-bit bipolar transceivers with 3-state outputs. The 8287 inverts the input data at its outputs while the 8286 does not. Thus, a wide variety of applications for buffering in microcomputer systems can be met.

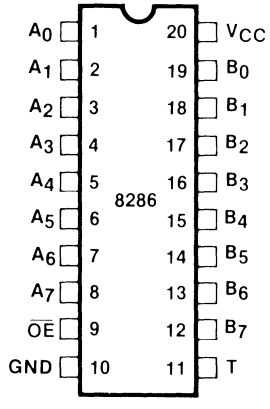


Figure 1. 8286 Pin Configuration

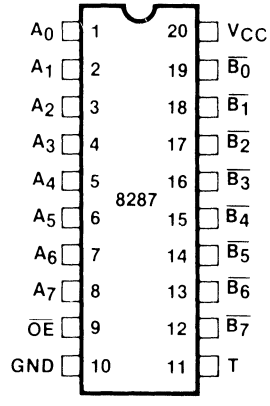


Figure 2. 8287 Pin Configuration

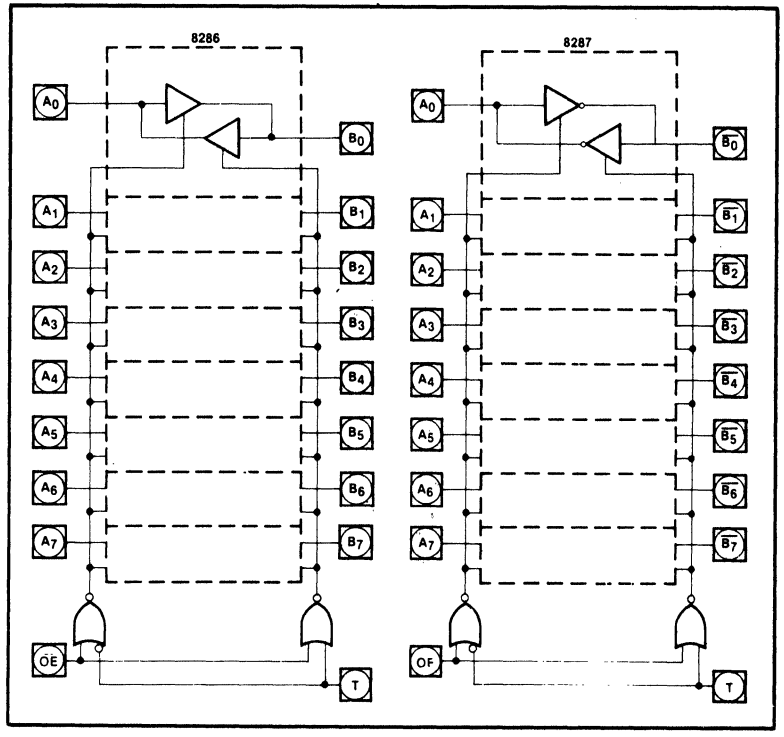


Figure 3. Logic Diagrams

Table 1. Pin Description

Pin	Description
T	TRANSMIT (Input). T is an input control signal used to control the direction of the transceivers. When HIGH, it configures the transceiver's B_0-B_7 as outputs with A_0-A_7 as inputs. T LOW configures A_0-A_7 as the outputs with B_0-B_7 serving as the inputs.
$\overline{OE}$	OUTPUT ENABLE (Input). $\overline{OE}$ is an input control signal used to enable the appropriate output driver (as selected by T) onto its respective bus. This signal is active LOW.
A_0-A_7	LOCAL BUS DATA PINS (Input/Output). These pins serve to either present data to or accept data from the processor's local bus depending upon the state of the T pin.
B_0-B_7 (8286) $\overline{B_0}$-$\overline{B_7}$ (8287)	SYSTEM BUS DATA PINS (Input/Output). These pins serve to either present data to or accept data from the system bus depending upon the state of the T pin.

FUNCTIONAL DESCRIPTION

The 8286 and 8287 transceivers are 8-bit transceivers with high impedance outputs. With T active HIGH and $\overline{OE}$ active LOW, data at the A_0-A_7 pins is driven onto the B_0-B_7 pins. With T inactive LOW and $\overline{OE}$ active LOW, data at the B_0-B_7 pins is driven onto the A_0-A_7 pins. No output low glitching will occur whenever the transceivers are entering or leaving the high impedance state.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
All Output and Supply Voltages . . . . . −0.5V to +7V
All Input Voltages . . . . . −1.0V to +5.5V
Power Dissipation . . . . . 1 Watt

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

D.C. CHARACTERISTICS FOR 8286/8287

Conditions: V_{CC} = 5V ±10% T_A = 0°C to 70°C

Symbol	Parameter	Min	Max	Units	Test Conditions
V_C	Input Clamp Voltage		-1	V	I_C = -5 mA
I_{CC}	Power Supply Current—8287 —8286		130 160	mA mA	
I_F	Forward Input Current		-0.2	mA	V_F = 0.45V
I_R	Reverse Input Current		50	μA	V_R = 5.25V
V_{OL}	Output Low Voltage —B Outputs —A Outputs		.45 .45	V V	I_{OL} = 32 mA I_{OL} = 16 mA
V_{OH}	Output High Voltage —B Outputs —A Outputs	2.4 2.4		V V	I_{OH} = -5 mA I_{OH} = -1 mA
I_{OFF} I_{OFF}	Output Off Current Output Off Current		I_F I_R		V_{OFF} = 0.45V V_{OFF} = 5.25V
V_{IL}	Input Low Voltage —A Side —B Side		0.8 0.9	V V	V_{CC} = 5.0V, See Note 1 V_{CC} = 5.0V, See Note 1
V_{IH}	Input High Voltage	2.0		V	V_{CC} = 5.0V, See Note 1
C_{IN}	Input Capacitance		12	pF	F = 1 MHz V_{BIAS} = 2.5V, V_{CC} = 5V T_A = 25°C

NOTE: 1. B Outputs — I_{OL} = 32 mA, I_{OH} = −5 mA, C_L = 300 pF; A Outputs — I_{OL} = 16 mA, I_{OH} = −1 mA, C_L = 100 pF.

A.C. CHARACTERISTICS FOR 8286/8287

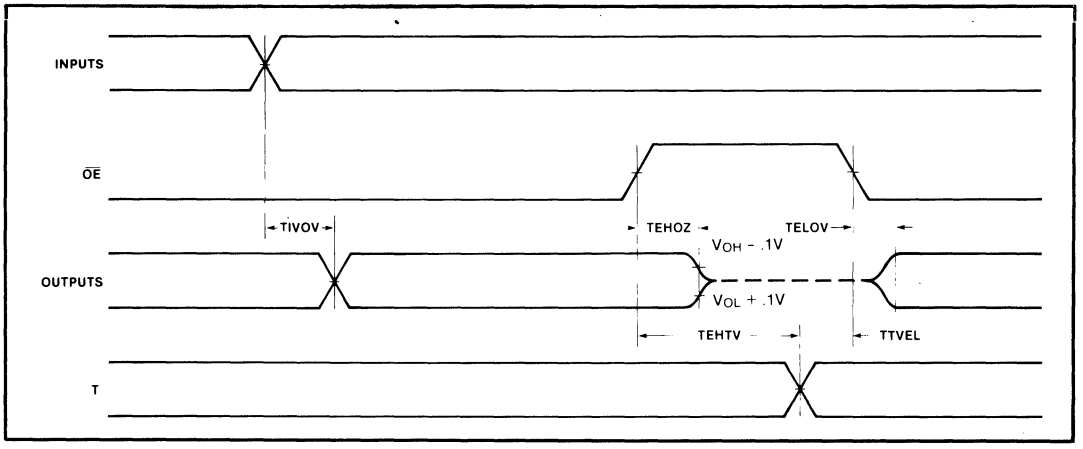
Conditions: V_{CC} = 5V ±10%, T_A = 0°C to 70°C

Loading: B Outputs — I_{OL} = 32 mA, I_{OH} = -5 mA, C_L = 300 pF
A Outputs — I_{OL} = 16 mA, I_{OH} = -1 mA, C_L = 100 pF

Symbol	Parameter	Min	Max	Units	Test Conditions
TIVOV	Input to Output Delay Inverting Non-Inverting	 5 5	 22 30	 ns ns	(See Note 1)
TEHTV	Transmit/Receive Hold Time	5		ns	
TTVEL	Transmit/Receive Setup	10		ns	
TEHOZ	Output Disable Time	5	18	ns	
TELOV	Output Enable Time	10	30	ns	

NOTE: 1. See waveforms and test load circuit on following page.

WAVEFORMS



NOTE: 1. All timing measurements are made at 1.5V unless otherwise noted.

Figure 4. 8286/8287 Timing

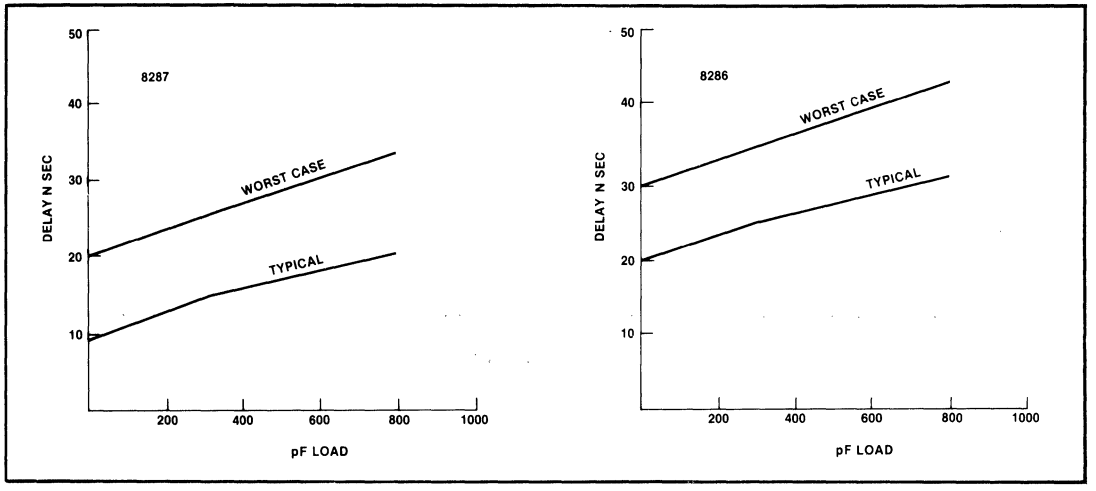


Figure 5. Output Delay vs. Capacitance

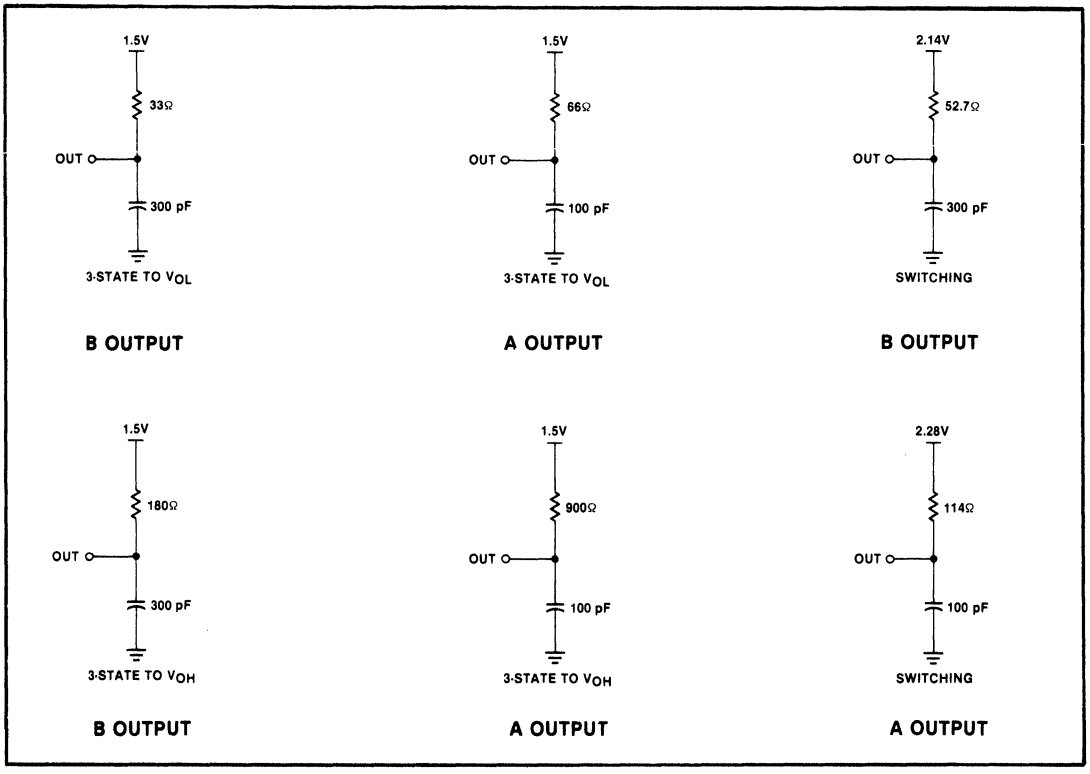


Figure 5. Test Load Circuits

Peripherals

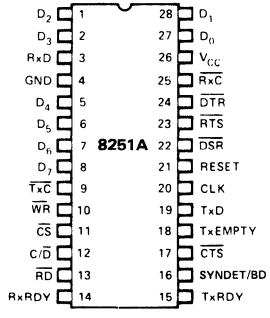
9

8251A/S2657
PROGRAMMABLE COMMUNICATION INTERFACE

- **Synchronous and Asynchronous Operation**
- **Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion**
- **Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling.**
- **Synchronous Baud Rate — DC to 64K Baud**
- **Asynchronous Baud Rate — DC to 19.2K Baud**
- **Full Duplex, Double Buffered, Transmitter and Receiver**
- **Error Detection — Parity, Overrun and Framing**
- **Fully Compatible with 8080/8085 CPU**
- **28-Pin DIP Package**
- **All Inputs and Outputs are TTL Compatible**
- **Single +5V Supply**
- **Single TTL Clock**

The Intel® 8251A is the enhanced version of the industry standard, Intel® 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's new high performance family of microprocessors such as the 8085. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is constructed using N-channel silicon gate technology.

PIN CONFIGURATION

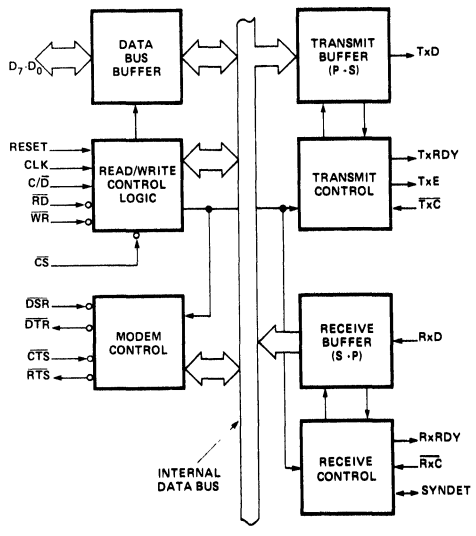


PIN NAMES

D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
$\overline{RD}$	Read Data Command
$\overline{WR}$	Write Data or Control Command
$\overline{CS}$	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
$\overline{TxC}$	Transmitter Clock
TxD	Transmitter Data
$\overline{RxC}$	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char from 8080)

$\overline{DSR}$	Data Set Ready
$\overline{DTR}$	Data Terminal Ready
SYNDET/BD	Sync Detect/ Break Detect
$\overline{RTS}$	Request to Send Data
$\overline{CTS}$	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

BLOCK DIAGRAM



FEATURES AND ENHANCEMENTS

8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors that includes the new 8085 CPU and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.
- Fully compatible with Intel's new industry standard, the MCS-85.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . 0°C to 70°C
Storage Temperature . . . . . –65°C to +150°C
Voltage On Any Pin
With Respect to Ground . . . . . –0.5V to +7V
Power Dissipation . . . . . 1 Watt

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

D.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} TO 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} TO 0.45V
I_{CC}	Power Supply Current		100	mA	All Outputs = High

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND

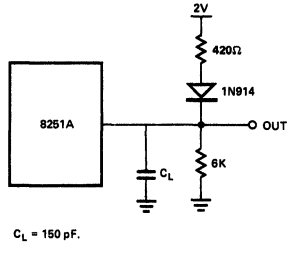


Figure 16. Test Load Circuit

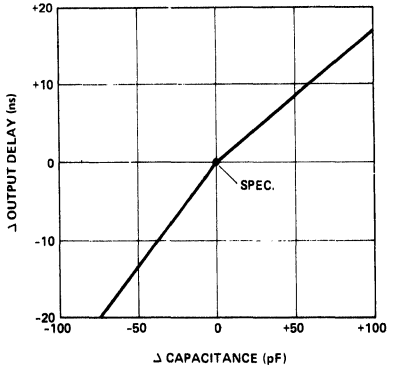


Figure 17. Typical Δ Output Delay vs. Δ Capacitance (pF)

A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Bus Parameters (Note 1)

Read Cycle:

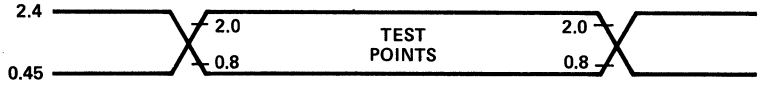
SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	50		ns	Note 2
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	50		ns	Note 2
t_{RR}	$\overline{READ}$ Pulse Width	250		ns	
t_{RD}	Data Delay from $\overline{READ}$		250	ns	3, C_L = 150 pF
t_{DF}	$\overline{READ}$ to Data Floating	10	100	ns	

Write Cycle:

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{AW}	Address Stable Before $\overline{WRITE}$	50		ns	
t_{WA}	Address Hold Time for $\overline{WRITE}$	50		ns	
t_{WW}	$\overline{WRITE}$ Pulse Width	250		ns	
t_{DW}	Data Set Up Time for $\overline{WRITE}$	150		ns	
t_{WD}	Data Hold Time for $\overline{WRITE}$	50		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	Note 4

NOTES:
1. AC timings measured V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select ($\overline{CS}$) and Command/Data (C/$\overline{D}$) are considered as Addresses.
3. Assumes that Address is valid before $\overline{RD}\downarrow$.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1. Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.

Input Waveforms for AC Tests



Other Timings:

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{CY}	Clock Period	320	1350	ns	Notes 5, 6
t_{ϕ}	Clock High Pulse Width	140	t_{CY-90}	ns	
$t_{\overline{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time		20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{TxC}$		1	μs	
f_{Tx}	Transmitter Input Clock Frequency				
	1x Baud Rate	DC	64	kHz	
	16x Baud Rate	DC	310	kHz	
	64x Baud Rate	DC	615	kHz	
t_{TPW}	Transmitter Input Clock Pulse Width				
	1x Baud Rate	12		t_{CY}	
	16x and 64x Baud Rate	1		t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay				
	1x Baud Rate	15		t_{CY}	
	16x and 64x Baud Rate	3		t_{CY}	
f_{Rx}	Receiver Input Clock Frequency				
	1x Baud Rate	DC	64	kHz	
	16x Baud Rate	DC	310	kHz	
	64x Baud Rate	DC	615	kHz	
t_{RPW}	Receiver Input Clock Pulse Width				
	1x Baud Rate	12		t_{CY}	
	16x and 64x Baud Rate	1		t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay				
	1x Baud Rate	15		t_{CY}	
	16x and 64x Baud Rate	3		t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of last Bit		8	t_{CY}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$		6	t_{CY}	Note 7
t_{RxRDY}	RxRDY Pin Delay from Center of last Bit		24	t_{CY}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$		6	t_{CY}	Note 7
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$		24	t_{CY}	Note 7
t_{ES}	External SYNDET Set-Up Time Before Falling Edge of $\overline{RxC}$	16		t_{CY}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit	20		t_{CY}	Note 7
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)	8		t_{CY}	Note 7
t_{CR}	Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)	20		t_{CY}	Note 7

5. The TxC and RxC frequencies have the following limitations with respect to CLK.
   For 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$
   For 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$

6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

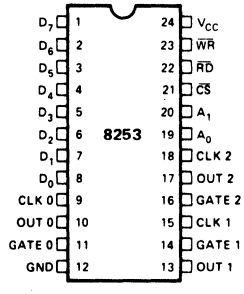
8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- **MCS—85™ Compatible 8253-5**
- **3 Independent 16-Bit Counters**
- **DC to 2 MHz**
- **Programmable Counter Modes**
- **Count Binary or BCD**
- **Single +5V Supply**
- **24-Pin Dual In-Line Package**

The Intel® 8253 is a programmable counter/timer chip designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

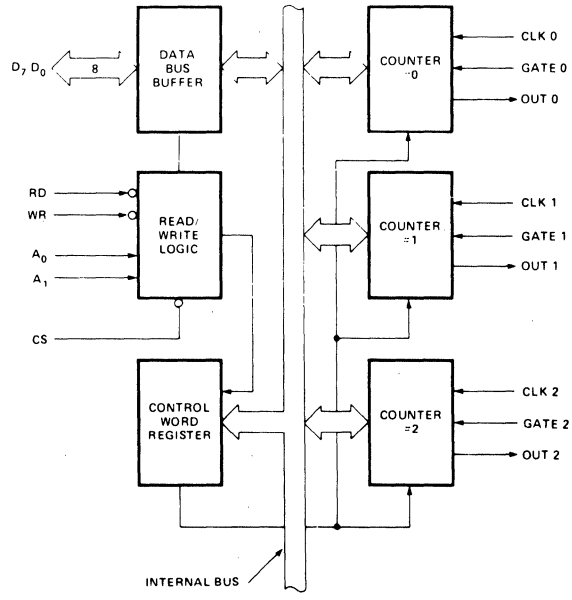
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (8 BIT)
CLK N	COUNTER CLOCK INPUTS
GATE N	COUNTER GATE INPUTS
OUT N	COUNTER OUTPUTS
$\overline{RD}$	READ COUNTER
WR	WRITE COMMAND OR DATA
CS	CHIP SELECT
A_0, A_1	COUNTER SELECT
V_{CC}	+5 VOLTS
GND	GROUND

BLOCK DIAGRAM



AFN-00745A-01

FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

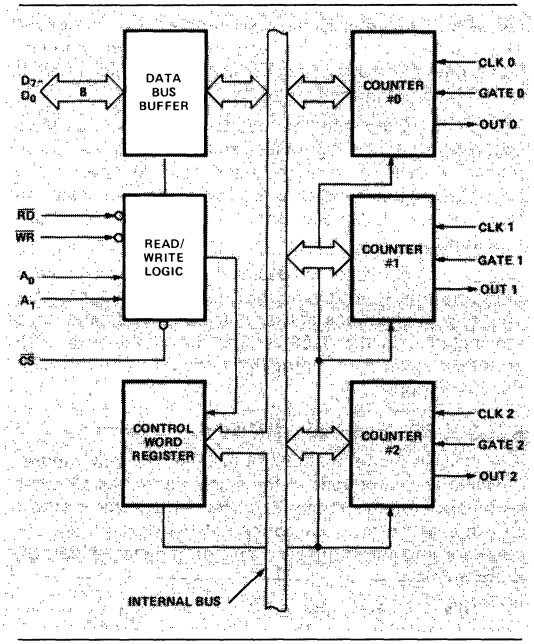


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel® 8205 for larger systems.

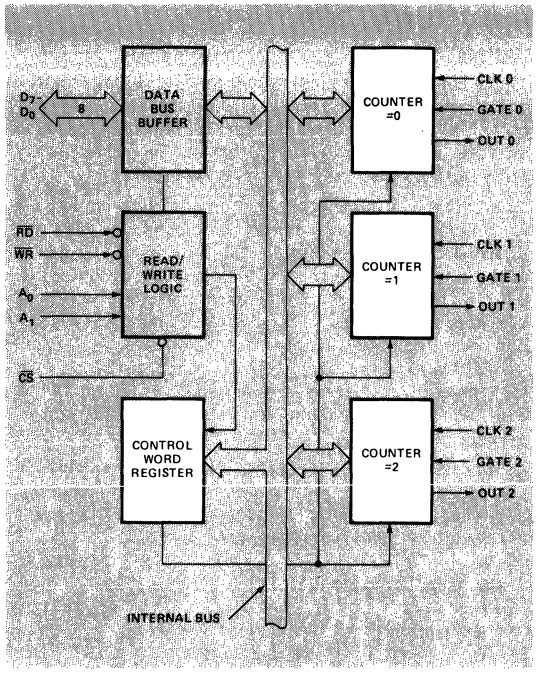


Figure 2. Block Diagram Showing Control Word Register and Counter Functions

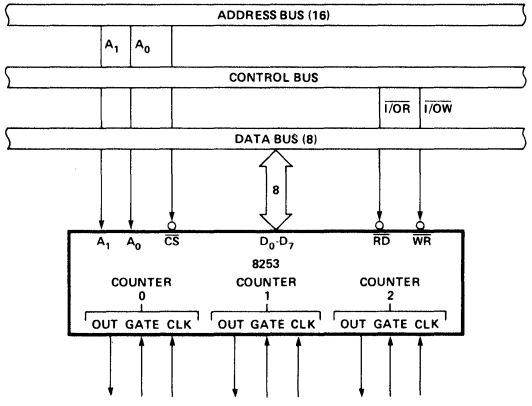


Figure 3. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. These control words program the MODE, Loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register. (A0, A1 = 11)

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL — Read/Load:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section)
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE Definition

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for (N + 1)/2 counts and low for (N – 1)/2 counts.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

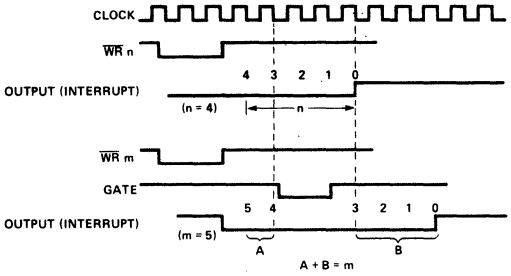
If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is low. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

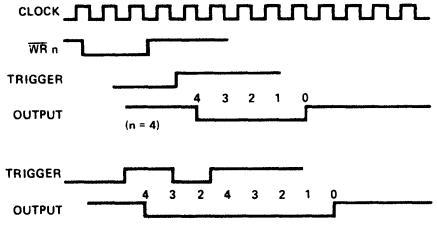
Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	— —	Enables counting
1	— —	1) Initiates counting 2) Resets output after next clock	— —
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	— —	Enables counting
5	— —	Initiates counting	— —

Figure 4. Gate Pin Operations Summary

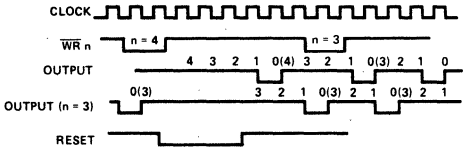
MODE 0: Interrupt on Terminal Count



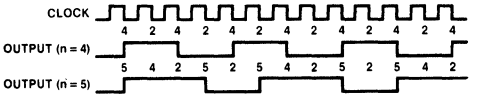
MODE 1: Programmable One-Shot



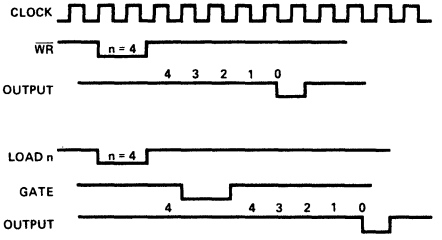
MODE 2: Rate Generator



MODE 3: Square Wave Generator



MODE 4: Software Triggered Strobe



MODE 5: Hardware Triggered Strobe

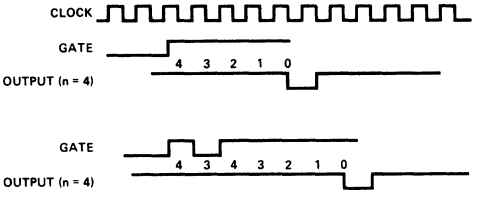


Figure 5. 8253 Timing Diagrams

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 6. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 7. Alternate Programming Formats

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

first I/O Read contains the least significant byte (LSB).

second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

MODE Register for Latching Count

A0, A1 = 11

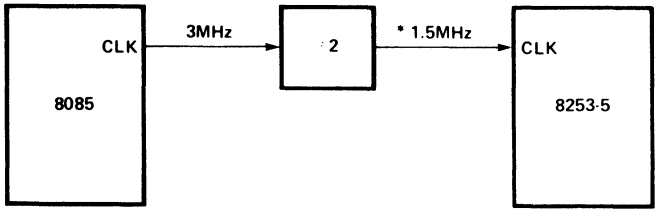
D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 — specify counter to be latched.

D5,D4 — 00 designates counter latching operation.

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 8. MCS-85™ Clock Interface*

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage On Any Pin
With Respect to Ground ..... −0.5 V to +7 V
Power Dissipation ..... 1 Watt

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 2
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V
I_{CC}	V_{CC} Supply Current		140	mA	

Note 1: 8253, I_{OL} = 1.6 mA; 8253-5, I_{OL} = 2.2 mA.
Note 2: 8253, I_{OH} = −150 μA; 8253-5, I_{OH} = −400 μA.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

A.C. CHARACTERISTICS T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Bus Parameters (Note 1)

Read Cycle:

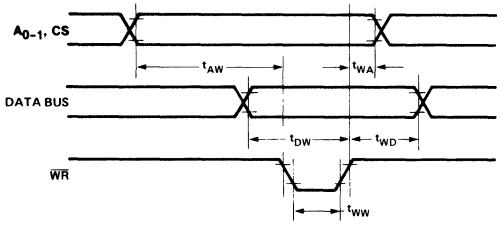
SYMBOL	PARAMETER	8253 MIN.	8253 MAX.	8253-5 MIN.	8253-5 MAX.	UNIT
t_{AR}	Address Stable Before $\overline{READ}$	50		30		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		ns
t_{RD}	Data Delay From $\overline{READ}$[2]		300		200	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	ns
t_{RV}	Recovery Time Between READ and Any Other Control Signal	1		1		μs

Write Cycle:

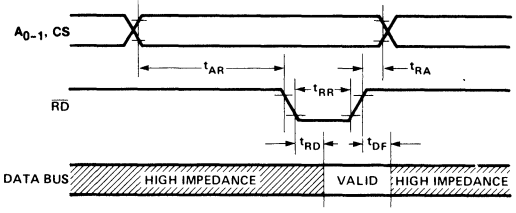
SYMBOL	PARAMETER	8253 MIN.	8253 MAX.	8253-5 MIN.	8253-5 MAX.	UNIT
t_{AW}	Address Stable Before $\overline{WRITE}$	50		30		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		250		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		ns
t_{RV}	Recovery Time Between $\overline{WRITE}$ and Any Other Control Signal	1		1		μs

Notes: 1. AC timings measured at V_{OH} = 2.2, V_{OL} = 0.8
2. Test Conditions: 8253, C_L = 100pF; 8253-5: C_L = 150pF.

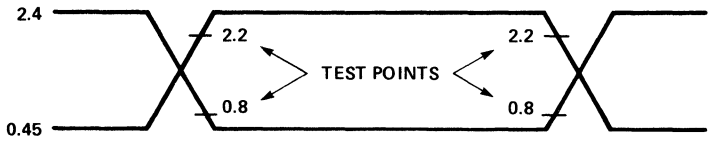
Write Timing:



Read Timing:



Input Waveforms for A.C. Tests:

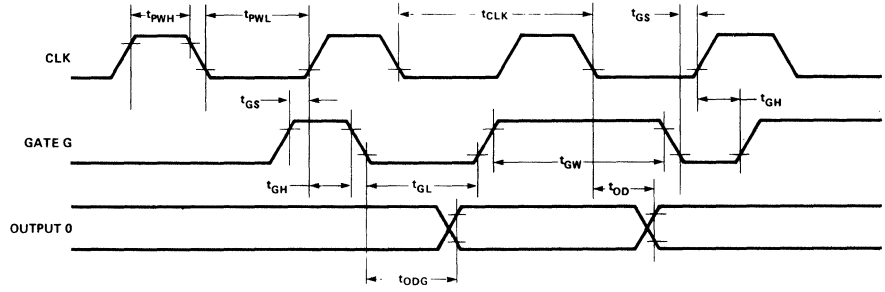


AFN-00745A-10

Clock and Gate Timing:

SYMBOL	PARAMETER	8253		8253-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{CLK}	Clock Period	380	dc	380	dc	ns
t_{PWH}	High Pulse Width	230		230		ns
t_{PWL}	Low Pulse Width	150		150		ns
t_{GW}	Gate Width High	150		150		ns
t_{GL}	Gate Width Low	100		100		ns
t_{GS}	Gate Set Up Time to CLK↑	100		100		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		ns
t_{OD}	Output Delay From CLK↓[1]		400		400	ns
t_{ODG}	Output Delay From Gate↓[1]		300		300	ns

Note 1: Test Conditions: 8253: C_L = 100pF; 8253-5: C_L = 150pF.

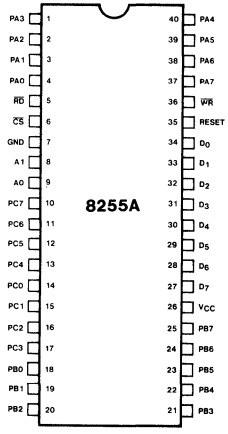


8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- **MCS-85™ Compatible 8255A-5**
- **24 Programmable I/O Pins**
- **Completely TTL Compatible**
- **Fully Compatible with Intel® Microprocessor Families**
- **Improved Timing Characteristics**
- **Direct Bit Set/Reset Capability Easing Control Application Interface**
- **40-Pin Dual In-Line Package**
- **Reduces System Package Count**
- **Improved DC Driving Capability**

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

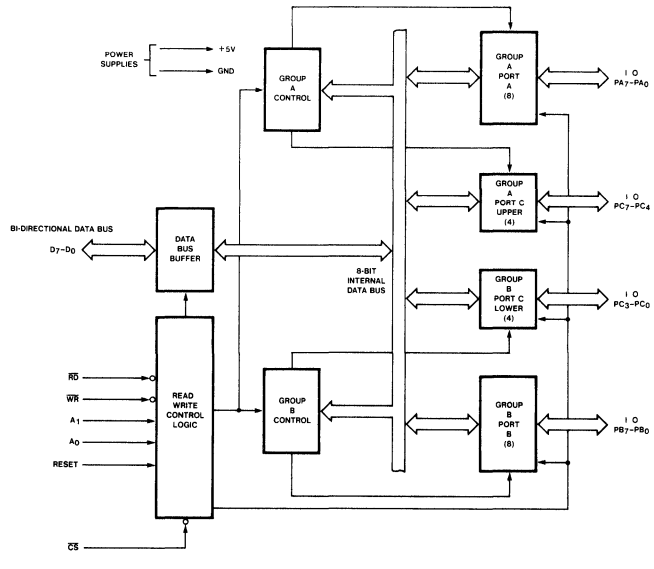
PIN CONFIGURATION



PIN NAMES

D_7–D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

8255A BLOCK DIAGRAM



8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communiction between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ⇒ DATA BUS
0	1	0	1	0	PORT B ⇒ DATA BUS
1	0	0	1	0	PORT C ⇒ DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS ⇒ PORT A
0	1	1	0	0	DATA BUS ⇒ PORT B
1	0	1	0	0	DATA BUS ⇒ PORT C
1	1	1	0	0	DATA BUS ⇒ CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS ⇒ 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS ⇒ 3-STATE

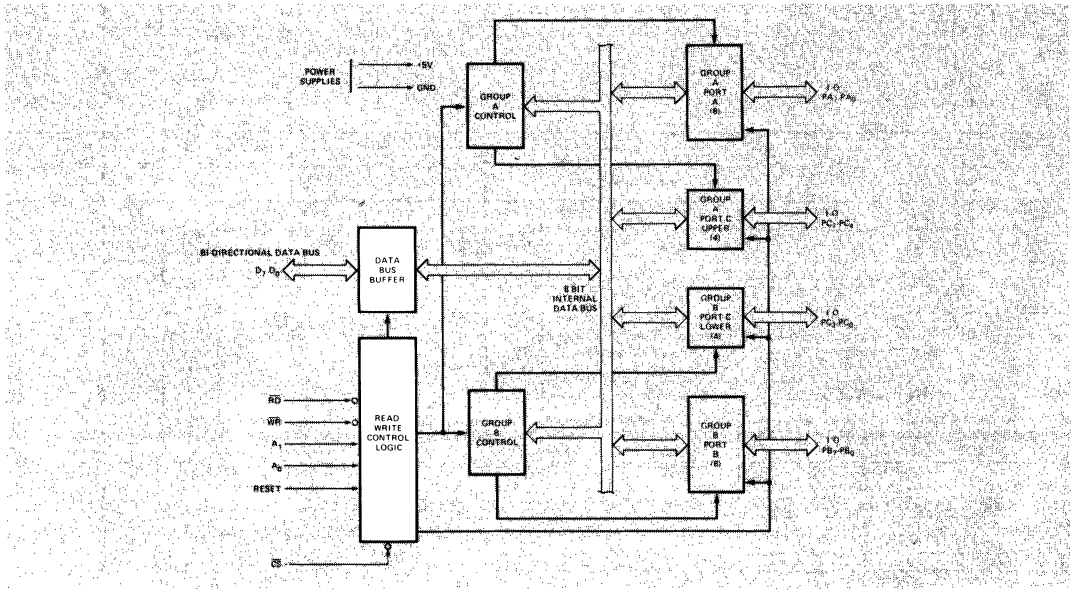


Figure 1. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "high on this input clears the control register and all ports (A, C, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

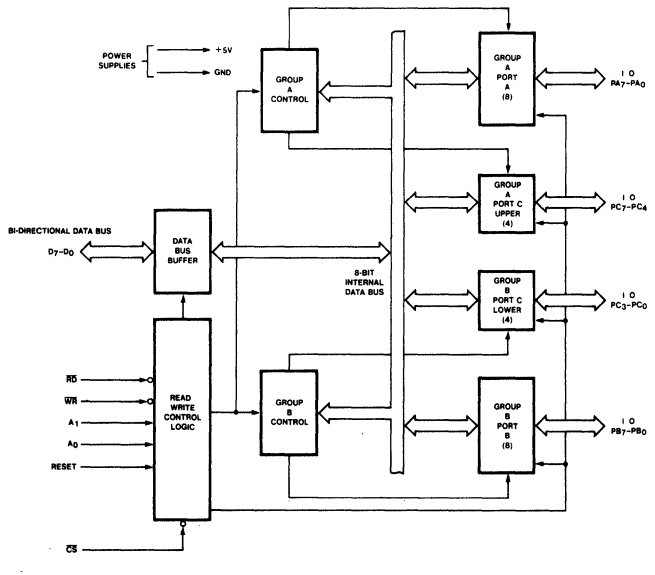
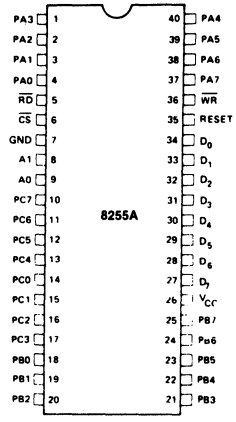


Figure 2. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

D_7, D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

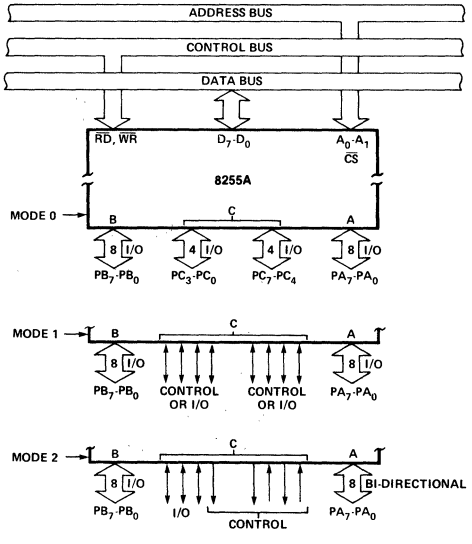


Figure 3. Basic Mode Definitions and Bus Interface

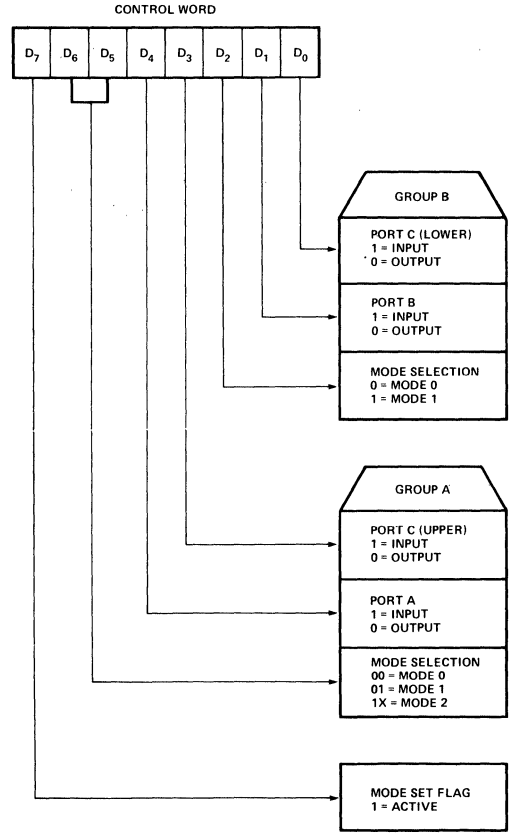


Figure 4. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

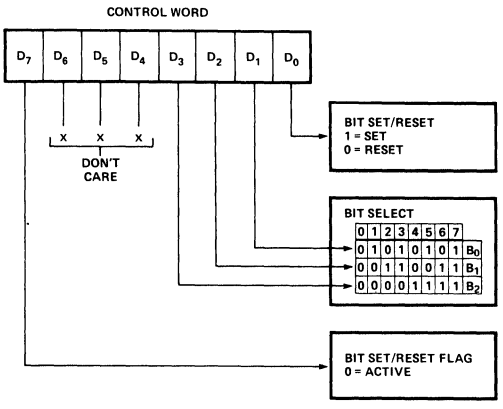


Figure 5. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable
(BIT-RESET) — INTE is RESET — Interrupt disable

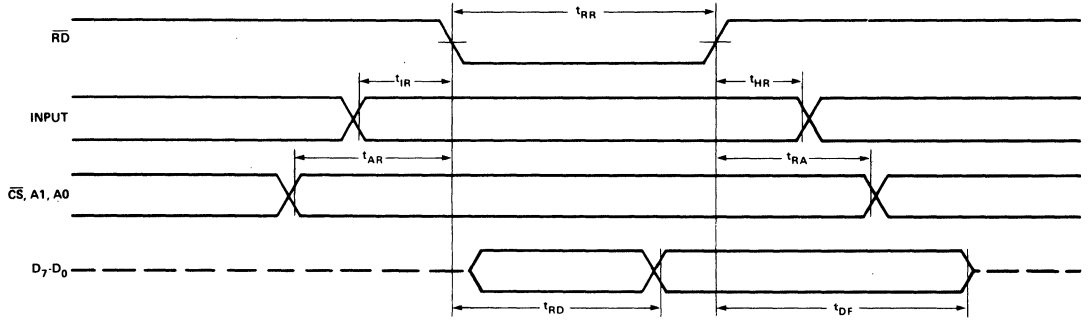
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

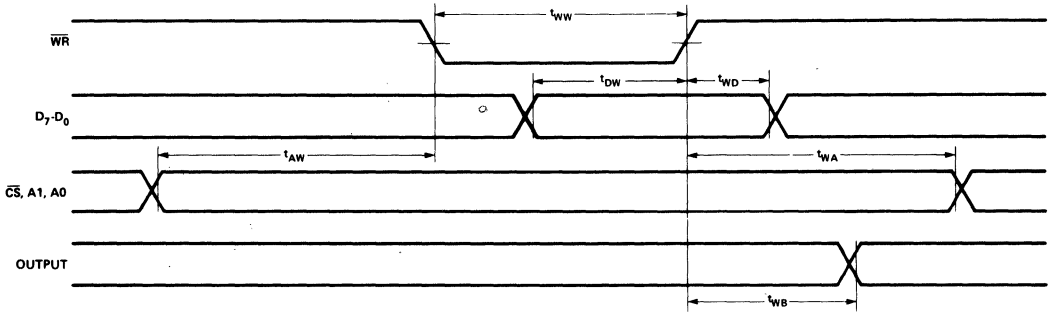
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)



MODE 0 (Basic Output)

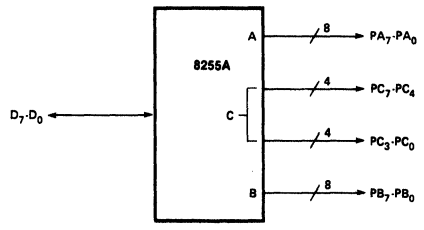
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

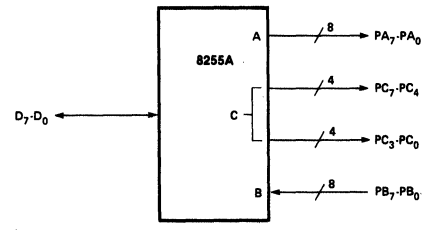
CONTROL WORD #0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



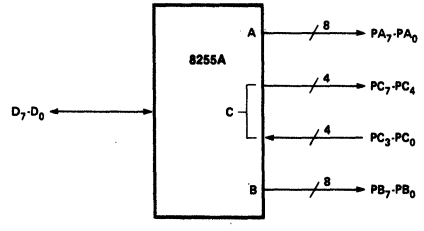
CONTROL WORD #2

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



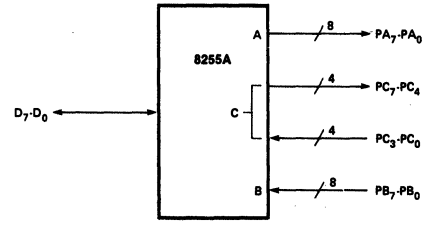
CONTROL WORD #1

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



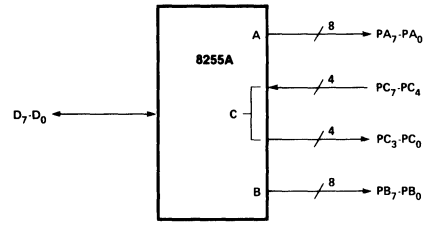
CONTROL WORD #3

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



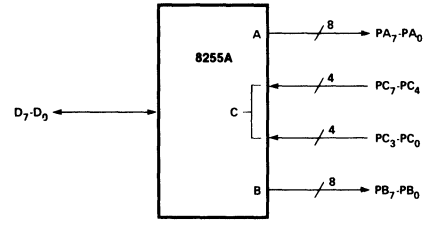
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



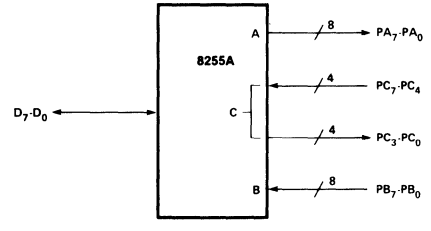
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



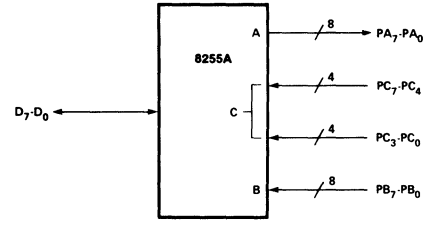
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



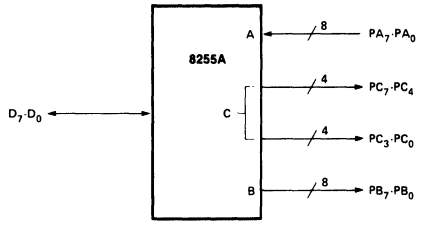
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



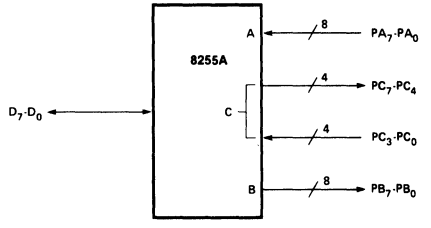
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



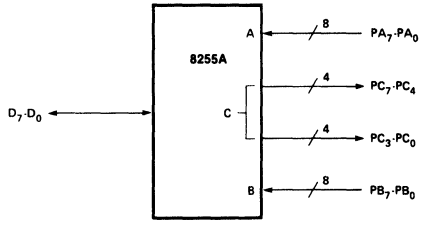
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



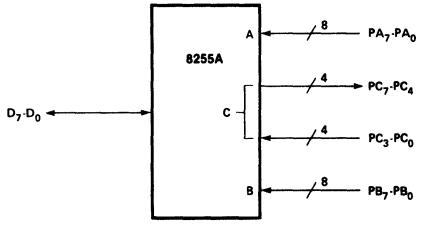
CONTROL WORD #10

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



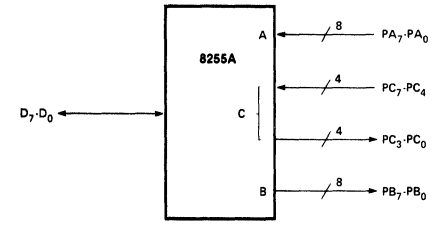
CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



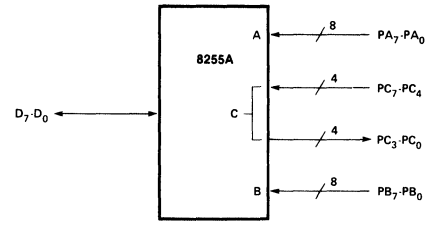
CONTROL WORD =12

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0



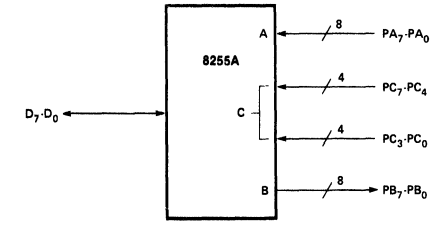
CONTROL WORD =14

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	0



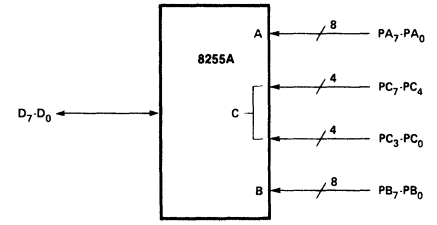
CONTROL WORD =13

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	1



CONTROL WORD #15

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	1



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

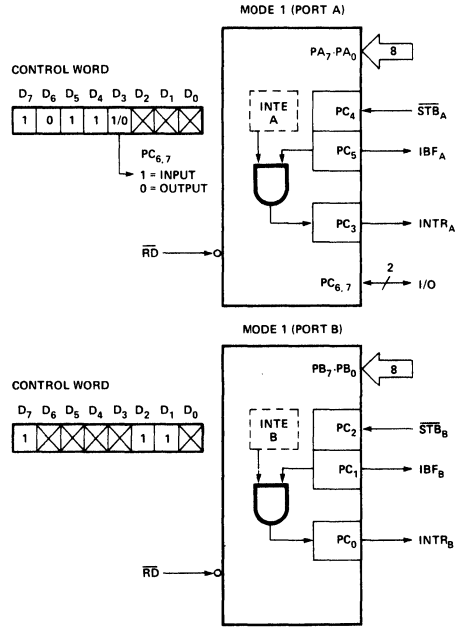


Figure 6. MODE 1 Input

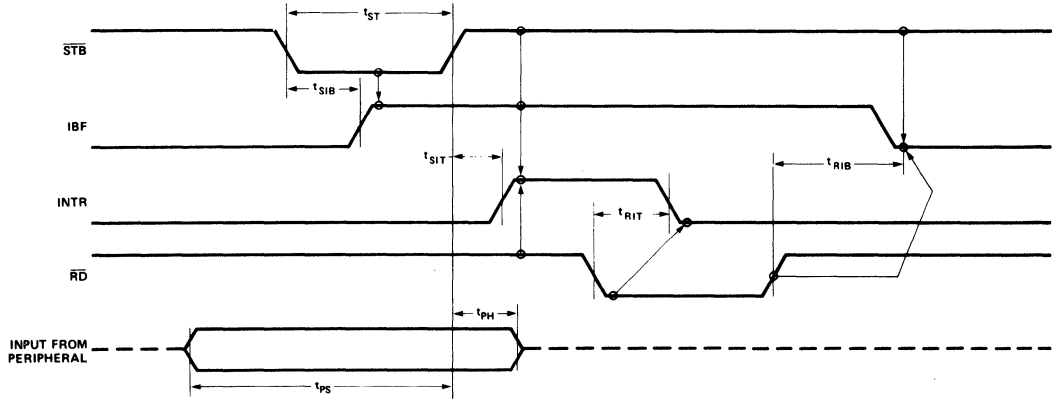


Figure 7. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by ACK Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

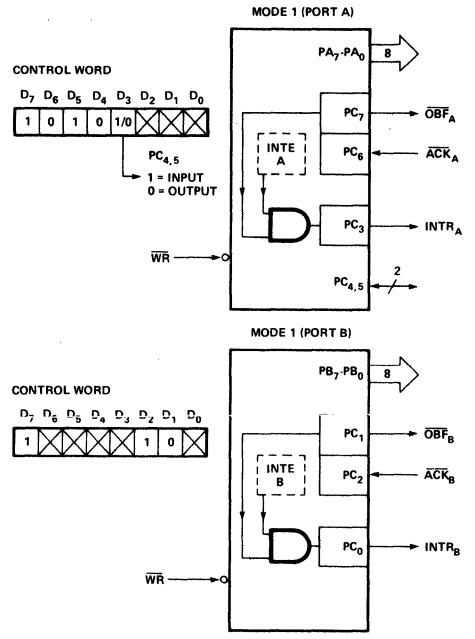


Figure 8. MODE 1 Output

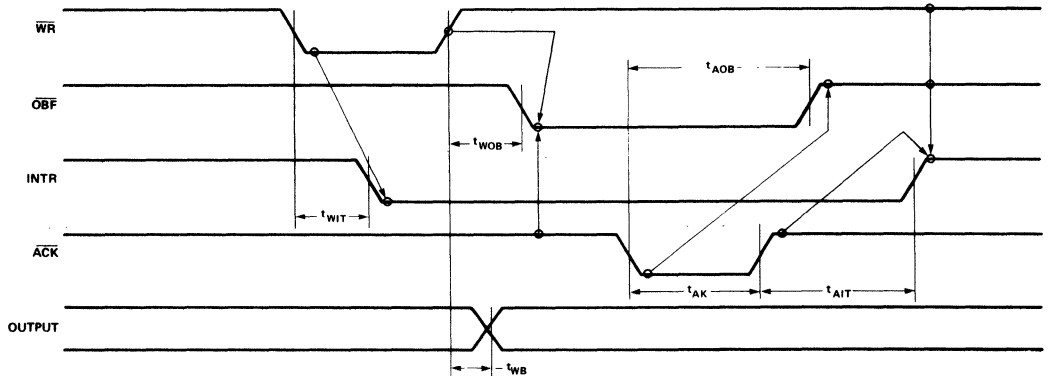


Figure 9. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

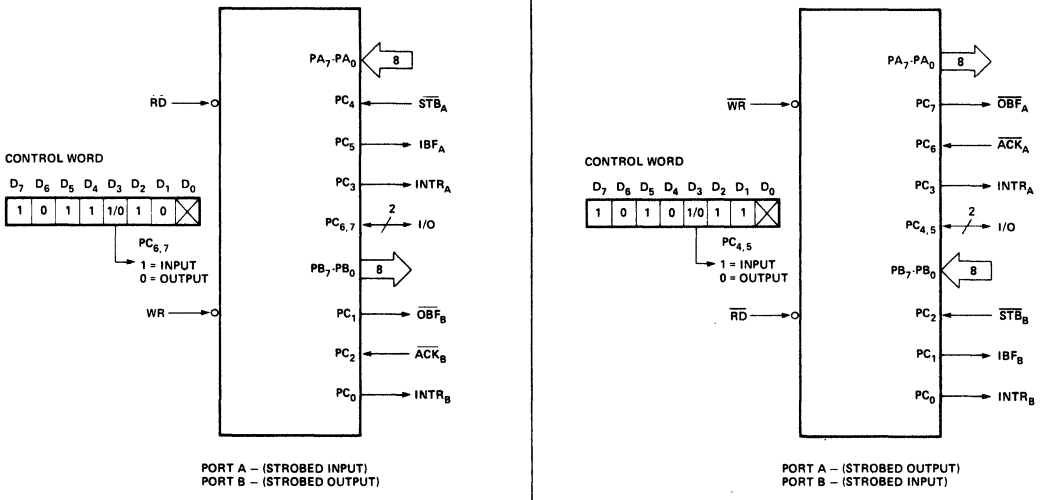


Figure 10. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{\text{OBF}}$ (Output Buffer Ful). The OBF output will go "low" to indicate that the CPU has written data out to port A.

$\overline{\text{ACK}}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{\text{STB}}$ (Strobe Input)

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

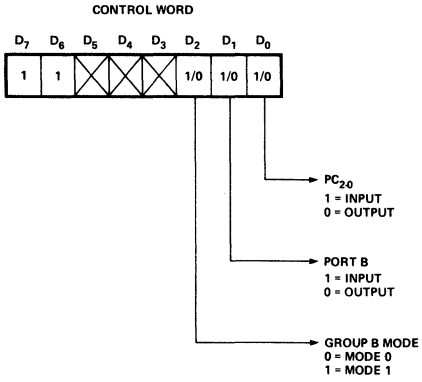


Figure 11. MODE Control Word

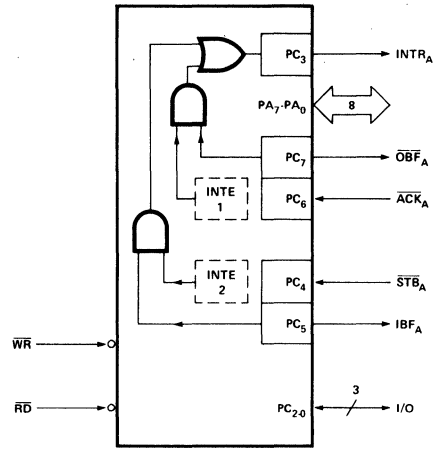


Figure 12. MODE 2

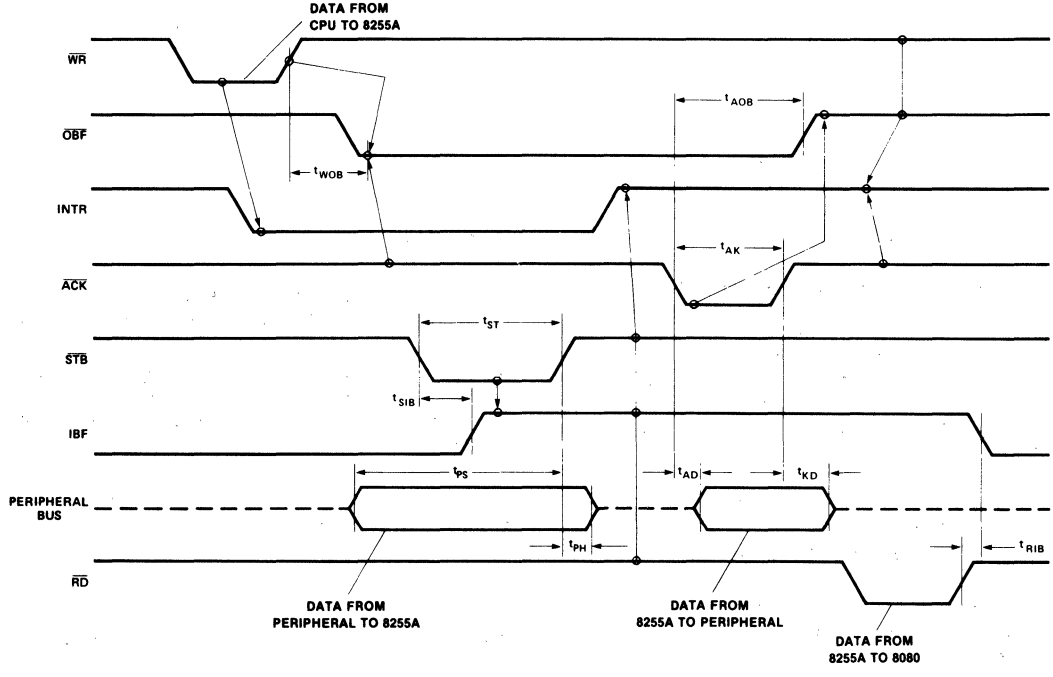


Figure 13. MODE 2 (Bidirectional)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

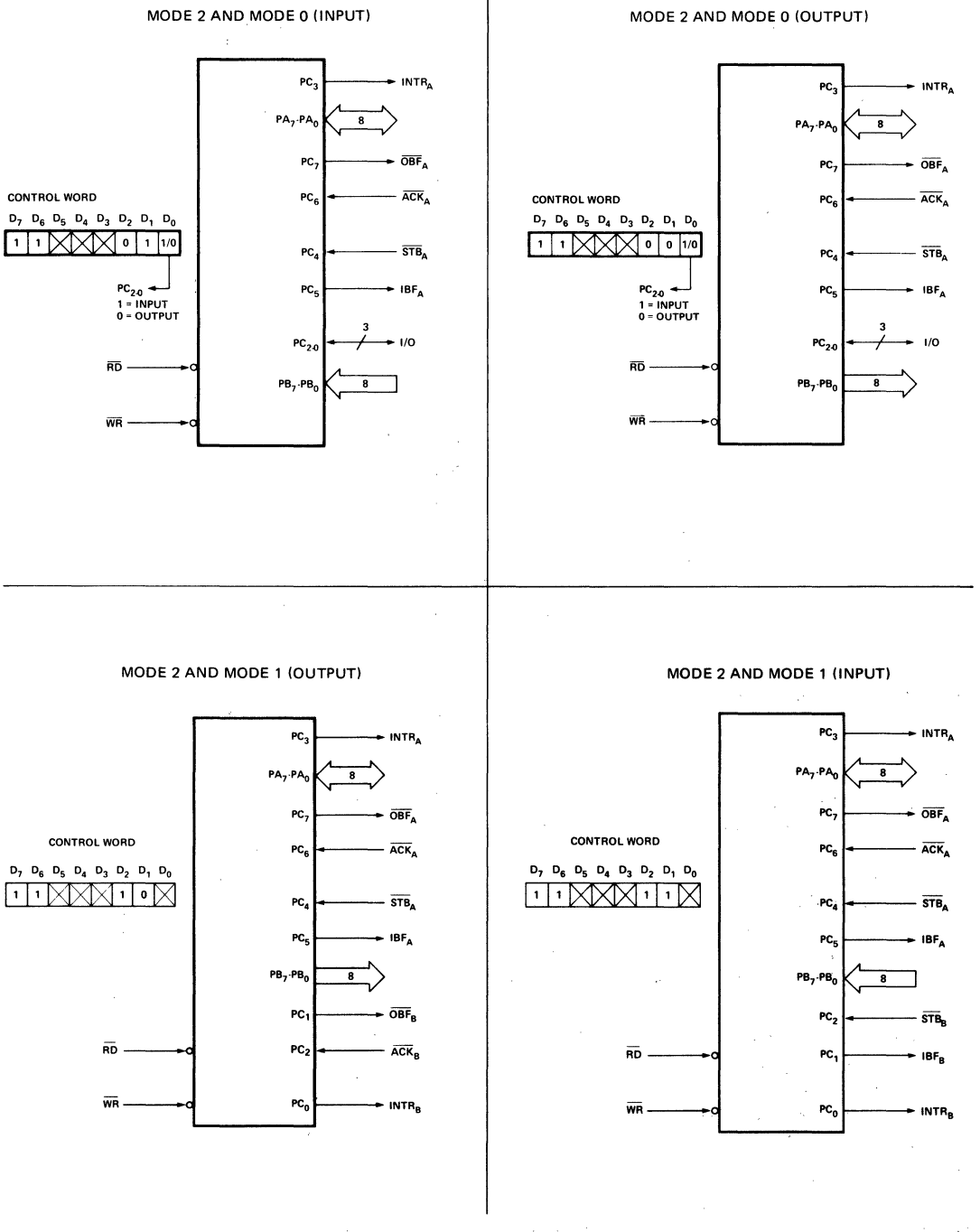


Figure 14. MODE 2 Combinations

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs –
All input lines can be accessed during a normal Port C read.

If Programmed as Outputs –
Bits in C upper (PC_7-PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3-PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

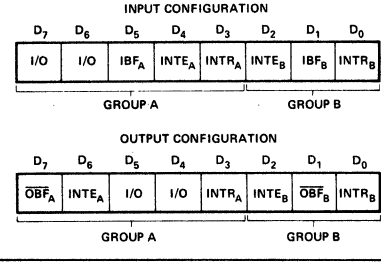


Figure 15. MODE 1 Status Word Format

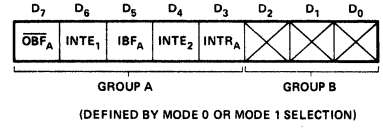


Figure 16. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 17 through 23 present a few examples of typical applications of the 8255A.

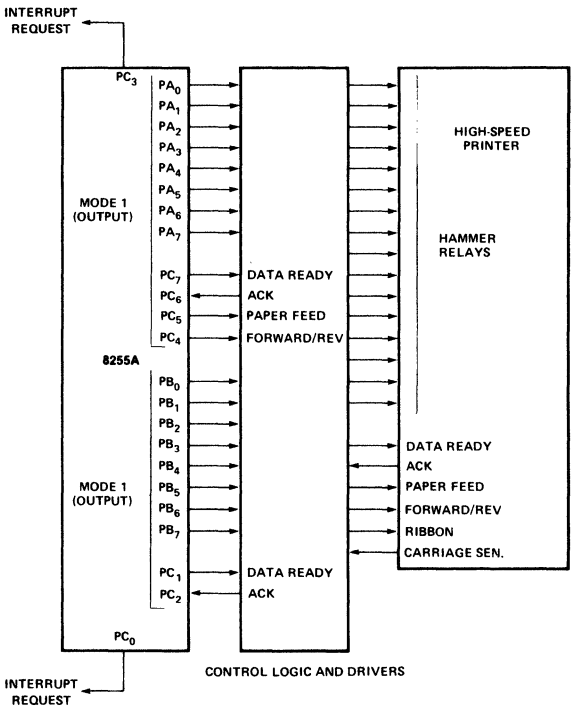


Figure 17. Printer Interface

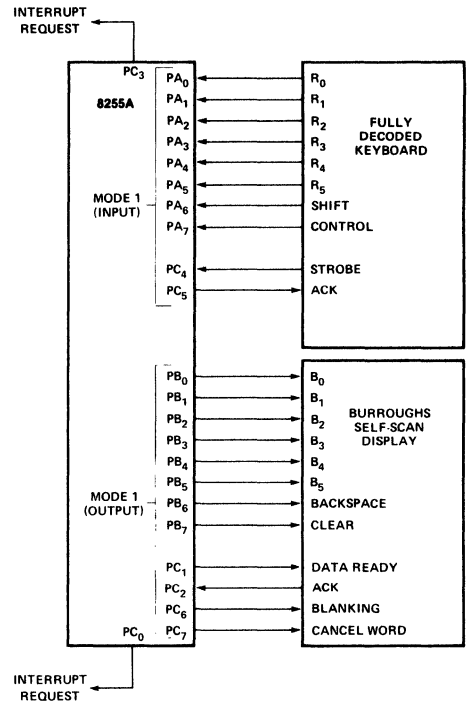


Figure 18. Keyboard and Display Interface

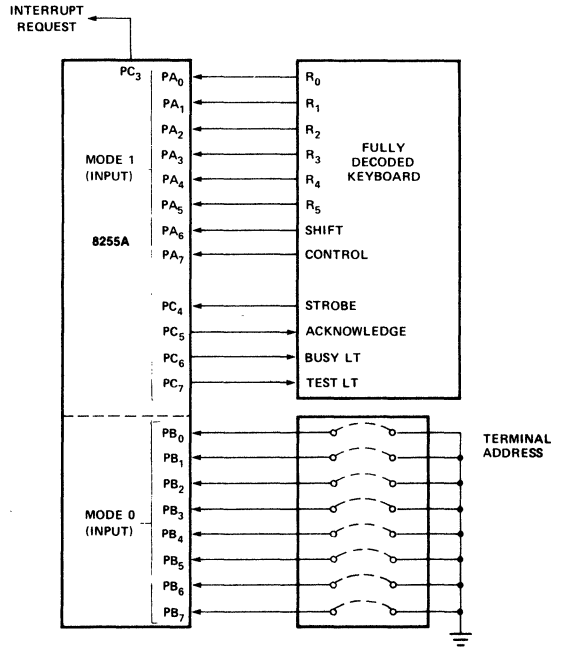


Figure 19. Keyboard and Terminal Address Interface

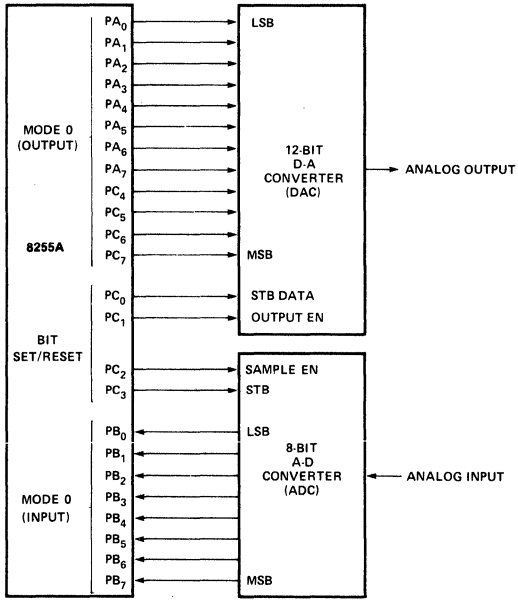


Figure 20. Digital to Analog, Analog to Digital

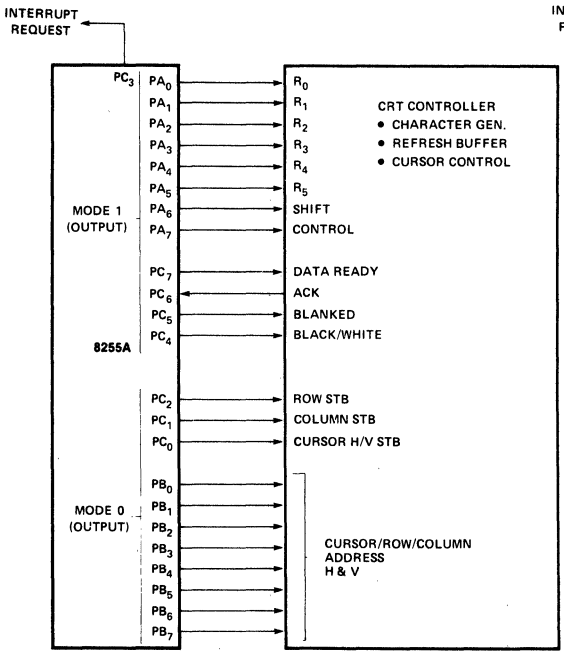


Figure 21. Basic CRT Controller Interface

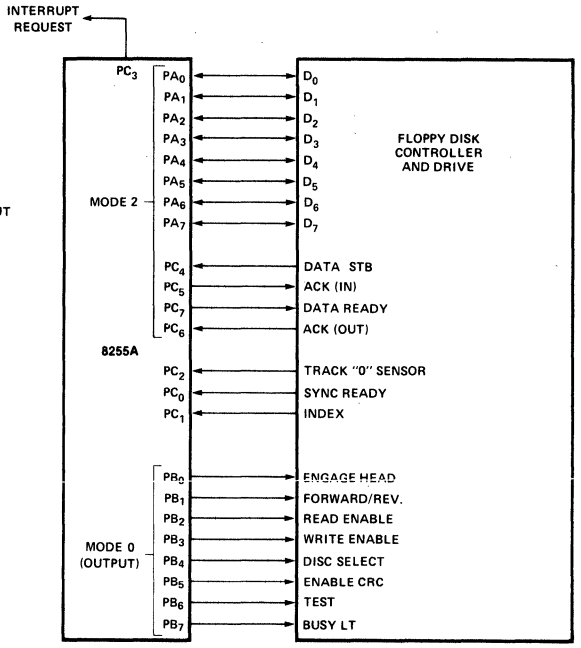


Figure 22. Basic Floppy Disc Interface

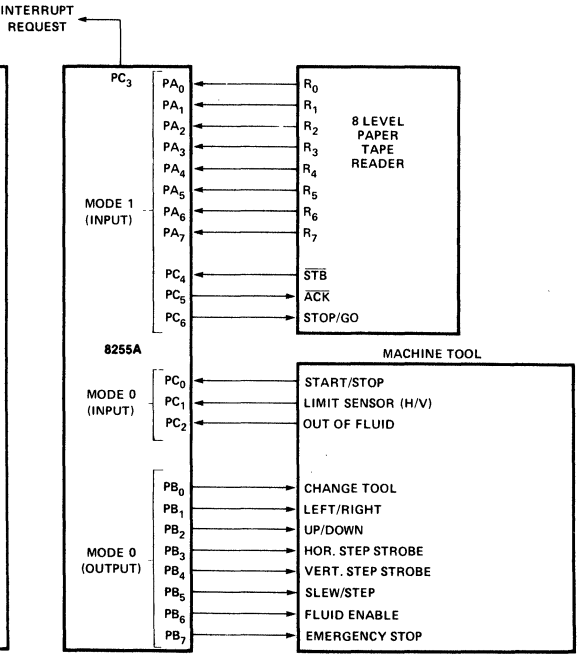


Figure 23. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . 0°C to 70°C
Storage Temperature . . . . . –65°C to +150°C
Voltage on Any Pin
With Respect to Ground . . . . . –0.5V to +7V
Power Dissipation . . . . . 1 Watt

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

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±5%; GND = 0V

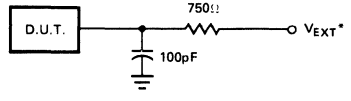
SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45	V	I_{OL} = 2.5mA
V_{OL}(PER)	Output Low Voltage (Peripheral Port)		0.45	V	I_{OL} = 1.7mA
V_{OH}(DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = -400μA
V_{OH}(PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = -200μA
I_{DAR}[1]	Darlington Drive Current	-1.0	-4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V

Note 1: Available on any 8 pins from Port B and C.

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND



*V_{EXT} is set at various voltages during testing to guarantee the specification.

Figure 24. Test Load Circuit (for dB)

A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = +5V ±5%; GND = 0V

NOTE: The 8255A-5 specifications are not final. Some parametric limits are subject to change.

Bus Parameters

Read:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ[1]		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

Write:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

Other Timings:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{WB}	WR = 1 to Output[1]		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output[1]		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0[1]		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1[1]		350		350	ns
t_{SIB}	STB = 0 to IBF = 1[1]		300		300	ns
t_{RIB}	RD = 1 to IBF = 0[1]		300		300	ns
t_{RIT}	RD = 0 to INTR = 0[1]		400		400	ns
t_{SIT}	STB = 1 to INTR = 1[1]		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1[1]		350		350	ns
t_{WIT}	WR = 0 to INTR = 0[1]		850		850	ns

Notes: 1. Test Conditions: 8255A: C_L = 100pF; 8255A-5: C_L = 150pF.
2. Period of Reset pulse must be at least 50μs during or after power on. Subsequent Reset pulse can be 500 ns min.

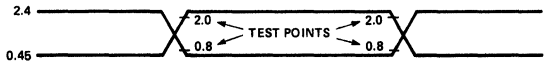


Figure 25. Input Waveforms for A.C. Tests

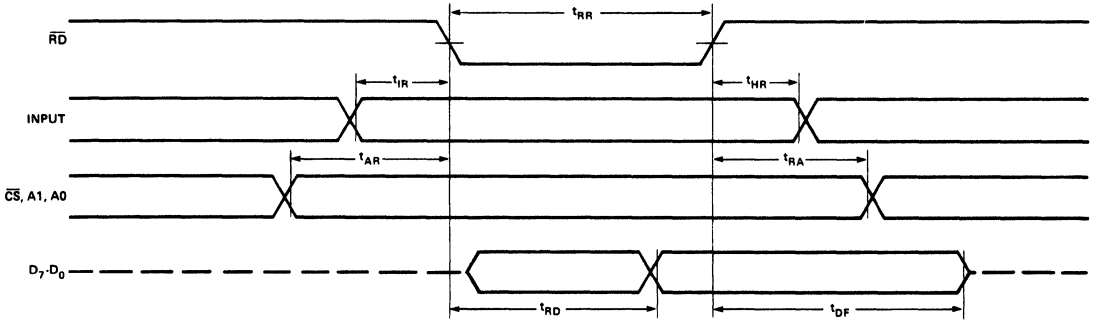


Figure 26. MODE 0 (Basic Input)

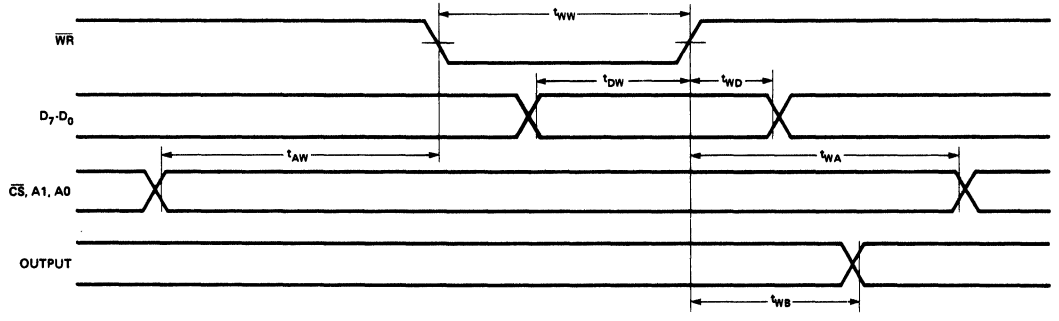


Figure 27. MODE 0 (Basic Output)

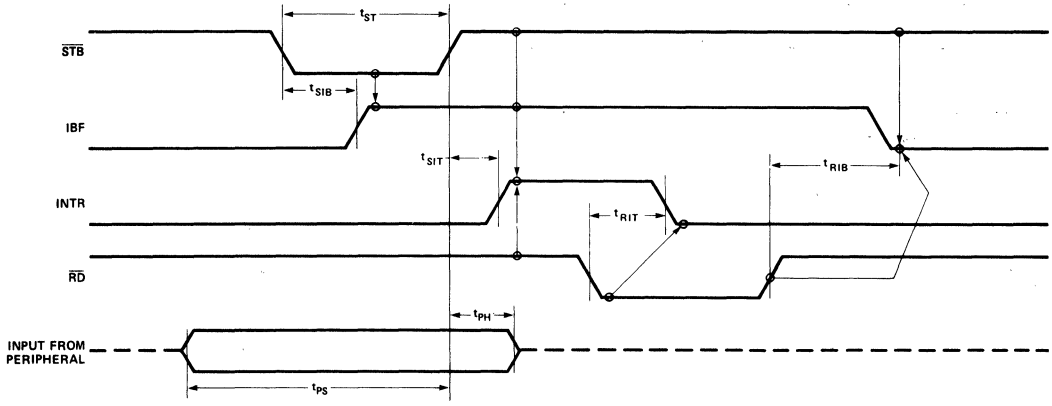


Figure 28. MODE 1 (Strobed Inut)

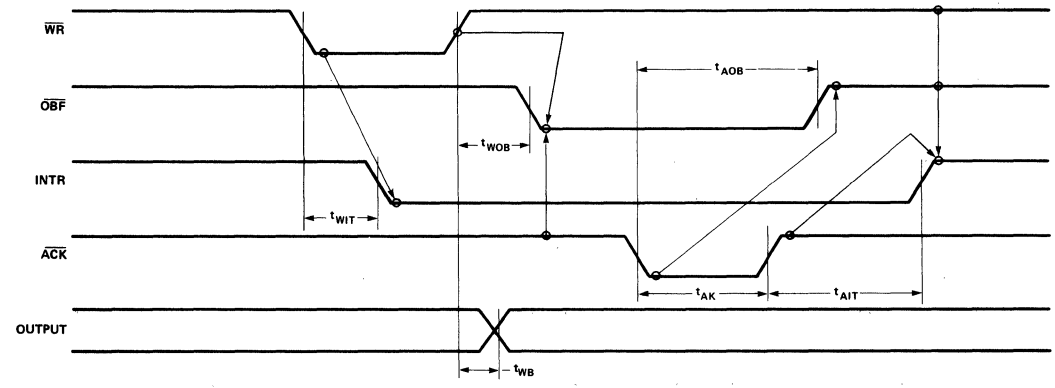


Figure 29. MODE 1 (Strobed Output)

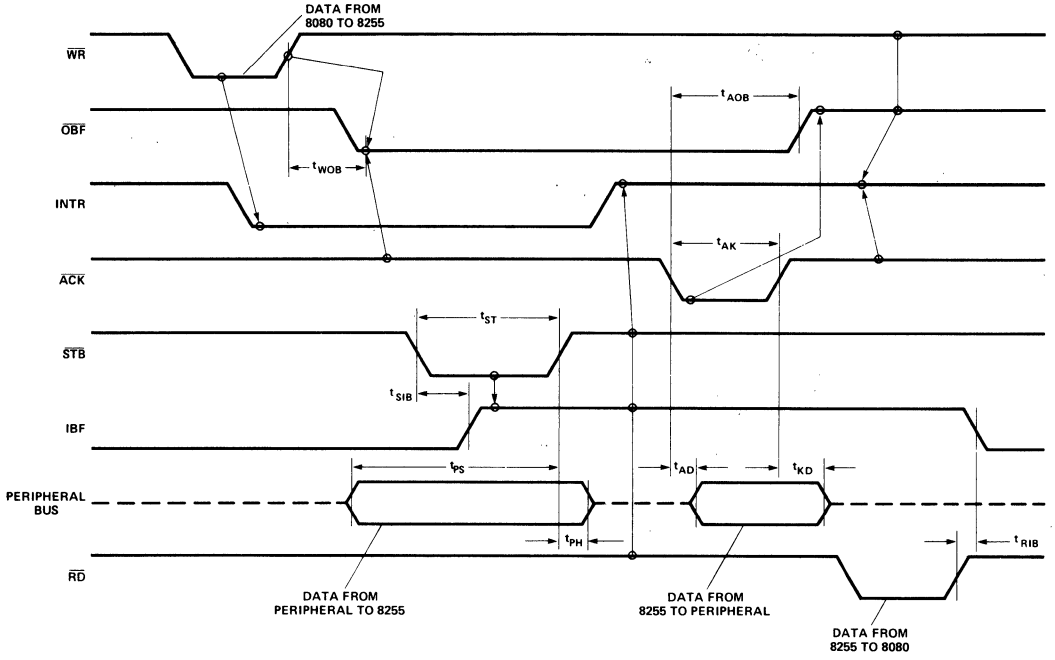


Figure 30. MODE 2 (Bidirectional)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

8259A
PROGRAMMABLE INTERRUPT CONTROLLER

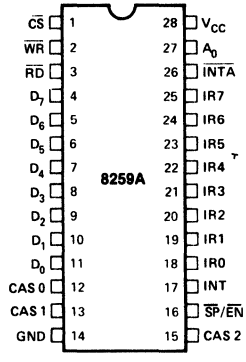
- **MCS-86™ Compatible**
- **MCS-80/85™ Compatible**
- **Eight-Level Priority Controller**
- **Expandable to 64 Levels**
- **Programmable Interrupt Modes**
- **Individual Request Mask Capability**
- **Single +5V Supply (No Clocks)**
- **28-Pin Dual-In-Line Package**

The Intel® 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel® 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

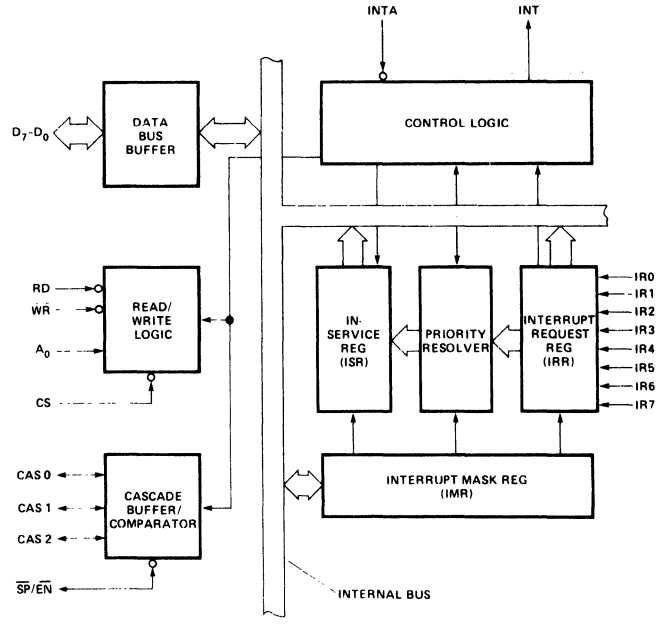
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI-DIRECTIONAL)
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A_0	COMMAND SELECT ADDRESS
$\overline{CS}$	CHIP SELECT
CAS2-CAS0	CASCADE LINES
$\overline{SP}/\overline{EN}$	SLAVE PROGRAM INPUT/ENABLE
INT	INTERRUPT OUTPUT
$\overline{INTA}$	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

BLOCK DIAGRAM



INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

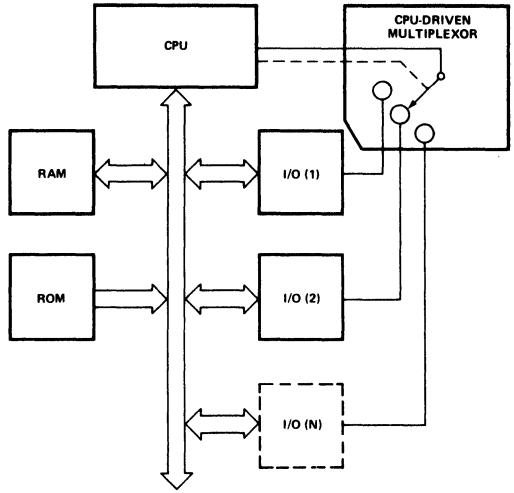
The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

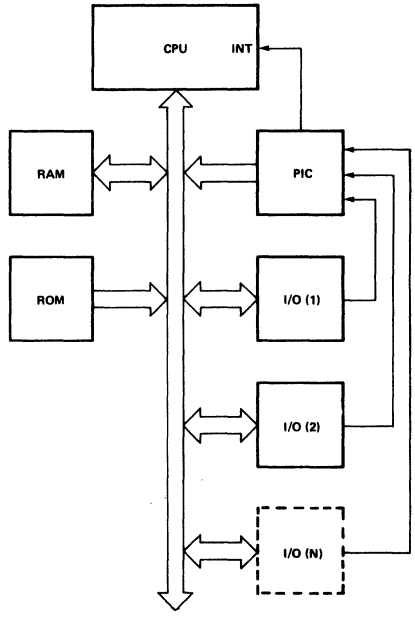
8259A BASIC FUNCTIONAL DESCRIPTION

GENERAL

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.



Polled Method



Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

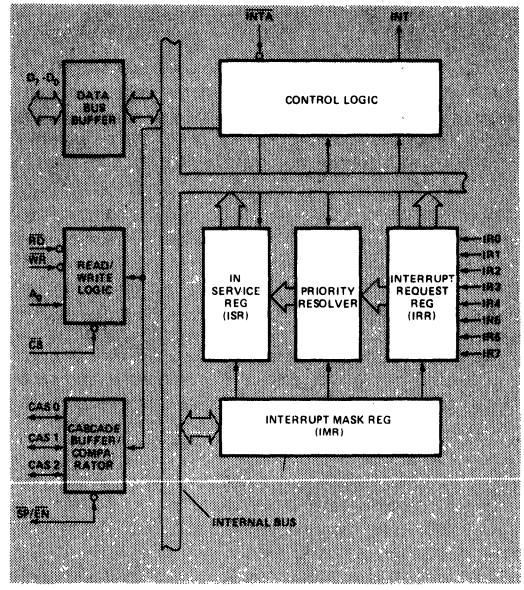
A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

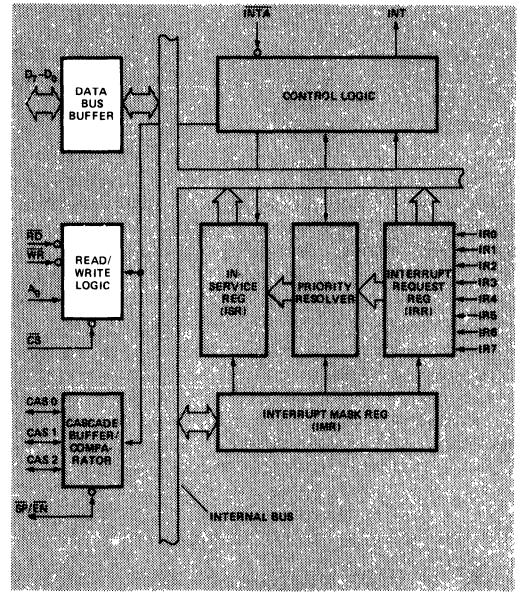
A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

$\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.



8259A Block Diagram



8259A Block Diagram

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259A".)

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

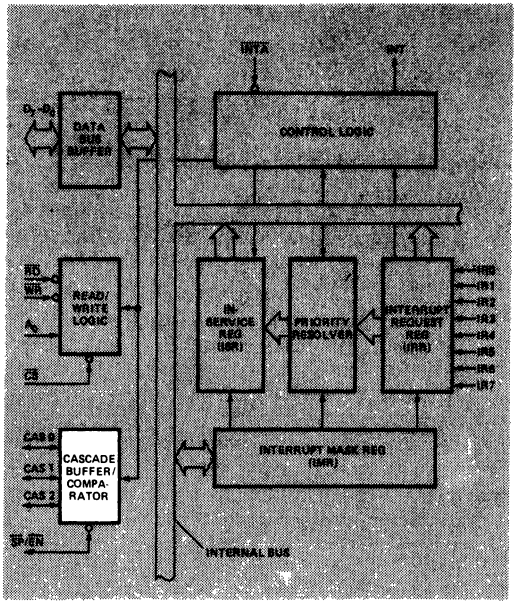
The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

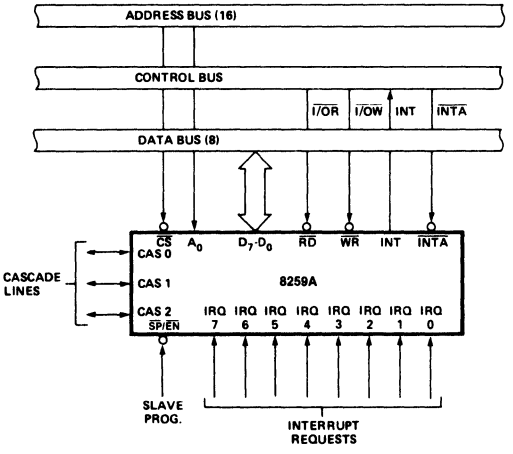
The events occurring in an MCS-86 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The MCS-86 CPU will initiate a second $\overline{INTA}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.



8259A Block Diagram



8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS

MCS-80/85 SYSTEM

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5-A_7 are programmed, while A_0-A_4 are automatically inserted by the 8259A. When Interval = 8 only A_6 and A_7 are programmed, while A_0-A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8 – A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

MCS-86 SYSTEM

MCS-86 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80/85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resoiution and as a master it issues the interrupt code on the cascade lines at the end of the $\overline{INTA}$ pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in MCS-86 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5-A_{11} are unused in MCS-86 mode):

Content of Interrupt Vector Byte for MCS-86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	A15	A14	A13	A12	A11	1	1	1
IR6	A15	A14	A13	A12	A11	1	1	0
IR5	A15	A14	A13	A12	A11	1	0	1
IR4	A15	A14	A13	A12	A11	1	0	0
IR3	A15	A14	A13	A12	A11	0	1	1
IR2	A15	A14	A13	A12	A11	0	1	0
IR1	A15	A14	A13	A12	A11	0	0	1
IR0	A15	A14	A13	A12	A11	0	0	0

PROGRAMMING THE 8259A

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses. This sequence is described in Figure 1.

2. *Operation Command Words (OCWs):* These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.

INITIALIZATION

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The Interrupt Mask Register is cleared.

b. IR 7 input is assigned priority 7.

c. The slave mode address is set to 7.

d. Special Mask Mode is cleared and Status Read is set to IRR.

e. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80/85 system, non SFNM).

***Note:** Master/Slave in ICW4 is only used in the buffered mode.

A_0	D_4	D_3	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level → DATA BUS (Note 1)
1			0	1	0	IMR → DATA BUS
						OUTPUT OPERATION (WRITE)
0	0	0	1	0	0	DATA BUS → OCW2
0	0	1	1	0	0	DATA BUS → OCW3
0	1	X	1	0	0	DATA BUS → ICW1
1	X	X	1	0	0	DATA BUS → OCW1, ICW2, ICW3, ICW4 (Note 2)
						DISABLE FUNCTION
X	X	X	1	1	0	DATA BUS — 3-STATE (NO OPERATION)
X	X	X	X	X	1	DATA BUS — 3-STATE (NO OPERATION)

Notes: 1. Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.
2. On-chip sequencer logic queues these commands into proper sequence.

8259A Basic Operation

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_5-A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0-A_{15}). When the routine interval is 4, A_0-A_4 are automatically inserted by the 8259A, while A_5-A_{15} are programmed externally. When the routine interval is 8, A_0-A_5 are automatically inserted by the 8259A, while A_6-A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an MCS-86 system A_{15}-A_{11} are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A_{10}-A_5 are ignored and ADI (Address interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for MCS-86 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for MCS-86) are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for MCS-80/85 system operation, μPM = 1 sets the 8259A for MCS-86 system operation.

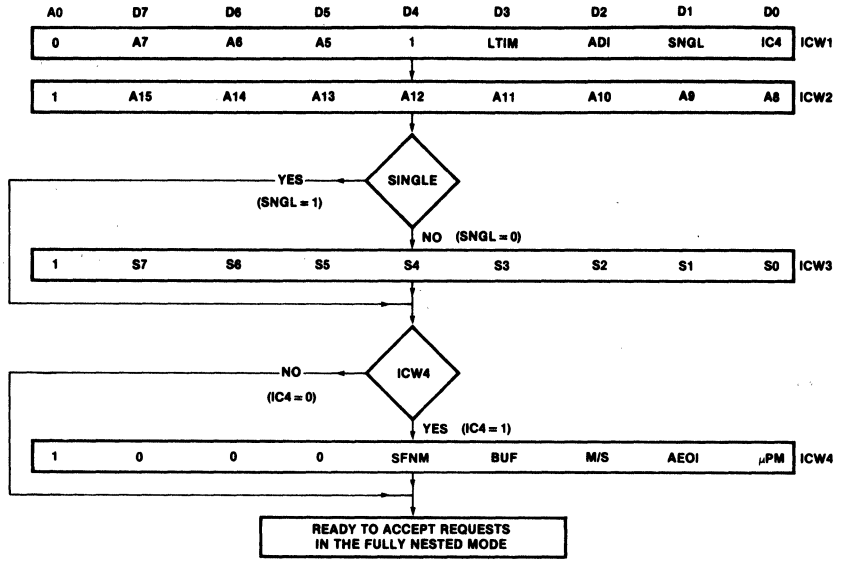
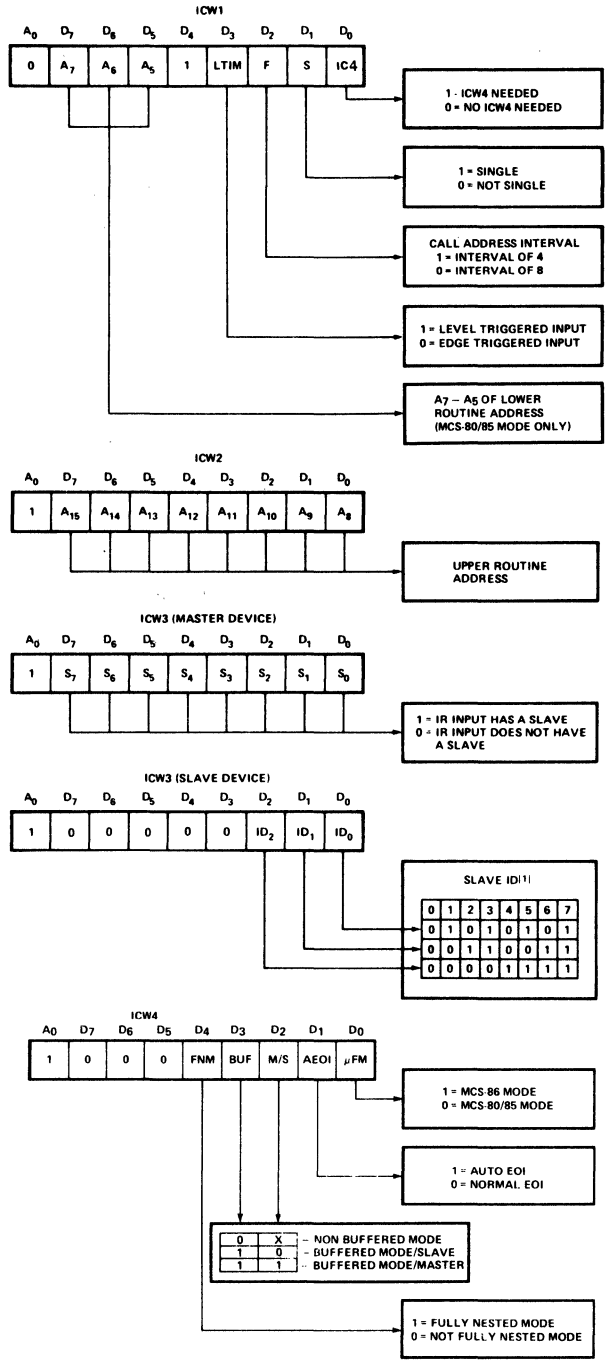


Figure 1. Initialization Sequence



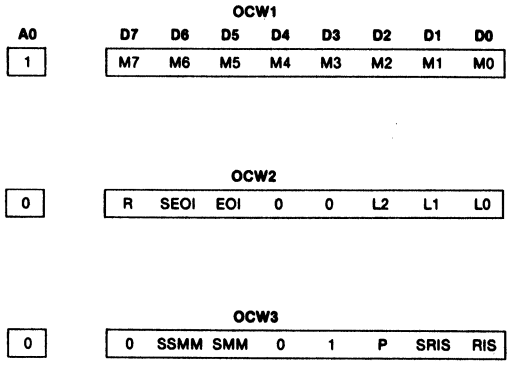
NOTE 1: SLAVED ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.
NOTE 2: X INDICATED "DON'T CARE".

Initialization Command Word Format

OPERATION COMMAND WORDS (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)



OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). $M_7 - M_0$ represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

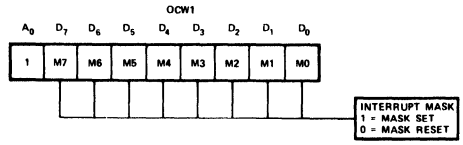
R, SEOI, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

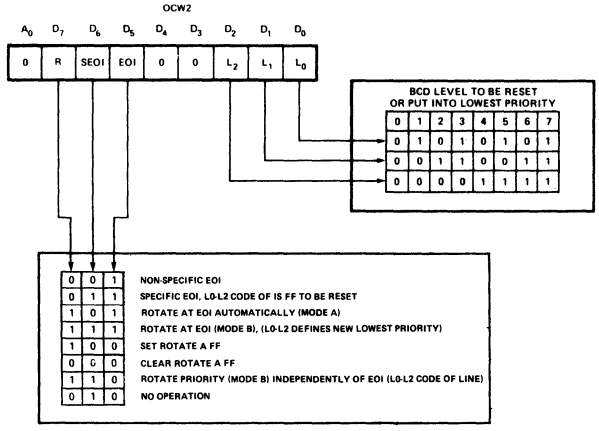
L_2, L_1, L_0 — These bits determine the interrupt level acted upon when the SEOI bit is active.

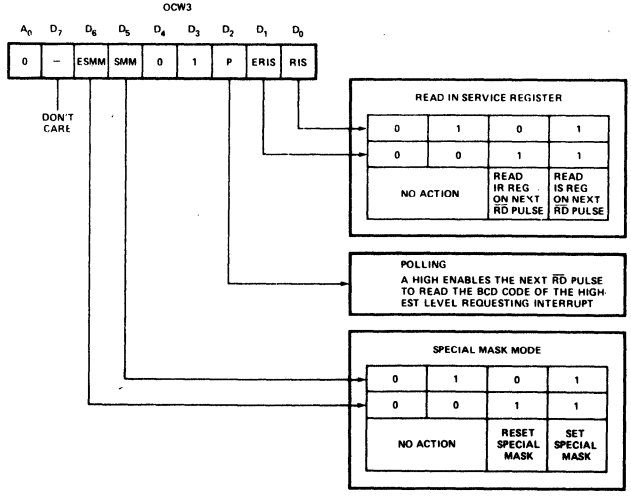
OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.







Operation Command Word Format

INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

BUFFERED MODE

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

POLL

In this mode the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by programmer initiative using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD} = 0$, $\overline{CS} = 0$) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0-W2: Binary code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{INTA}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice, once for the master and once for the corresponding slave if slaves are in use.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the nested mode the highest IS level was necessarily the last level acknowledged and serviced.

However, when a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt (SEOI) must be issued which includes as part of the command the IS level to be reset. EOI is issued whenever EOI = 1, in OCW2, where L0-L2 is the binary level of the IS bit to be reset. Note that although the Rotate command can be issued together with an EOI where EOI = 1, it is not necessarily tied to it.

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

AUTOMATIC END OF INTERRUPT (AEOI) MODE

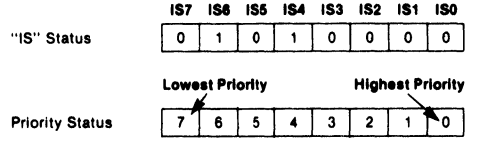
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in MCS-86). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

To achieve automatic rotation (Rotate Mode A) within AEOI, there is a special rotate flip-flop. It is set by OCW2 with R = 1, SEOI = 0, EOI = 0, and cleared with R = 0, SEOI = 0, EOI = 0.

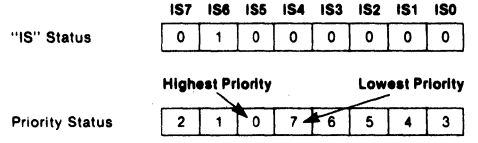
ROTATING PRIORITY MODE A (AUTOMATIC ROTATION) FOR EQUAL PRIORITY DEVICES

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



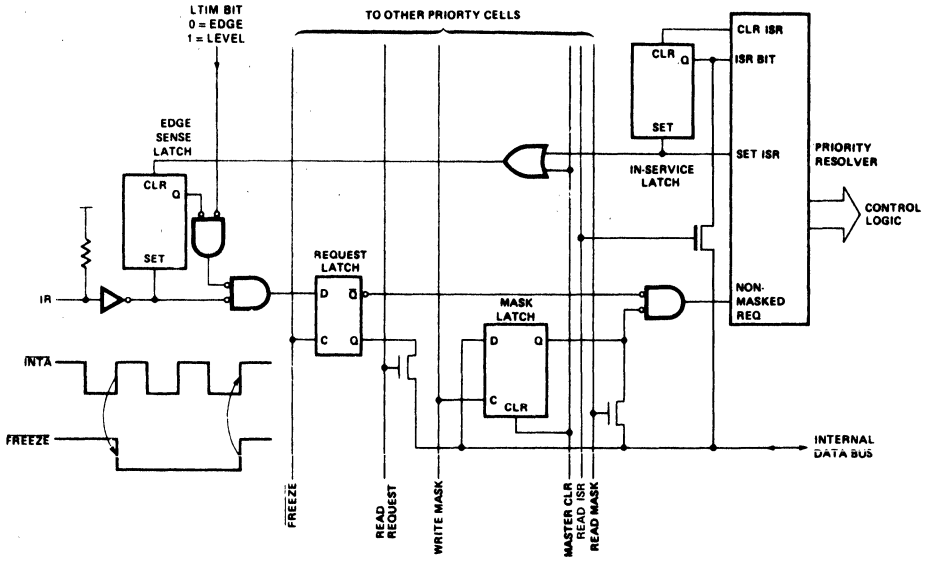
The Rotate command mode A is issued in OCW2 where: R = 1, EOI = 1, SEOI = 0. Internal status is updated by an End of Interrupt (EOI or AEOI) command. If R = 1, EOI = 0, SEOI = 0, a "Rotate-A" flip-flop is set. This is useful in AEOI, and described under Automatic End of Interrupt.

ROTATING PRIORITY MODE B (ROTATION BY SOFTWARE)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Rotate command is issued in OCW2 where: R = 1, SEOI = 1; L0-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command or independently.



NOTES

1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/ IS ACTIVE DURING $\overline{INTA}$/ AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	Di	Di	FOLLOW
0	X	Qn-1	HOLD

Priority Cell — Simplified Logic Diagram

LEVEL TRIGGERED MODE

This mode is programmed using bit 3 in ICW1.

If LTIM ='1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The above figure shows a conceptual circuit to give the reader an understanding of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

READING THE 8259A STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read by issuing a suitable OCW3 and reading with $\overline{RD}$.

Interrupt Mask Register: 8-bit register whose content specifies the interrupt request lines being masked. acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the $\overline{RD}$ pulse, a $\overline{WR}$ pulse is issued with OCW3 (ERIS = 1, RIS = 0.)

The ISR can be read in a similar mode when ERIS = 1, RIS = 1 in the OCW3.

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and A0 = 1.

Polling overrides status read when P = 1, ERIS = 1 in OCW3.

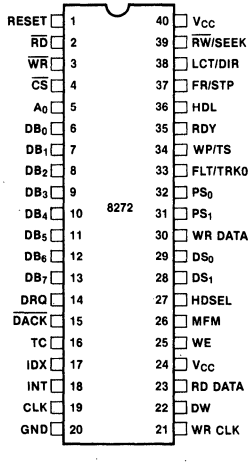
PRELIMINARY

8272
SINGLE/DOUBLE DENSITY FLOPPY DISK CONTROLLER

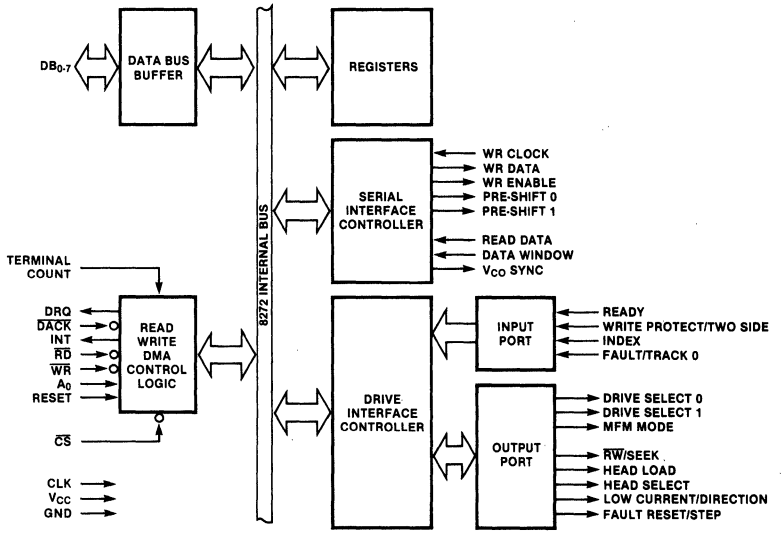
- **IBM Compatible in Both Single and Double Density Recording Formats**
- **Programmable Data Record Lengths: 128, 256, 512, or 1024 Bytes/Sector**
- **Multi-Sector and Multi-Track Transfer Capability**
- **Drive Up to 4 Floppy Disks**
- **Data Scan Capability — Will Scan a Single Sector or an Entire Cylinder's Worth of Data Fields, Comparing on a Byte by Byte Basis, Data in the Processor's Memory with Data Read from the Diskette**
- **Data Transfers in DMA or Non-DMA Mode**
- **Parallel Seek Operations on Up to Four Drives**
- **Compatible with Most Microprocessors Including 8080A, 8085A, 8086 and 8088**
- **Single-Phase 8 MHz Clock**
- **Single +5 Volt Power Supply**
- **Available in 40-Pin Plastic Dual-in-Line Package**

The 8272 is an LSI Floppy Disk Controller (FDC) Chip, which contains the circuitry and control functions for interfacing a processor to 4 Floppy Disk Drives. It is capable of supporting either IBM 3740 single density format (FM), or IBM System 34 Double Density format (MFM) including double sided recording. The 8272 provides control signals which simplify the design of an external phase locked loop, and write precompensation circuitry. The FDC simplifies and handles most of the burdens associated with implementing a Floppy Disk Drive Interface.

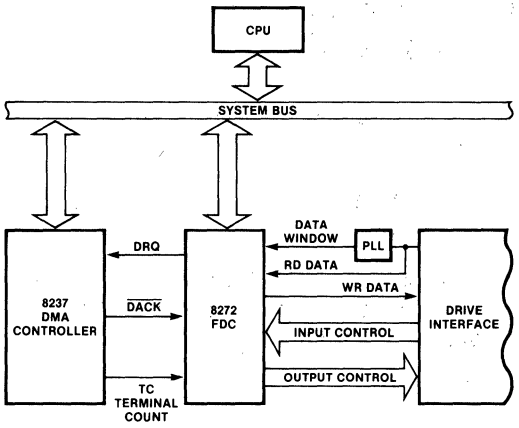
PIN CONFIGURATION



8272 INTERNAL BLOCK DIAGRAM



8272 SYSTEM BLOCK DIAGRAM



DESCRIPTION

Hand-shaking signals are provided in the 8272 which make DMA operation easy to incorporate with the aid of an external DMA Controller chip, such as the 8237. The FDC will operate in either DMA or Non-DMA mode. In the Non-DMA mode, the FDC generates interrupts to the processor for every transfer of a data byte between the CPU and the 8272. In the DMA mode, the processor need only load a command into the FDC and all data transfers occur under control of the 8272 and DMA controller.

There are 15 separate commands which the 8272 will execute. Each of these commands require multiple 8-bit bytes to fully specify the operation which the processor wishes the FDC to perform. The following commands are available.

- Read Data
- Read ID
- Read Deleted Data
- Read a Track
- Scan Equal
- Scan High or Equal
- Scan Low or Equal
- Specify
- Write Data
- Format a Track
- Write Deleted Data
- Seek
- Recalibrate (Restore to Track 0)
- Sense Interrupt Status
- Sense Drive Status

FEATURES

Address mark detection circuitry is internal to the FDC which simplifies the phase locked loop and read electronics. The track stepping rate, head load time, and head unload time may be programmed by the user. The 8272 offers many additional features such as multiple sector transfers in both read and write modes with a single command, and full IBM compatibility in both single (FM) and double density (MFM) modes.

8272 REGISTERS — CPU INTERFACE

The 8272 contains two registers which may be accessed by the main system processor; a Status Register and a Data Register. The 8-bit Main Status Register contains the status information of the FDC, and may be accessed at any time. The 8-bit Data Register (actually consists of several registers in a stack with only one register presented to the data bus at a time), stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to program or obtain the results after execution of a command. The Status Register may only be read and is used to facilitate the transfer of data between the processor and 8272.

The relationship between the Status/Data registers and the signals $\overline{RD}$, $\overline{WR}$, and A_0 is shown below.

A_0	$\overline{RD}$	$\overline{WR}$	FUNCTION
0	0	1	Read Main Status Register
0	1	0	Illegal
0	0	0	Illegal
1	0	0	Illegal
1	0	1	Read from Data Register
1	1	0	Write into Data Register

The bits in the Main Status Register are defined as follows:

BIT NUMBER	NAME	SYMBOL	DESCRIPTION
DB_0	FDD 0 Busy	D_0B	FDD number 0 is in the Seek mode.
DB_1	FDD 1 Busy	D_1B	FDD number 1 is in the Seek mode.
DB_2	FDD 2 Busy	D_2B	FDD number 2 is in the Seek mode.
DB_3	FDD 3 Busy	D_3B	FDD number 3 is in the Seek mode.
DB_4	FDC Busy	CB	A read or write command is in process.
DB_5	Non-DMA mode	NDM	The FDC is in the non-DMA mode. This bit is set only during the execution phase in non-DMA mode. Transition to "0" state indicates execution phase has ended.
DB_6	Data Input/Output	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO = "1" then transfer is from Data Register to the Processor. If DIO = "0", then transfer is from the Processor to Data Register.
DB_7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor.

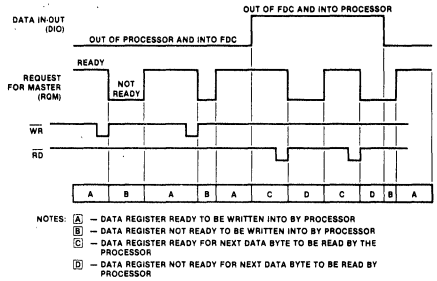
PIN DESCRIPTION

PIN NO.	PIN SYMBOL	I/O	CONNECTION TO	DESCRIPTION
1	RST	I	μP	Reset: Places FDC in idle state. Resets output lines to FDD to "0" (low)
2	$\overline{RD}$	I[1]	μP	Read: Control signal for transfer of data from FDC to Data Bus, when "0" (low)
3	$\overline{WR}$	I[1]	μP	Write: Control signal for transfer of data to FDC via Data Bus, when "0" (low)
4	$\overline{CS}$	I	μP	Chip Select: IC selected when "0" (low), allowing $\overline{RD}$ and $\overline{WR}$ to be enabled
5	A_0	I[1]	μP	Data/Status Reg Select: Selects Data Reg ($A_0 = 1$) or Status Reg ($A_0 = 0$) c o n t e n t be sent to Data Bus
6-13	DB_0-DB_7	I/O[1]	μP	Data Bus: Bidirectional 8-Bit Data Bus
14	DRQ	O	DMA	Data DMA Request: DMA Request is being made by FDC when DRQ "1"
15	$\overline{DACK}$	I	DMA	DMA Acknowledge: DMA cycle is active when "0" (low) and Controller is performing DMA transfer
16	TC	I	DMA	Terminal Count: Indicates the termination of a DMA transfer when "1" (high)
17	IDX	I	FDD	Index: Indicates the beginning of a disk track
18	INT	O	μP	Interrupt: Interrupt Request Generated by FDC
19	CLK	I		Clock: Single Phase 8 MHz Squarewave Clock
20	GND			Ground: D.C. Power Return

Note 1: Disabled when $\overline{CS} = 1$.

PIN NO.	PIN SYMBOL	I/O	CONNECTION TO	DESCRIPTION
40	V_{CC}			D.C. POWER +5V
39	$\overline{RW}$/SEEK	O	FDD	Read Write/SEEK: When "1" (high) Seek mode selected and when "0" (low) Read/Write mode selected
38	LCT/DIR	O	FDD	Low Current/Direction: Lowers Write current on inner tracks in Read/Write mode, determines direction head will step in Seek mode
37	FR/STP	O	FDD	Fault Reset/Step: Resets fault FF in FDD in Read/Write mode, provides step pulses to move head to another cylinder in Seek mode
36	HDL	O	FDD	Head Load: Command which causes read/write head in FDD to contact diskette
35	RDY	I	FDD	Ready: Indicates FDD is ready to send or receive data
34	WP/TS	I	FDD	Write Protect/Two-Side: Senses Write Protect status in Read/Write mode, and Two Side Media in Seek mode
33	FLT/TRK0	I	FDD	Fault/Track 0: Senses FDD fault condition in Read/Write mode and Track 0 condition in Seek mode.
31,32	PS_1,PS_0	O	FDD	Precompensation (pre-shift): Write precompensation status during MFM mode. Determines early, late, and normal times.
30	WR DATA	O	FDD	Write Data: Serial clock and data bits to FDD
28,29	DS_1,DS_0	O	FDD	Drive Select: Selects FDD unit
27	HDSEL	O	FDD	Head Select: Head 1 selected when "1" (high) Head 0 selected when "0" (low)
26	MFM	O	PLL	MFM Mode: MFM mode when "1", FM mode when "0"
25	WE	O	FDD	Write Enable: Enables write data into FDD
24	VCO	O	PLL	VCO Sync: Inhibits VCO in PLL when "0" (low), enables VCO when "1"
23	RD DATA	I	FDD	Read Data: Read data from FDD, containing clock and data bits
22	DW	I	PLL	Data Window: Generated by PLL, and used to sample data from FDD
21	WR CLK	I		Write Clock: Write data rate to FDD FM = 500 kHz, MFM = 1 MHz, with a pulse width of 250 ns for both FM and MFM

The DIO and ROM bits in the Status Register indicate when Data is ready and in which direction data will be transferred on the Data Bus.



STATUS REGISTER TIMING

The 8272 is capable of executing 15 different commands. Each command is initiated by a multi-byte transfer from the processor, and the result after execution of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the 8272 and the processor, it is convenient to consider each command as consisting of three phases:

Command Phase: The FDC receives all information required to perform a particular operation from the processor.

Execution Phase: The FDC performs the operation it was instructed to do.

Result Phase: After completion of the operation, status and other housekeeping information are made available to the processor.

During Command or Result Phases the Main Status Register (described earlier) must be read by the processor before each byte of information is written into or read from the Data Register. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the 8272. Many of the commands require multiple bytes, and as a result the Main Status Register must be read prior to each byte transfer to the 8272. On the other hand, during the Result Phase, D6 and D7 in the Main Status Register must both be 1's (D6 = 1 and D7 = 1) before reading each byte from the Data Register. Note, this reading of the Main Status Register before each byte transfer to the 8272 is required in only the Command and Result Phases, and NOT during the Execution Phase.

During the Execution Phase, the Main Status Register need not be read. If the 8272 is in the Non-DMA Mode, then the receipt of each data byte (if 8272 is reading data from FDD) is indicated by an Interrupt signal on pin 18 (INT = 1). The generation of a Read signal ($\overline{RD} = 0$) will reset the Interrupt as well as output the Data onto the Data Bus. For example, if the processor cannot handle Interrupts fast enough (every 13 μs for MFM mode) then it may poll the Main Status Register and then bit D7 (RQM) functions just like the Interrupt signal. If a Write Command is in process then the $\overline{WR}$ signal performs the reset to the Interrupt signal.

If the 8272 is in the DMA Mode, no Interrupts are generated during the Execution Phase. The 8272 generates DRQ's (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both a $\overline{DACK} = 0$ (DMA Acknowledge) and a $\overline{RD} = 0$ (Read signal). When the DMA Acknowledge signal goes low ($\overline{DACK} = 0$) then the DMA Request is reset (DRQ = 0). If a Write Command has been programmed then a $\overline{WR}$ signal will appear instead of $\overline{RD}$. After the Execution Phase has been completed (Terminal Count has occurred) then an Interrupt will occur (INT = 1). This signifies the beginning of the Result Phase. When the first byte of data is read during the Result Phase, the Interrupt is automatically reset (INT = 0).

It is important to note that during the Result Phase all bytes shown in the Command Table must be read. The Read Data Command, for example, has seven bytes of data in the Result Phase. All seven bytes must be read in order to successfully complete the Read Data Command. The 8272 will not accept a new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result Phase.

The 8272 contains five Status Registers. The Main Status Register mentioned above may be read by the processor at any time. The other four Status Registers (ST0, ST1, ST2, and ST3) are only available during the Result Phase, and may be read only after successfully completing a command. The particular command which has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the 8272 to form the Command Phase, and are read out of the 8272 in the Result Phase, must occur in the order shown in the Command Table. That is, the Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result Phases are allowed. After the last byte of data in the Command Phase is sent to the 8272 the Execution Phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result Phase, the command is automatically ended and the 8272 is ready for a new command. A command may be aborted by simply sending a Terminal Count signal to pin 16 (TC = 1). This is a convenient means of ensuring that the processor may always get the 8272's attention even if the disk system hangs up in an abnormal manner.

POLLING FEATURE OF THE 8272

After the Specify command has been sent to the 8272, the Drive Select Lines DS0 and DS1 will automatically go into a polling mode. In between commands (and between step pulses in the SEEK command) the 8272 polls all four FDDs looking for a change in the Ready line from any of the drives. If the Ready line changes state (usually due to a door opening or closing) then the 8272 will generate an interrupt. When Status Register 0 (ST0) is read (after Sense Interrupt Status is issued), Not Ready (NR) will be indicated. The polling of the Ready line by the 8272 occurs continuously between instructions, thus notifying the processor which drives are on or off line.

TABLE 1. 8272 COMMAND SET

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
Execution										Data transfer between the FDD and main-system
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after command
	R				N					execution
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EC 1					
	W				GPL					
	W				DTL					
Execution										Data transfer between the FDD and main-system
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after Command
	R				N					execution
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
Execution										Data transfer between the main-system and FDD
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after Command
	R				N					execution
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
Execution										Data transfer between the FDD and main-system
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after Command
	R				N					execution

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ A TRACK										
Command	W	0	MFM	SK	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EOT					
	W				GPL					
	W				DTL					
Execution										Data transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EOT
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after Command
	R				N					execution
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					during Execution
	R				N					Phase
FORMAT A TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				N					Bytes/Sector
	W				SC					Sectors/Track
	W				GPL					Gap 3
	W				D					Filter Byte
Execution										FDC formats an entire cylinder
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					In this case, the ID
	R				R					information has no
	R				N					meaning
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information
	W				H					prior to Command
	W				R					execution
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data compared between the FDD and main-system
Result	R				ST 0					Status information
	R				ST 1					after Command
	R				ST 2					execution
	R				C					
	R				H					Sector ID information
	R				R					after Command
	R				N					execution

Note: 1. Symbols used in this table are described at the end of this section.

2. $A_0 = 1$ for all operations.

3. X = Don't care, usually made to equal binary 0.

TABLE 1. COMMAND SET (Continued)

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information prior Command execution
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data compared between the FDD and main-system
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				C					Sector ID information after Command execution
	R				H					
	R				R					
	R				N					
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				C					Sector ID information prior Command execution
	W				H					
	W				R					
	W				N					
	W				EOT					
	W				GPL					
	W				STP					
Execution										Data compared between the FDD and main-system
Result	R				ST 0					Status information after Command execution
	R				ST 1					
	R				ST 2					
	R				C					Sector ID information after Command execution
	R				H					
	R				R					
	R				N					

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head retracted to Track 0
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R				ST 0					Status information at the end of each seek operation about the FDC
	R				PCN					
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SPT ◄ ► HUT								
	W	HLT ► ND								
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R				ST 3					Status information about FDD
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W				NCN					
Execution										Head is positioned over proper Cylinder on Diskette
INVALID										
Command	W				Invalid Codes					Invalid Command Codes (NoOp — FDC goes into Standby State)
Result	R				ST 0					ST 0 = 80 (16)

TABLE 2. COMMAND MNEMONICS

SYMBOL	NAME	DESCRIPTION
A_0	Address Line 0	A_0 controls selection of Main Status Register ($A_0 = 0$) or Data Register ($A_0 = 1$).
C	Cylinder Number	C stands for the current selected Cylinder track number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a Sector.
D_7-D_0	Data Bus	8-bit Data Bus where D_7 is the most significant bit, and D_0 is the least significant bit.
DS0, DS1	Drive Select	DS stands for a selected drive number 0 or 1.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the Sector.
EOT	End of Track	EOT stands for the final Sector number of a Cylinder.
GPL	Gap Length	GPL stands for the length of Gap 3 (spacing between Sectors excluding VCO Sync Field).
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HDS	Head Select	HDS stands for a selected head number 0 or 1 (H = HDS in all command words).
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MFM	FM or MFM Mode	If MF is low, FM mode is selected and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be performed (a cylinder under both HD0 and HD1 will be read or written).
N	Number	N stands for the number of data bytes written in a Sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R stands for the Sector number, which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (1 to 16 ms in 1 ms increments). Stepping Rate applies to all drives (F = 1 ms, E = 2 ms, etc.).
ST 0 ST 1 ST 2 ST 3	Status 0 Status 1 Status 2 Status 3	ST 0-3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by $A_0 = 0$). ST 0-3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2, then alternate sectors are read and compared.

COMMAND DESCRIPTIONS

During the Command Phase, the Main Status Register must be polled by the CPU before each byte is written into the Data Register. The DIO (DB6) and RQM (DB7) bits in the Main Status Register must be in the "0" and "1" states respectively, before each byte of the command may be written into the 8272. The beginning of the execution phase for any of these commands will cause DIO and RQM to switch to "1" and "0" states respectively.

READ DATA

A set of nine (9) byte words are required to place the FDC into the Read Data Mode. After the Read Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC outputs data (from the data field) byte-by-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous read function is called a "Multi-Sector Read Operation." The Read Data Command may be terminated by the receipt of a Terminal Count signal. Upon receipt of this signal, the FDC stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then at the end of the sector terminate the Read Data Command.

The amount of data which can be handled with a single command to the FDC depends upon MT (multi-track), MFM (MFM/FM), and N (Number of Bytes/Sector). Table 3 below shows the Transfer Capacity.

TABLE 3. TRANSFER CAPACITY

Multi-Track MT	MFM/FM MFM	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128) (26) = 3,328	26 at Side 0
0	1	01	(256) (26) = 8,656	or 26 at Side 1
1	0	00	(128) (52) = 6,656	26 at Side 1
1	1	01	(256) (52) = 13,312	
0	0	01	(256) (15) = 3,840	15 at Side 0
0	1	02	(512) (15) = 7,680	or 15 at Side 1
1	0	01	(256) (30) = 7,680	15 at Side 1
1	1	02	(512) (30) = 15,360	
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512) (16) = 8,192	8 at Side 1
1	1	03	(1024) (16) = 16,384	

The "multi-track" function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 0, Side 0 and completing at Sector L, Side 1 (Sector L = last sector on the side). Note, this function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, then DTL defines the data length which the FDC must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector, is not sent to the Data Bus. The FDC reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to 0FFH.

At the completion of the Read Data Command, the head is not unloaded until after Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

After reading the ID and Data Fields in each sector, the FDC checks the CRC bytes. If a read error is detected (incorrect CRC in ID field), the FDC sets the DE (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC also sets the DD (Data Error in Data Field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

If the FDC reads a Deleted Data Address Mark off the diskette, and the SK bit (bit D5 in the first Command Word) is not set (SK = 0), then the FDC sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector.

During disk data transfers between the FDC and the processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, and every 13 μs in the MFM Mode, or the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID Information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 4 shows the values for C, H, R, and N, when the processor terminates the Command.

TABLE 4. ID INFORMATION WHEN PROCESSOR TERMINATES COMMAND

MT	EOT	Final Sector Transferred to Processor	ID Information at Result Phase			
			C	H	R	N
0	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	C+1	NC	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	NC	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				
1	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	NC	LSB	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	LSB	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				

Notes: 1. NC (No Change): The same value as the one at the beginning of command execution.

2. LSB (Least Significant Bit): The least significant bit of H is complemented.

WRITE DATA

A set of nine (9) bytes are required to set the FDC into the Write Data mode. After the Write Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Fields. When the current sector number ("R"), stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC takes data from the processor byte-by-byte via the data bus, and outputs it to the FDD.

After writing data into the current sector, the Sector Number stored in "R" is incremented by one, and the next data field is written into. The FDC continues this "Multi-Sector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written then the remainder of the data field is filled with 00 (zeros).

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a read error (incorrect CRC) in one of the ID Fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high), and terminates the Write Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

The Write Command operates in much the same manner as the Read Command. The following items are the same; refer to the Read Data Command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag

- Head Unload Time Interval
- ID Information when the processor terminates command (see Table 2)
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC must occur every 31 μs in the FM mode, and every 15 μs in the MFM mode. If the time interval between data transfers is longer than this then the FDC sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Write Data Command.

WRITE DELETED DATA

This command is the same as the Write Data Command except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

READ DELETED DATA

This command is the same as the Read Data Command except that when the FDC detects a Data Address Mark at the beginning of a Data Field (and SK = 0 (low)), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, then the FDC skips the sector with the Data Address Mark and reads the next sector.

READ A TRACK

This command is similar to READ DATA Command except that the entire data field is read continuously from each of the sectors of a track. Immediately after encountering the INDEX HOLE, the FDC starts reading all data fields on the track as continuous blocks of data. If the FDC finds an error in the ID or DATA CRC check bytes, it continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multi-track or skip operations are not allowed with this command.

This command terminates when EOT number of sectors have been read. If the FDC does not find an ID Address Mark on the diskette after it encounters the INDEX HOLE for the second time, then it sets the MA (missing address mark) flag in Status Register 1 to a 1 (high), and terminates the command. (Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.)

READ ID

The READ ID Command is used to give the present position of the recording head. The FDC stores the values from the first ID Field it is able to read. If no proper ID Address Mark is found on the diskette, before the INDEX HOLE is encountered for the second time then the MA (Missing Address Mark) flag in Status Register 1 is set to a 1 (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a 1 (high) and the command is terminated.

FORMAT A TRACK

The Format Command allows an entire track to be formatted. After the INDEX HOLE is detected, Data is written on the Diskette: Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) Format are recorded. The particular format which will be written is controlled by the values programmed into N (number of bytes/sector), SC (sectors/cylinder), GPL (Gap Length), and D (Data Pattern) which are supplied by the processor during the Command Phase. The Data Field is filled with the Byte of data stored in D. The ID Field for each sector is supplied by the processor; that is, four data requests per sector are made by the FDC for C (Cylinder Number), H (Head Number), R (Sector Number) and N (Number of Bytes/Sector). This allows the diskette to be formatted with nonsequential sector numbers, if desired.

After formatting each sector, the processor must send new values for C, H, R, and N to the 8272 for each sector on the track. The contents of the R register is incremented by one after each sector is formatted, thus, the R register contains a value of R + 1 when it is read during the Result Phase. This incrementing and formatting continues for the whole track until the FDC encounters the INDEX HOLE for the second time, whereupon it terminates the command.

If a FAULT signal is received from the FDD at the end of a write operation, then the FDC sets the EC flag of Status Register 0 to a 1 (high), and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also the loss of a READY signal at the beginning of a command execution phase causes command termination.

Table 5 shows the relationship between N, SC, and GPL for various sector sizes:

TABLE 5. SECTOR SIZE RELATIONSHIPS

FORMAT	SECTOR SIZE	N	SC	GPL[1]	GPL[2]	REMARKS
FM Mode	128 bytes/Sector	00	$1A_{(16)}$	$07_{(16)}$	$1B_{(16)}$	IBM Diskette 1
	256	01	$0F_{(16)}$	$0E_{(16)}$	$2A_{(16)}$	IBM Diskette 2
	512	02	08	$1B_{(16)}$	$3A_{(16)}$	
FM Mode	1024 bytes/Sector	03	04	—	—	
	2048	04	02	—	—	
	4096	05	01	—	—	
MFM Mode	256	01	$1A_{(16)}$	$0E_{(16)}$	$36_{(16)}$	IBM Diskette 2D
	512	02	$0F_{(16)}$	$1B_{(16)}$	$54_{(16)}$	
	1024	03	08	$35_{(16)}$	$74_{(16)}$	IBM Diskette 2D
	2048	04	04	—	—	
	4096	05	02	—	—	
	8192	06	01	—	—	

Note: 1. Suggested values of GPL in Read or Write Commands to avoid splice point between data field and ID field of contiguous sections.

2. Suggested values of GPL in format command.

SCAN COMMANDS

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system (Processor in NON-DMA mode, and DMA Controller in DMA mode). The FDC compares the data on a byte-by-byte basis, and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur; the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 6 shows the status of bits SH and SN under various conditions of SCAN.

TABLE 6. SCAN STATUS CODES

COMMAND	STATUS REGISTER 2		COMMENTS
	BIT 2 = SN	BIT 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$

If the FDC encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors STP = 01, or alternate sectors STP = 02 sectors are read) or the MT (Multi-Track) are programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21; the following will happen. Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 27 μs (FM Mode) or 13 μs (MFM Mode). If an Overrun occurs the FDC terminates the command.

SEEK

The read/write head within the FDD is moved from cylinder to cylinder under control of the Seek Command. The FDC compares the PCN (Present Cylinder Number) which is the current head position with the NCN (New Cylinder Number), and performs the following operation if there is a difference:

PCN < NCN: Direction signal to FDD set to a 1 (high), and Step Pulses are issued. (Step In.)

PCN > NCN: Direction signal to FDD set to a 0 (low), and Step Pulses are issued. (Step Out.)

The rate at which Step Pulses are issued is controlled by SRT (Stepping Rate Time) in the SPECIFY Command. After each Step Pulse is issued NCN is compared against PCN, and when NCN = PCN, then the SE (Seek End) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

During the Command Phase of the Seek operation the FDC is in the FDC BUSY state, but during the Execution Phase it is in the NON BUSY state. While the FDC is in the NON BUSY state, another Seek Command may be issued, and in this manner parallel seek operations may be done on up to 4 Drives at once.

If an FDD is in a NOT READY state at the beginning of the command execution phase or during the seek operation, then the NR (NOT READY) flag is set in Status Register 0 to a 1 (high), and the command is terminated.

RECALIBRATE

This command causes the read/write head within the FDD to retract to the Track 0 position. The FDC clears the contents of the PCN counter, and checks the status of the Track 0 signal from the FDD. As long as the Track 0 signal is low, the Direction signal remains 1 (high) and Step Pulses are issued. When the Track 0 signal goes high, the SE (SEEK END) flag in Status Register 0 is set to a 1 (high) and the command is terminated. If the Track 0 signal is still low after 77 Step Pulses have been issued, the FDC sets the SE (SEEK END) and EC (EQUIPMENT CHECK) flags of Status Register 0 to both 1s (highs), and terminates the command.

The ability to overlap RECALIBRATE Commands to multiple FDDs, and the loss of the READY signal, as described in the SEEK Command, also applies to the RECALIBRATE Command.

SENSE INTERRUPT STATUS

An Interrupt signal is generated by the FDC for one of the following reasons:

1. Upon entering the Result Phase of:
   a. Read Data Command
   b. Read a Track Command
   c. Read ID Command
   d. Read Deleted Data Command
   e. Write Data Command
   f. Format a Cylinder Command
   g. Write Deleted Data Command
   h. Scan Commands
2. Ready Line of FDD changes state
3. End of Seek or Recalibrate Command
4. During Execution Phase in the NON-DMA Mode

Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. However, interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt Status Command. This command when issued resets the interrupt signal and via bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt.

TABLE 7. SEEK, INTERRUPT CODES

SEEK END BIT 5	INTERRUPT CODE		CAUSE
	BIT 6	BIT 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

Neither the Seek or Recalibrate Command have a Result Phase. Therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them and to provide verification of the head position (PCN).

SPECIFY

The Specify Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the Execution Phase of one of the Read/Write Commands to the head unload state. This timer is programmable from 16 to 240 ms in increments of 16 ms (01 = 16 ms, 02 = 32 ms .... OF = 240 ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1 ms, E = 2 ms, D = 3 ms, etc.). The HLT (Head Load Time) defines the time between when the Head Load signal goes high and when the Read/Write operation starts. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2 ms, 02 = 4 ms, 03 = 6 ms .... FE = 254 ms).

The time intervals mentioned above are a direct function of the clock (CLK on pin 19). Times indicated above are for an 8 MHz clock, if the clock was reduced to 4 MHz (mini-floppy application) then all time intervals are increased by a factor of 2.

The choice of DMA or NON-DMA operation is made by the ND (NON-DMA) bit. When this bit is high (ND = 1) the NON-DMA mode is selected, and when ND = 0 the DMA mode is selected.

SENSE DRIVE STATUS

This command may be used by the processor whenever it wishes to obtain the status of the FDDs. Status Register 3 contains the Drive Status information.

INVALID

If an invalid command is sent to the FDC (a command not defined above), then the FDC will terminate the command. No interrupt is generated by the 8272 during this condition. Bit 6 and bit 7 (DIO and RQM) in the Main Status Register are both high ("1") indicating to the processor that the 8272 is in the Result Phase and the contents of Status Register 0 (STO) must be read. When the processor reads Status Register 0 it will find a 80H indicating an invalid command was received.

A Sense Interrupt Status Command must be sent after a Seek or Recalibrate interrupt, otherwise the FDC will consider the next command to be an Invalid Command.

In some applications the user may wish to use this command as a No-Op command, to place the FDC in a standby or no operation state.

TABLE 8. STATUS REGISTERS

BIT NO.	BIT NAME	BIT SYMBOL	DESCRIPTION
STATUS REGISTER 0			
D_7	Interrupt Code	IC	$D_7 = 0$ and $D_6 = 0$ Normal Termination of Command, (NT). Command was completed and properly executed.
D_6			$D_7 = 0$ and $D_6 = 1$ Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed.
			$D_7 = 1$ and $D_6 = 0$ Invalid Command issue, (IC). Command which was issued was never started.
			$D_7 = 1$ and $D_6 = 1$ Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK Command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault Signal is received from the FDD, or if the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US 1	These flags are used to indicate a
D_0	Unit Select 0	US 0	Drive Unit Number at Interrupt
STATUS REGISTER 1			
D_7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set.
D_6			Not used. This bit is always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set.
D_4	Over Run	OR	If the FDC is not serviced by the main-systems during data transfers, within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN Command, if the FDC cannot find the Sector specified in the IDR Register, this flag is set.
			During executing the READ ID Command, if the FDC cannot read the ID field without an error, then this flag is set.
			During the execution of the READ A Cylinder Command, if the starting sector cannot be found, then this flag is set.
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A Cylinder Command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set.
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set.
STATUS REGISTER 2			
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During executing the READ DATA or SCAN Command, if the FDC encounters a Sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related with the ND bit, and when the contents of C on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution, the SCAN Command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During executing the SCAN Command, if the FDC cannot find a Sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related with the ND bit, and when the content of C on the medium is different from that stored in the IDR and the content of C is FF, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.
STATUS REGISTER 3			
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RDY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Side	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of Side Select signal to the FDD.
D_1	Unit Select 1	US 1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US 0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature ..... −10°C to +70°C
Storage Temperature ..... −40°C to +125°C
All Output Voltages ..... −0.5 to +7 Volts
All Input Voltages ..... −0.5 to +7 Volts
Supply Voltage V_{CC} ..... −0.5 to +7 Volts
Power Dissipation ..... 1 Watt

*T_A = 25°C

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

DC CHARACTERISTICS

T_A = 0°C to +70°C; V_{CC} = +5V ± 5%

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		MIN	MAX		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{IL}	(CLK & WR CLK)	−0.5	0.65	V	
V_{IH}	(CLK & WR CLK)	2.4	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4	V_{CC}	V	I_{OH} = −200 μA
I_{CC}	V_{CC} Supply Current		150	mA	
I_{IL}	Input Load Current (All Input Pins)		10 −10	μA μA	V_{IN} = V_{CC} V_{IN} = 0V
I_{LOH}	High Level Output Leakage Current		10	μA	V_{OUT} = V_{CC}
I_{LOL}	Low Level Output Leakage Current		−10	μA	V_{OUT} = +0.45V

CAPACITANCE

T_A = 25°C; f_c = 1 MHz; V_{CC} = 0V

SYMBOL	PARAMETER	LIMITS		UNIT	TEST CONDITIONS
		MIN	MAX		
$C_{IN(\Phi)}$	Clock Input Capacitance		20	pF	All Pins Except Pin Under Test Tied to AC Ground
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		20	pF	

A.C. CHARACTERISTICS

$t_A = 0°C$ to $70°C$, $V_{CC} = +5.0V \pm 5\%$

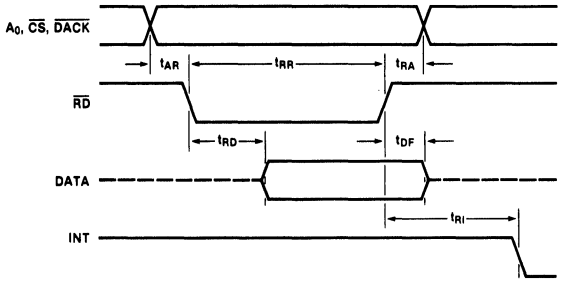
SYMBOL	PARAMETER	TYP[1]	MIN	MAX	UNIT	TEST CONDITIONS
t_{CY}	Clock Period		125		ns	
t_{CH}	Clock High Period		40		ns	
t_{RST}	Reset Width		14		t_{CY}	
Read Cycle						
t_{AR}	Select Setup to $\overline{RD}\downarrow$		0		ns	
t_{RA}	Select Hold from $\overline{RD}\uparrow$		0		ns	
t_{RR}	$\overline{RD}$ Pulse Width		250		ns	
t_{RD}	Data Delay from $\overline{RD}\downarrow$			200	ns	$C_L = 100$ pF
t_{DF}	Output Float Delay		20	100	ns	$C_L = 100$ pF
Write Cycle						
t_{AW}	Select Setup to $\overline{WR}\downarrow$		0		ns	
t_{WA}	Select Hold from $\overline{WR}\uparrow$		0		ns	
t_{WW}	$\overline{WR}$ Pulse Width		250		ns	
t_{DW}	Data Setup to $\overline{WR}\uparrow$		150		ns	
t_{WD}	Data Hold from $\overline{WR}\uparrow$		5		ns	
Interrupts						
t_{RI}	INT Delay from $\overline{RD}\uparrow$			500	ns	
t_{WI}	INT Delay from $\overline{WR}\uparrow$			500	ns	
DMA						
t_{RQCY}	DRQ Cycle Period		13		µs	
t_{AKRQ}	$\overline{DACK}\downarrow$ to DRQ↓			200	ns	
t_{RQR}	DRQ↑ to $\overline{RD}\downarrow$		800		ns	8 MHz clock
t_{RQW}	DRQ↑ to $\overline{WR}\downarrow$		250		ns	8 MHz clock
t_{RQRW}	DRQ↑ to $\overline{RD}\uparrow$ or $\overline{WR}\uparrow$			12	µs	8 MHz clock
FDD Interface						
t_{WCY}	WCK Cycle Time	2 or 4			µs	MFM = 0 Note 2
		1 or 2				MFM = 1
t_{WCH}	WCK High Time	250	80	350	ns	
t_{CP}	Pre-Shift Delay from WCK↑		20	100	ns	
t_{CD}	WDA Delay from WCK↑		20	100	ns	
t_{WDD}	Write Data Width		$t_{WCH} - 50$		ns	
t_{WE}	WE↑ to WCK↑ or WE↓ to WCK↓ Delay		20	100	ns	
t_{WWCY}	Window Cycle Time	2			µs	MFM = 0
		1				MFM = 1
t_{WRD}	Window Setup to RDD↑		15		ns	
t_{RDW}	Window Hold from RDD↓		15		ns	
t_{RDD}	RDD Active Time (HIGH)		40		ns	
FDD SEEK/ DIRECTION/ STEP						
t_{US}	$US_{0,1}$ Setup to $\overline{RW}$/SEEK↑		12		µs	8 MHz clock
t_{SU}	$US_{0,1}$ Hold from $\overline{RW}$/SEEK↓		15		µs	
t_{SD}	$\overline{RW}$/SEEK Setup to LCT/DIR		7		µs	
t_{DS}	$\overline{RW}$/SEEK Hold from LCT/DIR		30		µs	
t_{DST}	LCT/DIR Setup to FR/STEP↑		1		µs	
t_{STD}	LCT/DIR Hold from FR/STEP↓		24		µs	
t_{STU}	$DS_{0,1}$ Hold from FR/Step↓		5		µs	
t_{STP}	STEP Active Time (High)	5			µs	
t_{SC}	STEP Cycle Time		33		µs	Note 3
t_{FR}	FAULT RESET Active Time (High)		8	10	µs	
t_{IDX}	INDEX Pulse Width	625			µs	
t_{TC}	Terminal Count Width		1		t_{CY}	

NOTES:

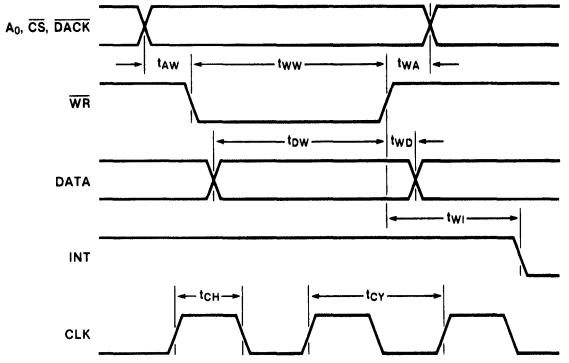
1. Typical values for $T_A = 25°C$ and nominal supply voltage.
2. The former values are used for standard floppy and the latter values are used for mini-floppies.
3. $t_{SC} = 33\mu s$ min. is for different drive units. In the case of same unit, t_{SC} can be ranged from 1 ms to 16 ms with 8 MHz clock period, and 2 ms to 32 ms with 4 MHz clock, under software control.

TIMING WAVEFORMS

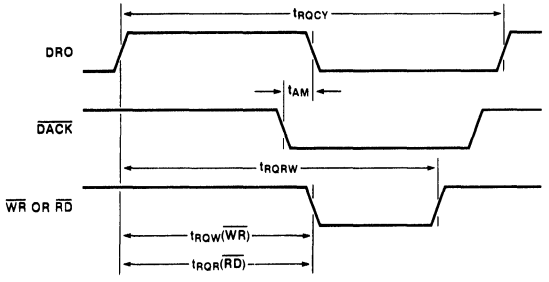
PROCESSOR READ OPERATION



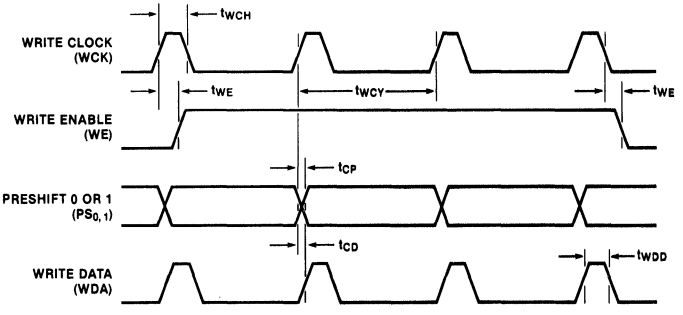
PROCESSOR WRITE OPERATION



DMA OPERATION

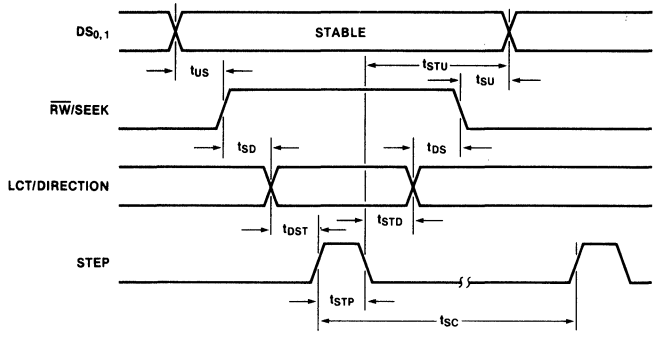


FDD READ OPERATION

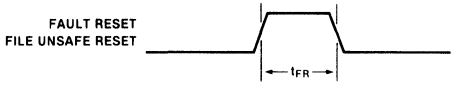


	PRESHIFT 0	PRESHIFT 1
NORMAL	0	0
LATE	0	1
EARLY	1	0
INVALID	1	1

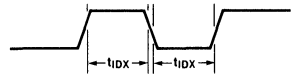
SEEK OPERATION



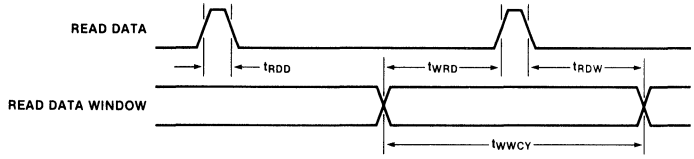
FLT RESET



INDEX

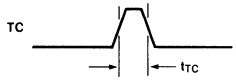


FDD READ OPERATION

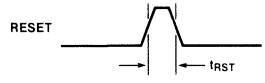


NOTE: EITHER POLARITY DATA WINDOW IS VALID.

TERMINAL COUNT

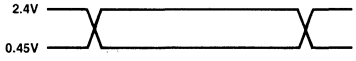


RESET



A.C. TIMING MEASUREMENT CONDITIONS

INPUT WAVEFORM



MEASUREMENT POINT
INPUT: 2.0V ∞ 0.8V
OUTPUT: 2.0V ∞ 0.8V

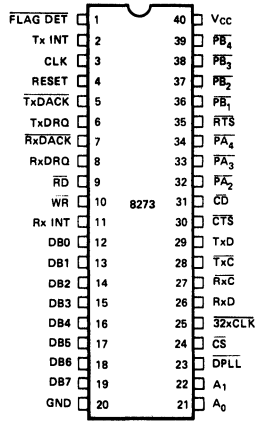
PRELIMINARY

8273, 8273-4, 8273-8
PROGRAMMABLE HDLC/SDLC PROTOCOL CONTROLLER

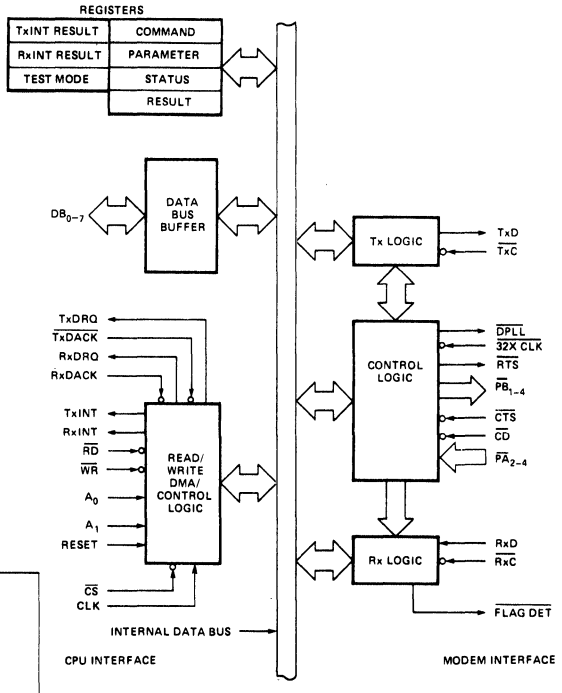
- **CCITT X.25 Compatible**
- **HDLC/SDLC Compatible**
- **Full Duplex, Half Duplex, or Loop SDLC Operation**
- **Up to 64K Baud Synchronous Transfers**
- **Automatic FCS (CRC) Generation and Checking**
- **Up to 9.6K Baud with On-Board Phase Locked Loop**
- **Programmable NRZI Encode/Decode**
- **Two User Programmable Modem Control Ports**
- **Digital Phase Locked Loop Clock Recovery**
- **Minimum CPU Overhead**
- **Fully Compatible with 8048/8080/8085/ 8088/8086 CPUs**
- **Single +5V Supply**

The Intel® 8273 Programmable HDLC/SDLC Protocol Controller is a dedicated device designed to support the ISO/ CCITT's HDLC and IBM's SDLC communication line protocols. It is fully compatible with Intel's new high performance microcomputer systems such as the MCS-88/86™. A frame level command set is achieved by a unique microprogrammed dual processor chip architecture. The processing capability supported by the 8273 relieves the system CPU of the low level real-time tasks normally associated with controllers.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

DB0–DB7	DATA BUS (8 BITS)	$\overline{CS}$	CHIP SELECT
$\overline{FLAG\ DET}$	FLAG DETECT	$\overline{32xCLK}$	32 TIMES CLOCK
TxINT	TRANSMITTER INTERRUPT	Rx D	RECEIVER DATA
CLK	CLOCK INPUT	$\overline{Rx\ C}$	RECEIVER CLOCK
$\overline{RESET}$	RESET	$\overline{Tx\ C}$	TRANSMITTER CLOCK
$\overline{Tx\ DACK}$	TRANSMITTER DMA ACKNOWLEDGE	Tx D	TRANSMITTER DATA
TxDRQ	TRANSMITTER DMA REQUEST	$\overline{CTS}$	CLEAR TO SEND
$\overline{RD}$	READ INPUT	$\overline{CD}$	CARRIER DETECT
$\overline{WR}$	WRITE INPUT	$\overline{PA_2–PA_4}$	GP INPUT PORTS
$\overline{Rx\ DACK}$	RECEIVER DMA ACKNOWLEDGE	$\overline{PB_1–PB_4}$	GP OUTPUT PORTS
Rx DRQ	RECEIVER DMA REQUEST	$\overline{RTS}$	REQUEST TO SEND
Rx INT	RECEIVER INTERRUPT	V_{CC}	+5 VOLT SUPPLY
$\overline{A0–A1}$	COMMAND REGISTER SELECT ADDRESS	GND	GROUND
$\overline{DPLL}$	DIGITAL PHASE LOCKED LOOP		

A BRIEF DESCRIPTION OF HDLC/SDLC PROTOCOLS

General

The High Level Data Link Control (HDLC) is a standard communication link protocol established by International Standards Organization (ISO). HDLC is the discipline used to implement ISO X.25 packet switching systems.

The Synchronous Data Link Control (SDLC) is an IBM communication link protocol used to implement the System Network Architecture (SNA). Both the protocols are bit oriented, code independent, and ideal for full duplex communication. Some common applications include terminal to terminal, terminal to CPU, CPU to CPU, satellite communication, packet switching and other high speed data links. In systems which require expensive cabling and interconnect hardware, any of the two protocols could be used to simplify interfacing (by going serial), thereby reducing interconnect hardware costs. Since both the protocols are speed independent, reducing interconnect hardware could become an important application.

Network

In both the HDLC and SDLC line protocols, according to a pre-assigned hierarchy, a PRIMARY (Control) STATION controls the overall network (data link) and issues commands to the SECONDARY (Slave) STATIONS. The latter comply with instructions and respond by sending appropriate RESPONSES. Whenever a transmitting station must end transmission prematurely it sends an ABORT character. Upon detecting an abort character, a receiving station ignores the transmission block called a FRAME. Time fill between frames can be accomplished by transmitting either continuous frame preambles called FLAGS or an abort character. A time fill within a frame is not permitted. Whenever a station receives a string of more that fifteen consecutive ones, the station goes into an IDLE state.

Frames

A single communication element is called a FRAME which can be used for both Link Control and data transfer purposes. The elements of a frame are the beginning eight bit FLAG (F) consisting of one zero, six ones, and a zero, an eight bit ADDRESS FIELD (A), an eight bit CONTROL FIELD (C), a variable (N-bit) INFORMATION FIELD (I), a sixteen bit FRAME CHECK SEQUENCE (FCS), and an eight bit end FLAG (F), having the same bit pattern as the beginning flag. In HDLC the Address (A) and Control (C) bytes are extendable. The HDLC and the SDLC use three types of frames; an Information Frame is used to transfer data, a Supervisory Frame is used for control purposes, and a Non-sequenced Frame is used for initialization and control of the secondary stations.

Frame Characteristics

An important characteristic of a frame is that its contents are made code transparent by use of a zero bit insertion and deletion technique. Thus, the user can adopt any format or code suitable for his system — it may even be a computer word length or a "memory dump". The frame is bit oriented that is, bits, not characters in each field, have specific meanings. The Frame Check Sequence (FCS) is an error detection scheme similar to the Cyclic Redundancy Checkword (CRC) widely used in magnetic disk storage devices. The Command and Response information frames contain sequence numbers in the control fields identifying the sent and received frames. The sequence numbers are used in Error Recovery Procedures (ERP) and as implicit acknowledgement of frame communication, enhancing the true full-duplex nature of the HDLC/SDLC protocols.

In contrast, BISYNC is basically half-duplex (two way alternate) because of necessity to transmit immediate acknowledgement frames. HDLC/SDLC therefore saves propagation delay times and have a potential of twice the throughput rate of BISYNC.

It is possible to use HDLC or SDLC over half duplex lines but there is a corresponding loss in throughput because both are primarily designed for full-duplex communication. As in any synchronous system, the bit rate is determined by the clock bits supplied by the modem, protocols themselves are speed independent.

A byproduct of the use of zero-bit insertion-deletion technique is the non-return-to-zero invert (NRZI) data transmission/reception compatibility. The latter allows HDLC/SDLC protocols to be used with asynchronous data communication hardware in which the clocks are derived from the NRZI encoded data.

References

IBM Synchronous Data Link Control General Information, IBM, GA 27-3093-1.

Standard Network Access Protocol Specification, DATAPAC, Trans-Canada Telephone System CCG111

Recommendation X.25, ISO/CCITT March 2, 1976.

IBM 3650 Retail Store System Loop Interface OEM Information, IBM, GA 27-3098-0

Guidebook to Data Communications, Training Manual, Hewlett-Packard 5955-1715

IBM Introduction to Teleprocessing, IBM, GC 20-8095-02

System Network Architecture, Technical Overview, IBM, GA 27-3102

System Network Architecture Format and Protocol, IBM GA 27-3112

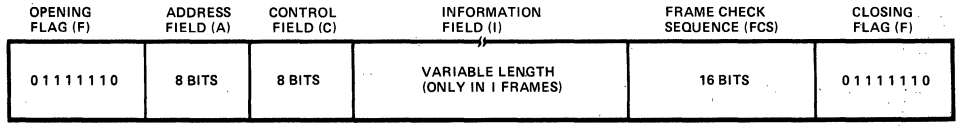


Figure 1. Frame Format

FUNCTIONAL DESCRIPTION

General

The Intel® 8273 HDLC/SDLC controller is a microcomputer peripheral device which supports the International Standards Organization (ISO) High Level Data Link Control (HDLC), and IBM Synchronous Data Link Control (SDLC) communications protocols. This controller minimizes CPU software by supporting a comprehensive frame-level instruction set and by hardware implementation of the low level tasks associated with frame assembly/disassembly and data integrity. The 8273 can be used in either synchronous or asynchronous applications. In asynchronous applications the data can be programmed to be encoded/decoded in NRZI code. The clock is derived from the NRZI data using a digital phase locked loop. The data transparency is achieved by using a zero-bit insertion/deletion technique. The frames are automatically checked for errors during reception by verifying the Frame Check Sequence (FCS); the FCS is automatically generated and appended before the final flag in transmit.

The 8273 recognizes and can generate flags (01111110), Abort, Idle, and GA (EOP) characters.

The 8273 can assume either a primary (control) or a secondary (slave) role. It can therefore be readily implemented in an SDLC loop configuration as typified by the IBM 3650 Retail Store System by programming the 8273 into a one-bit delay mode. In such a configuration, a two wire pair can be effectively used for data transfer between controllers and loop stations. The digital phase locked loop output pin can be used by the loop station without the presence of an accurate Tx clock.

Hardware Description

The 8273 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Pin Name (No.)	I/O	Description
V_{CC} (40)		+5V Supply
GND (20)		Ground
RESET (4)	I	A high signal on this pin will force the 8273 to an idle state. The 8273 will remain idle until a command is issued by the CPU. The modem interface output signals are forced high. Reset must be true for a minimum of 10 TCY.
$\overline{CS}$ (24)	I	The $\overline{RD}$ and $\overline{WR}$ inputs are enabled by the chip select input.
DB_7-DB_0 (19-12)	I/O	The Data Bus lines are bidirectional three-state lines which interface with the system Data Bus.
$\overline{WR}$ (10)	I	The Write signal is used to control the transfer of either a command or data from CPU to the 8273.
$\overline{RD}$ (9)	I	The Read signal is used to control the transfer of either a data byte or a status word from the 8273 to the CPU.
TxINT (2)	O	The Transmitter interrupt signal indicates that the transmitter logic requires service.
RxINT (11)	O	The Receiver interrupt signal indicates that the Receiver logic requires service.
TxDRQ (6)	O	Requests a transfer of data between memory and the 8273 for a transmit operation.
RxRDQ (8)	O	Requests a transfer of data between the 8273 and memory for a receive operation.
$\overline{TxDACK}$ (5)	I	The Transmitter DMA acknowledge signal notifies the 8273 that the TxDMA cycle has been granted.
$\overline{RxDACK}$ (7)	I	The Receiver DMA acknowledge signal notifies the 8273 that the RxDMA cycle has been granted.
A_1-A_0 (22-21)	I	These two lines are CPU Interface Register Select lines.
TxD (29)	O	This line transmits the serial data to the communication channel.
$\overline{TxC}$ (28)	I	The transmitter clock is used to synchronize the transmit data.
RxD (26)	I	This line receives serial data from the communication channel.
$\overline{RxC}$ (27)	I	The Receiver Clock is used to synchronize the receive data.
$\overline{32X\ CLK}$ (25)	I	The 32X clock is used to provide clock recovery when an asynchronous modem is used. In loop configuration the loop station can run without an accurate 1X clock by using the 32X CLK in conjunction with the DPLL output. (This pin must be grounded when not used).
$\overline{DPLL}$ (23)	O	Digital Phase Locked Loop output can be tied to RxC and/or TxC when 1X clock is not available. DPLL is used with 32X CLK.
$\overline{FLAG\ DET}$ (1)	O	Flag Detect signals that a flag (01111110) has been received by an active receiver.
$\overline{RTS}$ (35)	O	Request to Send signals that the 8273 is ready to transmit data.
$\overline{CTS}$ (30)	I	Clear to Send signals that the modem is ready to accept data from the 8273.
$\overline{CD}$ (31)	I	Carrier Detect signals that the line transmission has started and the 8273 may begin to sample data on RxD line.
$\overline{PA_{2-4}}$ (32-34)	I	General purpose input ports. The logic levels on these lines can be Read by the CPU through the Data Bus Buffer.
$\overline{PB_{1-4}}$ (36-39)	O	General purpose output ports. The CPU can write these output lines through Data Bus Buffer.
CLK (3)	I	A square wave TTL clock.

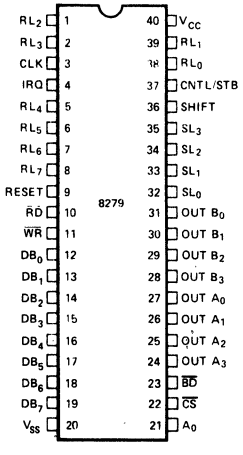
8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-Key Rollover with Contact Debounce
- Dual 8- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

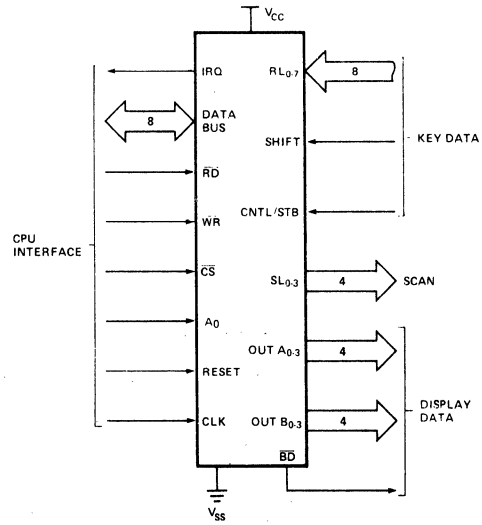
PIN CONFIGURATION



PIN NAMES

DB_{0-7}	I/O	DATA BUS (BI DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
$\overline{CS}$	I	CHIP SELECT
$\overline{RD}$	I	READ INPUT
$\overline{WR}$	I	WRITE INPUT
A_0	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL_{0-3}	O	SCAN LINES
RL_{0-7}	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
$OUT A_{0-3}$	O	DISPLAY (A) OUTPUTS
$OUT B_{0-3}$	O	DISPLAY (B) OUTPUTS
$\overline{BD}$	O	BLANK DISPLAY OUTPUT

LOGIC SYMBOL



AFN-00742A-01

HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

No. Of Pins	Designation	Function
8	DB_0-DB_7	Bi-directional data bus. All data and commands between the CPU and the 8279 are transmitted on these lines.
1	CLK	Clock from system used to generate internal timing.
1	RESET	A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display —left entry. 2) Encoded scan keyboard—2 key lockout. Along with this the program clock prescaler is set to 31.
1	$\overline{CS}$	Chip Select. A low on this pin enables the interface functions to receive or transmit.
1	A_0	Buffer Address. A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
2	$\overline{RD}$, $\overline{WR}$	Input/Output read and write. These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
1	IRQ	Interrupt Request. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
2	V_{SS}, V_{CC}	Ground and power supply pins.
4	SL_0-SL_3	Scan Lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).
8	RL_0-RL_7	Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
1	SHIFT	The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
1	CNTL/STB	For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
4 4	OUT A_0-OUT A_3 OUT B_0-OUT B_3	These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0-SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8 bit port.
1	$\overline{BD}$	Blank Display. This output is used to blank the display during digit switching or by a display blanking command.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 1.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR} \bullet \overline{CS}$ and output during $\overline{RD} \bullet \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with $A_0 = 1$ and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$.

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

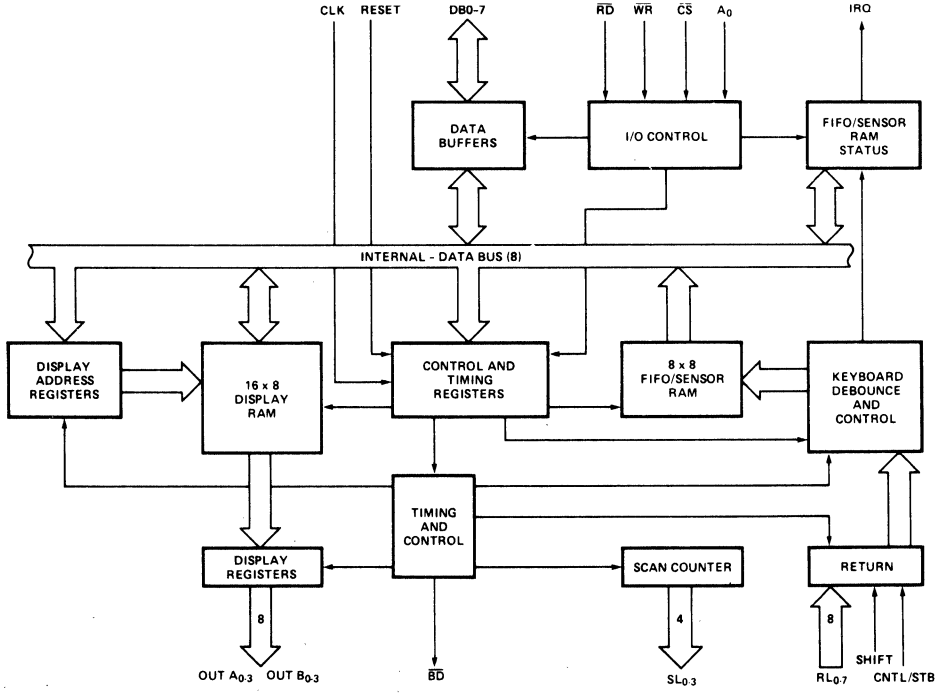
- Scanned Keyboard — with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- **Scanned Sensor Matrix — with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.**
- Strobed Input -- Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit ($B_0 = D_0$, $A_3 = D_7$).
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.



The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note than when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0	0	8 8-bit character display — Left entry
0	1	16 8-bit character display — Left entry*
1	0	8 8-bit character display — Right entry
1	1	16 8-bit character display — Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard — 2 Key Lockout*
0	0	1	Decoded Scan Keyboard — 2-Key Lockout
0	1	0	Encoded Scan Keyboard — N-Key Rollover
0	1	1	Decoded Scan Keyboard — N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

Program Clock

Code:	0	0	1	P	P	P	P	P

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, PPPPP should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Read FIFO/Sensor RAM

Code:	0	1	0	AI	X	A	A	A	X = Don't Care

The CPU sets up the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Key-

*Default after reset.

board Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0 = 0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set (AI = 1), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set (AI = 1), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read *or write* address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0 = 1$, all subsequent writes with $A_0 = 0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display or FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder, the CPU may write a digit to the Display RAM without affecting the other digit being displayed. It is important to note that bit B_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C_D C_D C_D	
0 X	All Zeros (X = Don't Care)
1 0	AB = Hex 20 (0010 0000)
1 1	All Ones

Enable clear display when = 1 (or by C_A = 1)

During the time the Display RAM is being cleared (~160 μs), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted ($C_F = 1$), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X

X = Don't care.

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode — if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard — Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with C_F = 1.

Sensor Matrix Mode

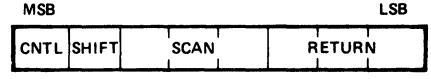
In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them.

The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End interrupt command if the Auto-Increment flag is set to one.

Note: Multiple changes in the matrix Addressed by (SL_{0-3} =0) may cause multiple interrupts. (SL_0=0 in the Decoded Mode). Reset may cause the 8279 to see multiple changes.

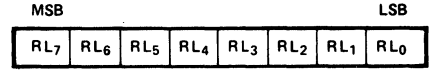
Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

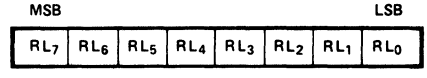


SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch postion maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.



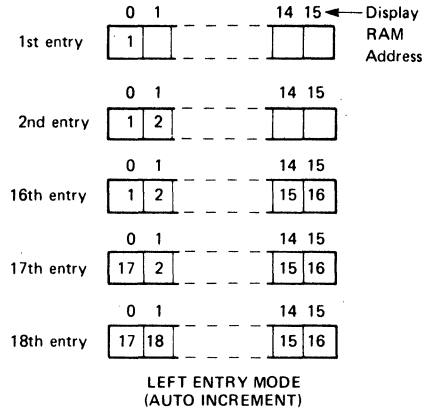
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.



Display

Left Entry

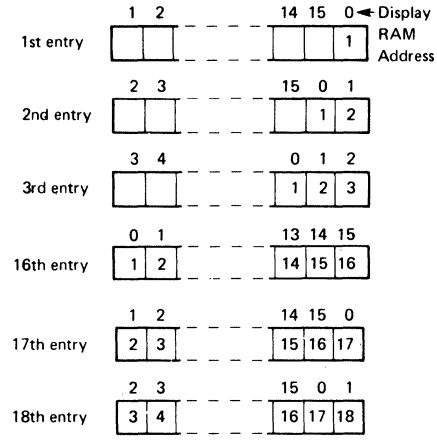
Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.



LEFT ENTRY MODE
(AUTO INCREMENT)

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.

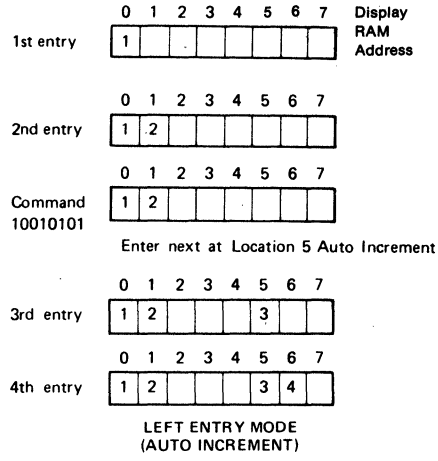


RIGHT ENTRY MODE
(AUTO INCREMENT)

Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

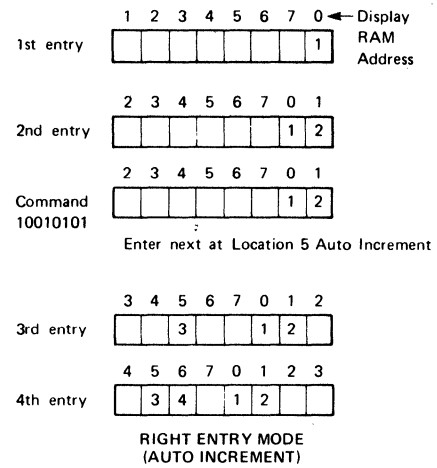
Auto Increment

In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:



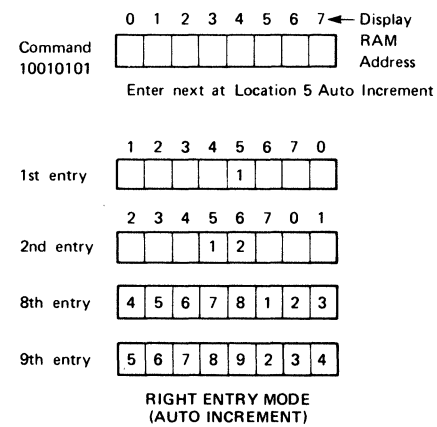
LEFT ENTRY MODE
(AUTO INCREMENT)

In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Starting at an arbitrary location operates as shown below:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Entry appears to be from the initial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display, the on duty-cycle is double what it would be for a 16 character display (e.g., 5.1 ms scan time for 8 characters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

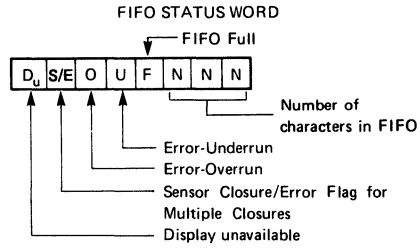
G. FIFO Status

FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.



APPLICATIONS

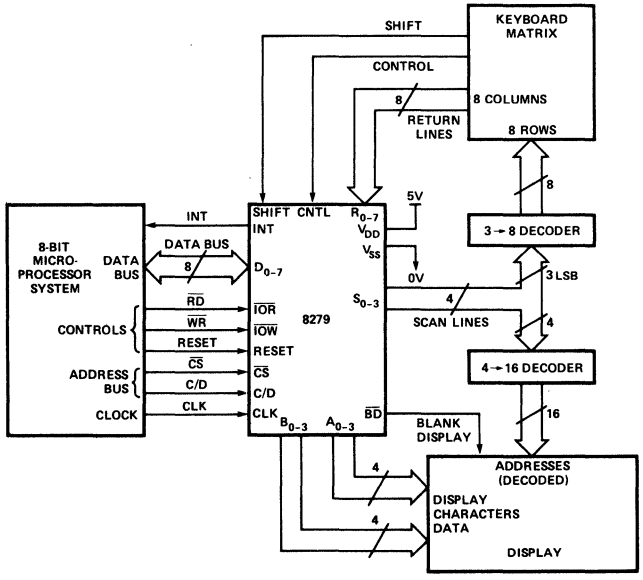


FIGURE 2. GENERAL BLOCK DIAGRAM

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature . . . . . 0°C to 70°C
Storage Temperature . . . . . -65°C to 125°C
Voltage on any Pin with
Respect to Ground . . . . . -0.5V to +7V
Power Dissipation . . . . . 1 Watt

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

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{SS} = 0V, Note 1

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL1}	Input Low Voltage for Shift Control and Return Lines	-0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	-0.5	0.8	V	
V_{IH1}	Input High Voltage for Shift, Control and Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	Note 2
V_{OH}	Output High Voltage on Interrupt Line	3.5		V	Note 3
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 -100	μA μA	$V_{IN} = V_{CC}$ $V_{IN} = 0V$
I_{IL2}	Input Leakage Current on All Others		±10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	μA	$V_{OUT} = V_{CC}$ to 0V
I_{CC}	Power Supply Current		120	mA	

Notes:
1. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%.
2. 8279, I_{OL} = 1.6mA; 8279-5, I_{OL} = 2.2mA.
3. 8279, I_{OH} = -100μA; 8279-5, I_{OH} = -400μA.

CAPACITANCE

SYMBOL	TEST	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{in}	Input Capacitance	5	10	pF	$V_{in}=V_{CC}$
C_{out}	Output Capacitance	10	20	pF	$V_{out}=V_{CC}$

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{SS} = 0V, (Note 1)

BUS PARAMETERS

READ CYCLE:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AR}	Address Stable Before $\overline{READ}$	50		0		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		0		ns
t_{RR}	$\overline{READ}$ Pulse Width	420		250		ns
t_{RD}[2]	Data Delay from $\overline{READ}$		300		150	ns
t_{AD}[2]	Address to Data Valid		450		250	ns
t_{DF}	$\overline{READ}$ to Data Floating	10	100	10	100	ns
t_{RCY}	Read Cycle Time	1		1		μs

WRITE CYCLE:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AW}	Address Stable Before $\overline{WRITE}$	50		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	20		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		250		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		150		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		0		ns

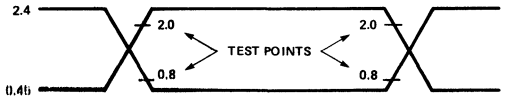
Notes:
1. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%.
2. 8279, C_L = 100pF; 8279-5, C_L = 150pF.

OTHER TIMINGS:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
$t_{\phi W}$	Clock Pulse Width	230		120		nsec
t_{CY}	Clock Period	500		320		nsec

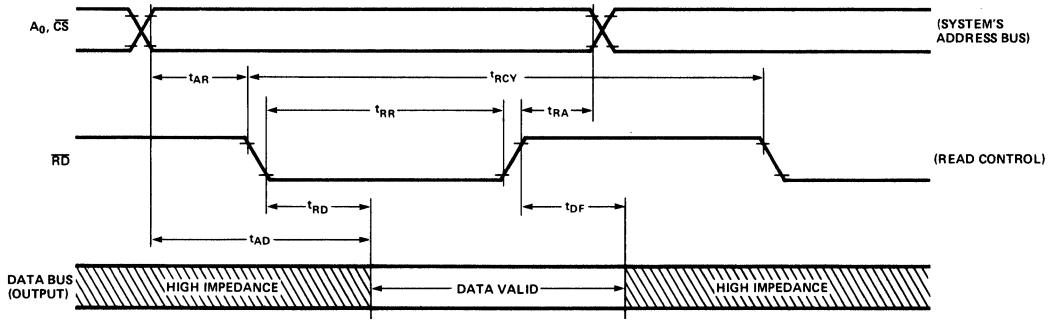
Keyboard Scan Time:	5.1 msec	Digit-on Time:	480 μsec
Keyboard Debounce Time:	10.3 msec	Blanking Time:	160 μsec
Key Scan Time:	80 μsec	Internal Clock Cycle:	10 μsec
Display Scan Time:	10.3 msec		

INPUT WAVEFORMS FOR A.C. TESTS:

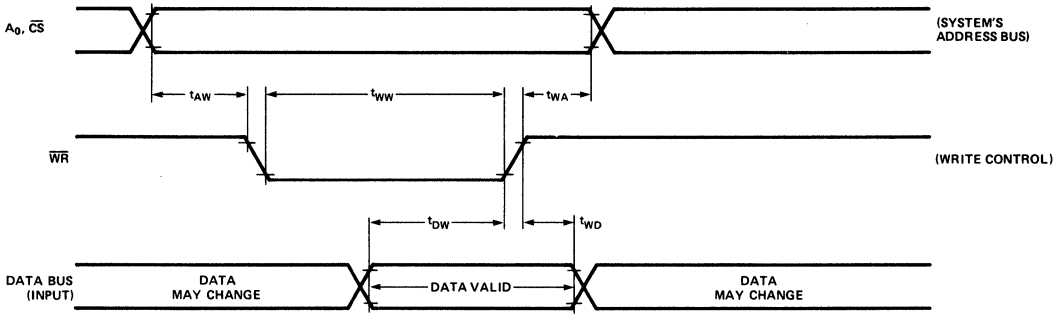


WAVEFORMS

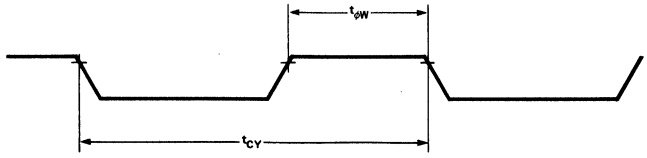
1. Read Operation



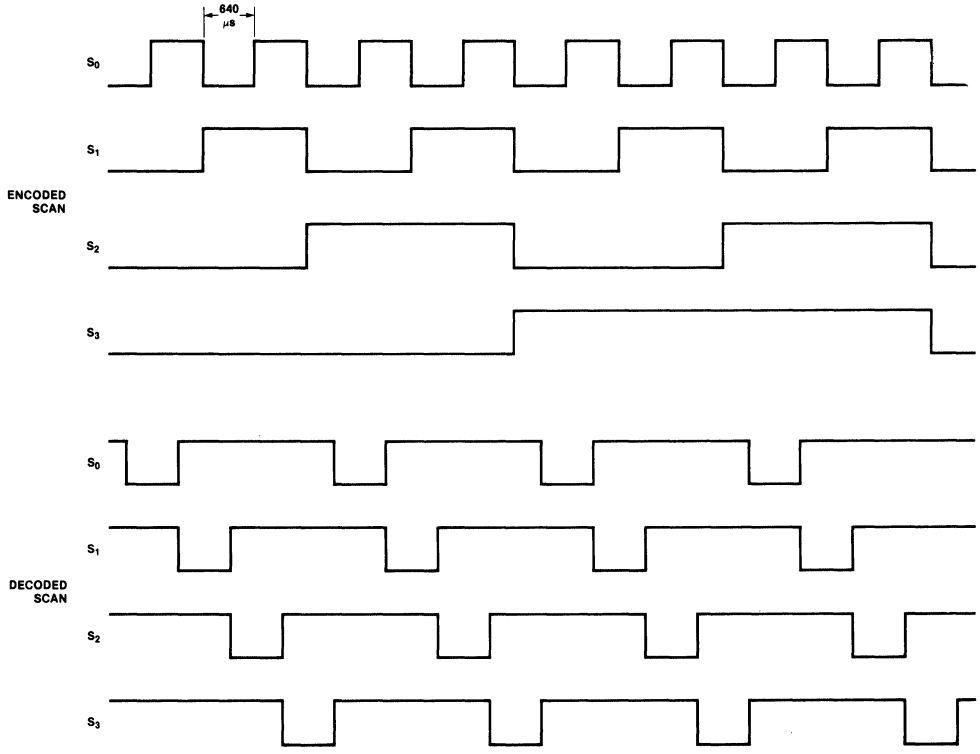
2. Write Operation



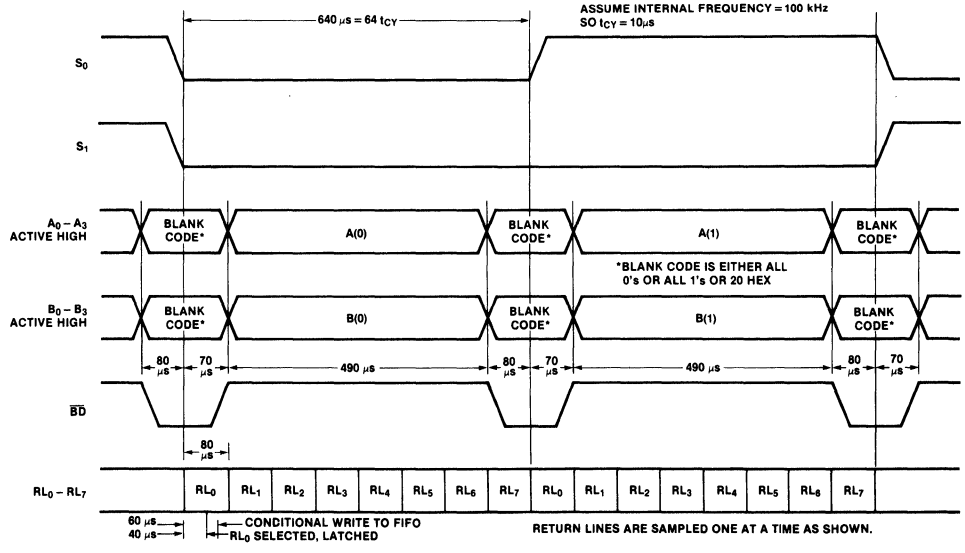
3. Clock Input



SCAN WAVEFORMS



DISPLAY WAVEFORMS



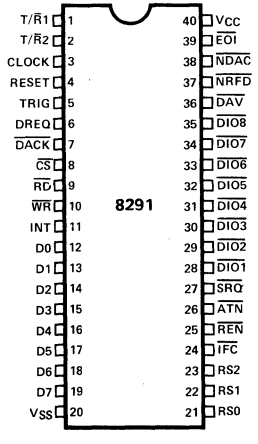
NOTE: SHOWN IS ENCODED SCAN LEFT ENTRY
S_2-S_3 ARE NOT SHOWN BUT THEY ARE SIMPLY S_1 DIVIDED BY 2 AND 4

8291
GPIB TALKER/LISTENER

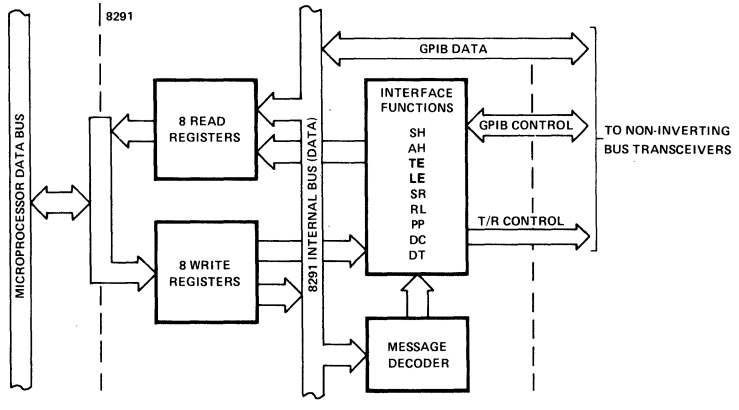
- Designed to Interface Microprocessors (e.g., 8080, 8085, 8086, 8048) to an IEEE Standard 488 Digital Interface Bus
- Programmable Data Transfer Rate
- Complete Source and Acceptor Handshake
- Complete Talker and Listener Functions with Extended Addressing
- Service Request, Parallel Poll, Device Clear, Device Trigger, Remote/Local Functions
- Selectable Interrupts
- On-Chip Primary and Secondary Address Recognition
- Automatic Handling of Addressing and Handshake Protocol
- Provision for Software Implementation of Additional Features
- 1 – 8 MHz Clock Range
- 16 Registers (8 Read, 8 Write), 2 for Data Transfer, the Rest for Interface Function Control, Status, etc.
- Directly Interfaces to External Non-Inverting Transceivers for Connection to the GPIB
- Provides Three Addressing Modes, Allowing the Chip to be Addressed Either as a Major or a Minor Talker/Listener with Primary or Secondary Addressing
- DMA Handshake Provision Allows for Bus Transfers without CPU Intervention
- Trigger Output Pin
- On-Chip EOS (End of Sequence) Message Recognition Facilitates Handling of Multi-Byte Transfers

The 8291 GPIB Talker/Listener is a microprocessor-controlled chip designed to interface microprocessors (e.g., 8048, 8080, 8085, 8086) to an IEEE Standard 488 Instrumentation Interface Bus. It implements all of the Standard's interface functions except for the controller.

PIN CONFIGURATION



BLOCK DIAGRAM

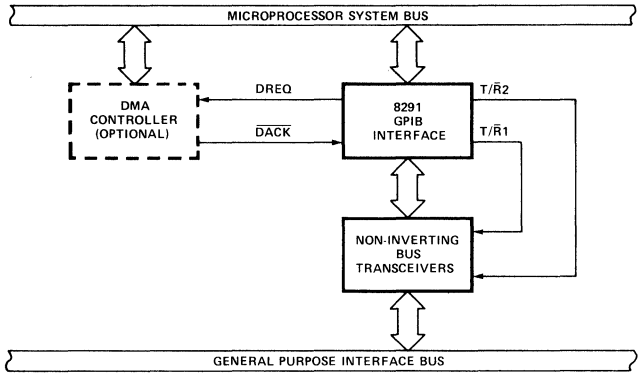


PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
D_0-D_7	I/O	12-19	Data bus port, to be connected to microprocessor data bus.
RS_0-RS_2	I	21-23	Register select inputs, to be connected to three non-multiplexed microprocessor address bus lines. Select which of the 8 internal read (write) registers will be read from (written into) with the execution of $\overline{RD}$ ($\overline{WR}$).
$\overline{CS}$	I	8	Chip select. When low, enables reading from or writing into the register selected by RS_0-RS_2.
$\overline{RD}$	I	9	Read strobe. When low, selected register contents are read by the CPU.
$\overline{WR}$	I	10	Write strobe. When low, data is written into the selected register.
INT ($\overline{INT}$)	O	11	Interrupt request to the microprocessor, set high for request and cleared when the appropriate register is accessed by the CPU. May be software configured to be active low.
DREQ	O	6	DMA request, normally low, set high to indicate byte output or byte input, in DMA mode; reset by $\overline{DACK}$.
$\overline{DACK}$	I	7	DMA acknowledge. When low, resets DREQ and selects data in/data out register for DMA data transfer (actual transfer done by $\overline{RD}$/$\overline{WR}$ pulse). Must be high if DMA is not used.
TRIG	O	5	Trigger output, normally low; generates a triggering pulse with 1μsec min. width in response to the GET bus command or Trigger auxiliary command.
CLOCK	I	3	External clock input, used only for T_1 delay generator. May be any speed in 1-8 MHz range.
RESET	I	4	Reset input. When high, forces the device into an "Idle" (initialization) mode. The device will remain at "Idle" until released by the microprocessor.
$\overline{DIO_1}$-$\overline{DIO_8}$	I/O	28-35	8-bit GPIB data port, used for bidirectional data byte transfer between 8291 and GPIB via non-inverting external line transceivers.
$\overline{DAV}$	I/O	36	Data valid; GPIB handshake control line. Indicates the availability and validity of information on the DIO lines.
$\overline{NRFD}$	I/O	37	Not ready for data; GPIB handshake control line. Indicates the condition of readiness of device(s) connected to the bus to accept data.
$\overline{NDAC}$	I/O	38	Not data accepted; GPIB handshake control line. Indicates the condition of acceptance of data by the device(s) connected to the bus.
$\overline{ATN}$	I	26	Attention; GPIB command line. Specifies how data on DIO lines are to be interpreted.
$\overline{IFC}$	I	24	Interface clear; GPIB command line. Places the interface functions in a known quiescent state.
$\overline{SRQ}$	O	27	Service request; GPIB command line. Indicates the need for attention and requests an interruption of the current sequence of events on the GPIB.
$\overline{REN}$	I	25	Remote enable; GPIB command line. Selects (in conjunction with other messages) remote or local control of the device.
$\overline{EOI}$	I/O	39	End or identify; GPIB command line. Indicates the end of a multiple byte transfer sequence or, in conjunction with ATN, addresses the device during a polling sequence.
T/$\overline{R}$1	O	1	External transceivers control line. Set high to indicate output data/signals on the DIO_1-DIO_8 and DAV lines and input signals on the NRFD and NDAC lines (active source handshake). Set low to indicate input data/signals on the DIO_1-DIO_8 and DAV lines and output signals on the NRFD and NDAC lines (active acceptor handshake).
T/$\overline{R}$2	O	2	External transceivers control line. Set high to indicate output signals on the EOI line. Set low to indicate expected input signal on the EOI line during parallel poll.
V_{CC}	P.S.	40	Positive power supply (5V ± 10%).
GND	P.S.	20	Potential ground circuit.

Note: All signals on the 8291 pins are specified with positive logic. However, IEEE 488 specifies negative logic on its 16 signal lines. Thus, the data is inverted once from D_0-D_7 to $\overline{DIO_1}$-$\overline{DIO_8}$ and non-inverting bus transceivers should be used.

8291 SYSTEM DIAGRAM



THE GENERAL PURPOSE INTERFACE BUS (GPIB)

The General Purpose Interface Bus (GPIB) is defined in the IEEE Standard 488-1978 "Digital Interface for Programmable Instrumentation." Although a knowledge of this standard is assumed, Figure 1 provides the bus structure for quick reference. Also, Tables 1 and 2 reference the interface state mnemonics and the interface messages respectively. Modified state diagrams for the 8291 are presented in Appendix A.

GENERAL DESCRIPTION

The 8291 is a microprocessor controlled device designed to interface microprocessors e.g., 8048, 8080, 8085, 8086 to the GPIB. It implements all of the interface functions defined in the IEEE 488 Standard. If an implementation of the Standard's Controller function is desired, it can be connected with an Intel® 8292 to form a complete interface.

The 8291 handles communication between a microprocessor controlled device and the GPIB. Its capabilities include data transfer, handshake protocol, talker/listener addressing procedures, device clearing and triggering, service request, and both serial and parallel polling schemes. In most procedures, it does not disturb the microprocessor unless a byte is waiting on input or a byte sent on output (output buffer empty).

The 8291 architecture includes 16 registers. Eight of these registers may be written into by the microprocessor. The other eight registers may be read by the microprocessor. One each of these read and write registers is for direct data transfers. The rest of the write registers control the various features of the chip, while the rest of the read registers provide the microprocessor with a monitor of GPIB states, various bus conditions, and device conditions.

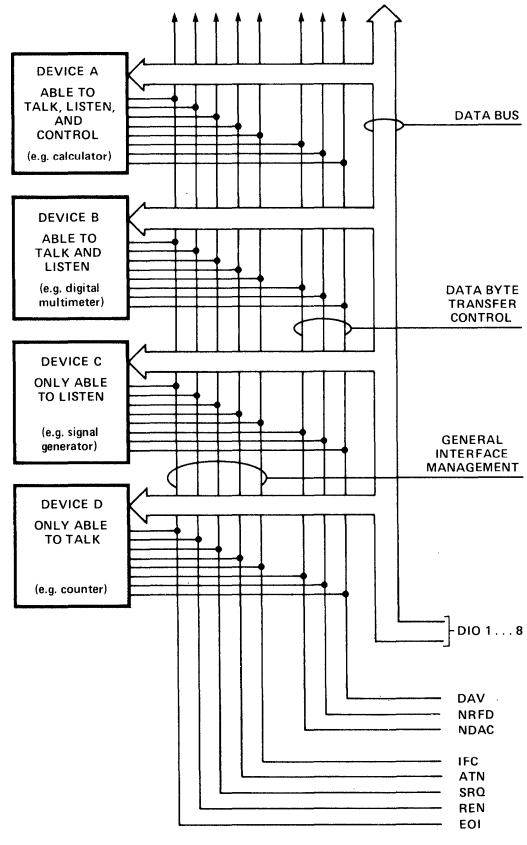


Figure 1. Interface Capabilities and Bus Structure.

GPIB Addressing

Each device connected to the GPIB must have at least one address whereby the controller device in charge of the bus can configure it to talk, listen, or send status. An 8291 implementation of the GPIB offers the user three addressing modes from which the device can be initialized for each application. The first of these modes allows for the device to have two separate primary addresses. The second mode allows the user to implement a single talker/listener with a two byte address (primary address + secondary address). The third mode again allows for two distinct addresses but in this instance, they can each have a two-byte address. However, this mode requires that the secondary addresses be passed to the microprocessor for verification. These three addressing schemes are described in more detail in the discussion of the Address registers.

TABLE 1.
IEEE 488 INTERFACE STATE MNEMONICS

Mnemonic	State Represented
ACDS	Accept Data State
ACRS	Acceptor Ready State
AIDS	Acceptor Idle State
ANRS	Acceptor Not Ready State
APRS	Affirmative Poll Response State
AWNS	Acceptor Wait for New Cycle State
CACS	Controller Active State
CADS	Controller Addressed State
CAWS	Controller Active Wait State
CIDS	Controller Idle State
CPPS	Controller Parallel Poll State
CPWS	Controller Parallel Poll Wait State
CSBS	Controller Standby State
CSNS	Controller Service Not Requested State
CSRS	Controller Service Requested State
CSWS	Controller Synchronous Wait State
CTRS	Controller Transfer State
DCAS	Device Clear Active State
DCIS	Device Clear Idle State
DTAS	Device Trigger Active State
DTIS	Device Trigger Idle State
LACS	Listener Active State
LADS	Listener Addressed State
LIDS	Listener Idle State
LOCS	Local State
LPAS	Listener Primary Addressed State
LPIS	Listener Primary Idle State
LWLS	Local With Lockout State
NPRS	Negative Poll Response State

Mnemonic	State Represented
PACS	Parallel Poll Addressed to Configure State
PPAS	Parallel Poll Active State
PPIS	Parallel Poll Idle State
PPSS	Parallel Poll Standby State
PUCS	Parallel Poll Unaddressed to Configure State
REMS	Remote State
RWLS	Remote With Lockout State
SACS	System Control Active State
SDYS	Source Delay State
SGNS	Source Generate State
SIAS	System Control Interface Clear Active State
SIDS	Source Idle State
SIIS	System Control Interface Clear Idle State
SINS	System Control Interface Clear Not Active State
SIWS	Source Idle Wait State
SNAS	System Control Not Active State
SPAS	Serial Poll Active State
SPIS	Serial Poll Idle State
SPMS	Serial Poll Mode State
SRAS	System Control Remote Enable Active State
SRIS	System Control Remote Enable Idle State
SRNS	System Control Remote Enable Not Active State
SRQS	Service Request State
STRS	Source Transfer State
SWNS	Source Wait for New Cycle State
TACS	Talker Active State
TADS	Talker Addressed State
TIDS	Talker Idle State
TPIS	Talker Primary Idle State

– – – – – – The Controller function is implemented on the Intel® 8292.

TABLE 2.
IEEE 488 INTERFACE MESSAGE REFERENCE LIST

Mnemonic	Message	Interface Function(s)
LOCAL MESSAGES RECEIVED (By Interface Functions)		
*gts	go to standby	C
ist	individual status	PP
lon	listen only	L, LE
lpe	local poll enable	PP
nba	new byte available	SH
pon	power on	SH,AH,T,TE,L,LE,SR,RL,PP,C
rdy	ready	AH
*rpp	request parallel poll	C
*rsc	request system control	C
rsv	request service	SR
rtl	return to local	RL
*sic	send interface clear	C
*sre	send remote enable	C
*tca	take control asynchronously	C
*tcs	take control synchronously	AH, C
ton	talk only	T, TE
REMOTE MESSAGES RECEIVED		
ATN	Attention	SH,AH,T,TE,L,LE,PP,C
DAB	Data Byte	(Via L, LE)
DAC	Data Accepted	SH
DAV	Data Valid	AH
DCL	Device Clear	DC
END	End	(via L, LE)
GET	Group Execute Trigger	DT
GTL	Go to Local	RL
IDY	Identify	L,LE,PP
IFC	Interface Clear	T,TE,L,LE,C
LLO	Local Lockout	RL
MLA	My Listen Address	L,LE,RL,T,TE
MSA	My Secondary Address	TE,LE,RL
MTA	My Talk Address	T,TE,L,LE
OSA	Other Secondary Address	TE
OTA	Other Talk Address	T, TE
PCG	Primary Command Group	TE,LE,PP
†PPC	Parallel Poll Configure	PP
†[PPD)	Parallel Poll Disable	PP
†[PPE]	Parallel Poll Enable	PP
*PPR_N	Parallel Poll Response N	(via C)
†PPU	Parallel Poll Unconfigure	PP
REN	Remote Enable	RL
RFD	Ready for Data	SH
RQS	Request Service	(via L, LE)
[SDC]	Select Device Clear	DC
SPD	Serial Poll Disable	T, TE
SPE	Serial Poll Enable	T, TE
*SQR	Service Request	(via C)
STB	Status Byte	(via L, LE)
*TCT or [TCT]	Take Control	C
UNL	Unlisten	L, LE

*These messages are handled only by Intel's 8292.
†Undefined commands which may be passed to the microprocessor.

TABLE 2. (Cont'd)
IEEE 488 INTERFACE MESSAGE REFERENCE LIST

Mnemonic	Message	** Interface Function(s)
REMOTE MESSAGES SENT		
ATN	Attention	C
DAB	Data Byte	(via T, TE)
DAC	Data Accepted	AH
DAV	Data Valid	SH
DCL	Device Clear	(via C)
END	End	(via T)
GET	Group Execute Trigger	(via C)
GTL	Go to Local	(via C)
IDY	Identify	C
IFC	Interface Clear	C
LLO	Local Lockout	(via C)
MLA or [MLA]	My Listen Address	(via C)
MSA or [MSA]	My Secondary Address	(via C)
MTA or [MTA]	My Talk Address	(via C)
OSA	Other Secondary Address	(via C)
OTA	Other Talk Address	(via C)
PCG	Primary Command Group	(via C)
PPC	Parallel Poll Configure	(via C)
[PPD]	Parallel Poll Disable	(via C)
[PPE]	Parallel Poll Enable	(via C)
PPR_N	Parallel Poll Response N	PP
PPU	Parallel Poll Unconfigure	(via C)
REN	Remote Enable	C
RFD	Ready for Data	AH
RQS	Request Service	T, TE
[SDC]	Selected Device Clear	(via C)
SPD	Serial Poll Disable	(via C)
SPE	Serial Poll Enable	(via C)
SRQ	Service Request	SR
STB	Status Byte	(via T, TE)
TCT	Take Control	(via C)
UNL	Unlisten	(via C)

**All Controller messages must be sent via Intel's 8292.

8291 Registers

A bit-by-bit map of the 16 registers on the 8291 is presented in Table 3. A more detailed explanation of each of these registers and their functions follows. The access of these registers by the microprocessor is accomplished by using the $\overline{CS}$, $\overline{RD}$, $\overline{WR}$, and RS_0-RS_2 pins.

Register	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	RS_0-RS_2
All Read Registers	0	0	1	CCC
All Write Registers	0	1	0	CCC
Don't Care	1	X	X	XXX

TABLE 3. 8291 REGISTERS

READ REGISTERS	RS2	RS1	RS0	WRITE REGISTERS
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0	0	0	0	DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0
DATA IN				DATA OUT
CPT APT GET END DEC ERR BO BI	0	0	1	CPT APT GET END DEC ERR BO BI
INTERRUPT STATUS 1				INTERRUPT ENABLE 1
INT SPAS LLO REM SPASC LLOC REMC ADSC	0	1	0	0 0 DMAO DMAI SPASC LLOC REMC ADSC
INTERRUPT STATUS 2				INTERRUPT ENABLE 2
S8 SRQS S6 S5 S4 S3 S2 S1	0	1	1	S8 rsv S6 S5 S4 S3 S2 S1
SERIAL POLL STATUS				SERIAL POLL MODE
ton lon EOI LPAS TPAS LA TA MJMN	1	0	0	TO LO 0 0 0 0 ADM1 ADM0
ADDRESS STATUS				ADDRESS MODE
CPT7 CPT6 CPT5 CPT4 CPT3 CPT2 CPT1 CPT0	1	0	1	CNT2 CNT1 CNT0 COM4 COM3 COM2 COM1 COM0
COMMAND PASS THROUGH				AUX MODE
X DT0 DL0 AD5-0 AD4-0 AD3-0 AD2-0 AD1-0	1	1	0	ARS DT DL AD5 AD4 AD3 AD2 AD1
ADDRESS 0				ADDRESS 0/1
X DT1 DL1 AD5-1 AD4-1 AD3-1 AD2-1 AD1-1	1	1	1	EC7 EC6 EC5 EC4 EC3 EC2 EC1 EC0
ADDRESS 1				EOS

Data Registers

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

DATA-IN REGISTER (0R)

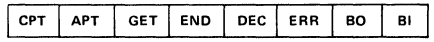
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

DATA-OUT REGISTER (0W)

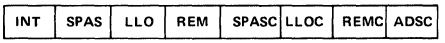
The data-in register is used to move data from the GPIB to the microprocessor or to memory when the 8291 is addressed to listen. Incoming information is separately latched by this register, and its contents are not destroyed by a write to the data-out register. The RFD (Ready for Data) message is held false until the byte is removed from the data in register, either by the microprocessor or by DMA. The 8291 then completes the handshake automatically. In RFD/DAV holdoff mode (see Auxiliary Register A), the handshake is not finished until a command is sent telling the 8291 to release the holdoff. In this way, the same byte may be read several times, or an over anxious talker may be held off until all available data has been processed.

When the 8291 is addressed to talk, it uses the data-out register to move data onto the GPIB. Upon a write to this register, the 8291 initiates and completes the handshake while sending the byte out over the bus. When the RFD/DAV holdoff mode is in effect, data is held until the release command is issued. Also, a read of the data-in register does not destroy the information in the data-out register.

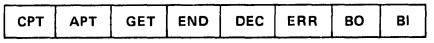
Interrupt Registers



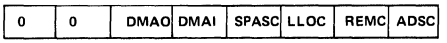
INTERRUPT STATUS 1 (1R)



INTERRUPT STATUS 2 (2R)



INTERRUPT ENABLE 1 (1W)



INTERRUPT ENABLE 2 (2W)

The 8291 can be configured to generate an interrupt to the microprocessor upon the occurrence of any of 12 conditions or events on the GPIB. Upon receipt of an interrupt, the microprocessor must read the Interrupt Status registers to determine which event has occurred, and then execute the appropriate service routine (if necessary). Each of the 12 interrupt status bits has a matching enable bit in the interrupt enable registers. These enable bits are used to select the events that will cause the INT pin to be asserted. Writing a logic "1" into any of these bits enables the corresponding interrupt status bits to generate an interrupt. Bits in the Interrupt Status registers are set regardless of the states of the enable bits. The Interrupt Status registers are then cleared upon being read or when a local pon (power-on) message is executed. If an event occurs while one of the Interrupt Status registers is being read, the event is typically held until after its register is cleared and then placed in the register.

The mnemonics for each of the bits in these registers and a brief description of their respective functions appears in Table 4. This table also indicates how each of the interrupt bits is set.

TABLE 4. Interrupt Bits

Indicates Undefined Commands	CPT	An undefined command has been received.
Set by (TPAS + LPAS)•SCG•ACDS•MODE 3	APT	A secondary address must be passed through to the microprocessor for recognition.
Set by DTAS	GET	A group execute trigger has occurred.
Set by (EOS + EOI)•LACS	END	An EOS or EOI message has been received.
Set by DCAS	DEC	Device Clear Active State has occurred.
Set by TACS•nba•DAC•RFD	ERR	Interface error has occurred; no listeners are active.
TACS•(SWNS + SGNS)	BO	A byte should be output.†
Set by LACS•ACDS	BI	A byte has been input.
Shows status of the INT pin	INT	These are status only. They will not generate interrupts, nor do they have corresponding mask bits.
The device has been enabled for a serial poll	SPAS	
The device is in local lock out state. (LWLS+RWLS)	LLO	
The device is in a remote state. (REMS+RWLS)	REM	
SPAS ⇄ $\overline{SPAS}$	SPASC	Serial Poll Active State change interrupt.†
LLO ⇄ NO LLO	LLOC	Local lock out change interrupt.
Remote ⇄ Local	REMC	Remote/Local change interrupt.
Addressed ⇄ Unaddressed	ADSC	Address status change interrupt.*

†See section on 8291A compatibility.
*In ton (talk-only) and lon (listen-only) modes, no ADSC interrupt is generated.

The BO and BI interrupts enable the user to perform data transfer cycles. BO indicates that a byte has been sent to the GPIB or the 8291 has been addressed to talk. A new data byte may be written into the Data Out register. It is set by the occurrence of TACS • (SWNS + SGNS). Hence, it is reset when a data byte is written into the Data Out register, when ATN is asserted on the GPIB, or when the device stops being addressed to talk. Similarly, BI is set when an input byte is accepted into the 8291 and reset when the microprocessor reads the Data In register. BO and BI are also reset by pon (power-on local message) and by a read of the Interrupt Status 1 register. However, if it is so desired, data transfer cycles may be performed without reading the Interrupt Status 1 register if all interrupts except for BO or BI are enabled; BO and BI will automatically reset after each byte is transferred.

If the 8291 is used without DMA, the BO and BI interrupts may be enabled through the DREQ pin. The DMAO and DMAI bits in the Interrupt Enable 2 register would be the corresponding enable bits for this feature. Thus, implementing this feature, with BO and BI enabled from the INT pin, allows for servicing of these interrupts without reading the Interrupt Status registers.

The ERR bit is set to indicate the bus error condition where the 8291 is an active talker, tries sending a byte to the GPIB, but there are no active listeners (e.g., all devices on the GPIB are in AIDS). The logical equivalent of (nba • TACS • DAC • RFD) will set this bit.

The DEC bit is set whenever DCAS has occurred. The user must define a known state to which all device functions will return in DCAS. Typically this state will be a power-on state. However, the state of the device functions at DCAS is at the designer's discretion. It should be noted that DCAS has no effect on the interface functions which are returned to a known state by the IFC (interface clear) message or the pon local message.

The END Interrupt bit may be used by the microprocessor to detect that a multi-byte transfer has been completed. The bit will be set when the 8291 is an active listener (LACS) and either EOS or EOI is received. EOS will generate an interrupt when the byte in the Data In register matches the byte in the EOS register. Otherwise the interrupt will be generated when a true input is detected at the EOI pin of the 8291.

The GET interrupt bit is used by the microprocessor to detect that DTAS has occurred. It is set by the 8291 when the GET message is received while it is addressed to listen. The TRIG output pin of the 8291 is also asserted when the GET message is received. Thus, the basic operation of the device may be started without involving the microprocessor.

The APT interrupt bit indicates to the processor that a secondary address is available in the CPT register for validation. This interrupt will only occur if Mode 3 addressing is in effect. (Refer to the section on addressing.) In Mode 2, secondary addresses will be recognized on the 8291. They will be ignored in Mode 1.

The CPT interrupt bit flags the occurrence of an undefined command and of all secondary commands following an undefined command. The Command pass through feature is enabled by the BO bit of Auxiliary register B.

$$UDC = [UCG + ACG(TADS \bullet \overline{PPC} + LADS \bullet \overline{TCT})] \bullet undefined \bullet BO$$

where:

ACG — Addressed Command Group
UCG — Universal Command Group
SCG — Secondary Command Group

Any message not decoded by the 8291 (not included in the state diagrams in Appendix B) becomes an undefined command. Note from the logic equation that any addressed command is automatically ignored when the 8291 is not addressed.

Undefined commands are read by the CPU from the Command Pass Through Register of the 8291. Until this register is read, the 8291 will hold off the handshake (only if the CPT feature is enabled).

An especially useful feature of the 8291 is its ability to generate interrupts from state transitions in the interface functions. In particular, the lower 4 bits of the Interrupt Status 2 register, if enabled by the corresponding enable bits, will cause an interrupt upon changes in the following states as defined in IEEE 488:

Bit 0	ADSC	change in LIDS or TIDS or MJMN
Bit 1	RLC	change in LOCS or REMS
Bit 2	LLOC	change in LWLS or RWLS
Bit 3	SPASC	change in SPAS

The upper 4 bits of the Interrupt Status 2 register are available to the processor as status bits. Thus, if one of the bits 1-3 generates an interrupt indicating a state change has taken place, the corresponding status bit (bits 5-7) may be read to determine what the new state is. To determine the nature of a change in addressed status (bit 0) the Address Status Register is available to be read. And finally, bit 7 monitors the state of the 8291 INT pin. Logically, it is an OR of all enabled interrupt status bits. One should note that bits 4-7 of the Interrupt Status 2 Register do not generate interrupts, but are available only to be read as status bits by the processor.

Bits 4 and 5 (DMAI, DMAO) of the Interrupt Enable 2 Register are available to enable direct data transfers between memory and the GPIB, DMAI (DMA in) enables the DREQ (DMA request) pin of the 8291 to be asserted upon the occurrence of BI. Similarly, DMAO (DMA out) enables the DREQ pin to be asserted upon the occurrence of BO. One might note that the DREQ pin may be used as a second interrupt output pin, monitoring BI and/or BO and enabled by DMAI and DMAO. One should note that the DREQ pin is not affected by a read of the Interrupt Status 1 Register. It is reset whenever a byte is written to the Data Out Register or read from the Data In Register.

To ensure that an interrupt status bit will not be cleared without being read, and will not remain uncleared after being read, the 8291 implements a special interrupt

handling procedures. When an enabled interrupt bit is set in either of the Interrupt Status Registers, the input of the registers are blocked until the set bit is read and reset by the microprocessor. Thus, potential problems arise when interrupt status changes while the register is being blocked. However, the 8291 stores all new interrupts in a temporary register and transfers them to the appropriate interrupt Status Register after the interrupt has been reset. In the Interrupt Status 1 Register and in ADSC bit, this transfer takes place only if the corresponding bits were read as zeroes. For the other status change bits in the Interrupt Status 2 Register, the transfer will always take place. However, even number of changes in these status bits during blocking time will cause no interrupt.

Serial Poll Registers

S8	SRQS	S6	S5	S4	S3	S2	S1

SERIAL POLL STATUS (3R)

S8	rsv	S6	S5	S4	S3	S2	S1

SERIAL POLL MODE (3W)

The Serial Poll Mode Register is used to establish the status byte that the 8291 sends out on the GPIB data lines when it receives the SPE (Serial Poll Enable) message. Bit 6 of this register is reserved for the rsv (request service) local message. Setting this bit to 1 causes the 8291 to assert its SRQ line, indicating its need for attention from the controller-in-charge of the GPIB. When service has been granted, the bit should be cleared by the microprocessor. The other bits of this register are available for sending status information over the GPIB. Sometime after the microprocessor initiates a request for service by setting bit 6, the controller of the GPIB sends the SPE message and then addresses the 8291 to talk. At this point, one byte of status is returned by the 8291 via the Serial Poll Mode Register.

The Serial Poll Status Register is available for reading the status byte in the Serial Poll Mode Register. The processor may check the status of a request for service by polling bit 6 of this register, which corresponds to SRQS (Service Request State). When a Serial Poll is conducted and the controller-in-charge reads the status byte, the SRQS bit is cleared. The SRQ line is tied to this bit, so that a request for service is terminated when the 8291's status byte is read. The rsv bit of the Serial Poll Mode Register must then be cleared by the microprocessor.

Address Registers

ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN

ADDRESS STATUS (4R)

X	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0

ADDRESS 0 (6R)

X	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1

ADDRESS 1 (7R)

TO	LO	0	0	0	0	ADM1	ADM0

ADDRESS MODE (4W)

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1

ADDRESS 0/1 (6W)

The Address Mode Register is used to select one of the five modes of addressing available on the 8291. It determines the way in which the 8291 uses the information in the Address 0 and Address 1 registers:

—In Mode 1, the contents of the Address 0 Register constitute the "Major" talker/listener address while the Address 1 Register represents the "Minor" talker/listener address. In applications where only one address is needed, the major talker/listener is used, and the minor talker/listener should be disabled. Loading an addres via the Address 0/1 Register into Address Registers 0 and 1 enables the major and minor talker/listener functions respectively.

—In Mode 2 the 8291 recognizes two sequential address bytes: a primary followed by a secondary. Both address bytes must be received in order to enable the device to talk or listen. In this manner, Mode 2 addressing implements the extended talker and listener functions as defined in IEEE 488.

To use Mode 2 addressing the primary address must be loaded into the Address 0 Register, and the Secondary address is placed in the Address 1 Register. With both primary and secondary addresses residing on chip, the 8291 can handle all addressing sequences without processor intervention.

—In Mode 3, the 8291 handles addressing just as it does in Mode 1, except that each Major or Minor primary address must be followed by a secondary address. All secondary addresses must be verified by the microprocessor when Mode 3 is used. When the 8291 is in TPAS or LPAS (talker/listener primary addresses state), and it does not recognize the byte on the DIO lines, an APT interrupt is generated (see section on Interrupt Registers) and the byte is available in the CPT (Command Pass-Through) Register. As part of its interrupt service routine, the microprocessor must read the CPT Register and write one of the following responses to the Auxiliary Mode Register:

1. 07H implies a non-valid secondary address
2. 0FH implies a valid secondary address

Setting the "ton" bit generates the local ton (talk-only) message and sets the 8291 to a talk-only mode. This mode allows the device to operate as a talker in an interface system without a controller.

Setting the "lon" bit generates the local lon (listen-only) message and sets the 8291 to a listen-only mode. This mode allows the device to operate as a listener in an interface system without a controller.

The mode of addressing implemented by the 8291 may be selected by writing one of the following bytes to the Address Mode Register:

Register Contents	Mode
10000000	Enable talk only mode (ton)
01000000	Enable listen only mode (lon)
11000000	The 8291 may talk to itself
00000001	Mode 1, (Primary-Primary)
00000010	Mode 2 (Primary-Secondary)
00000011	Mode 3 (Primary/APT-Primary/APT)

The Address Status Register contains information used by the microprocessor to handle its own addressing. This information includes status bits that monitor the address state of each talker/listener, "ton" and "lon" flags which indicate the talk only and listen only states, and an EOI bit which, when set, signifies that the END message came with the last data byte. LPAS and TPAS indicate that the listener or talker primary address has been received. The microprocessor can then use these bits when the secondary address is passed through to determine whether the 8291 is addressed to talk or listen. The LA (listener addressed) bit will be set when the 8291 is in LACS (Listener Active State) or in LADS (Listener Addressed State). Similarly, the TA (Talker Addressed bit will be set to indicate TACS or TADS, but also to indicate SPAS (Serial Poll Active State). The MJMN bit is used to determine whether the information in the other bits applies to the Major or Minor talker/listener. It is set to "1" when the Minor talker/listener is addressed. It should be noted that only one talker/listener may be active at any one time. Thus, the MJMN bit will indicate which, if either, of the talker/listeners is addressed or active.

The Address 0/1 Register is used for specifying the device's addresses according to the format selected in the Address Mode Register. Five bit addresses may be loaded into the Address 0 and Address 1 registers by writing into the Address 0/1 Register. The ARS bit is used to select which of these registers the other seven bits will be loaded into. The DT and DL bits may be used to disable the talker or listener function at the address signified by the other five bits. When Mode 1 addressing is used and only one primary address is desired, both the talker and the listener should be disabled at the Minor address.

As an example of how the Address 0/1 Register might be used, consider an example where two primary addresses are needed in the device. The Major primary address will be selectable only as a talker and the Minor primary address will be selectable only as a listener. This configuration of the 8291 is formed by the following sequence of writes by the microprocessor:

Operation	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	Data	RS2-RS0
1. Select addressing Mode 1	0	1	0	00000001	100
2. Load major address into Address 0 Register with listener function disabled.	0	1	0	001AAAAA	110
3. Load minor address into Address 1 Register with talker function disabled.	0	1	0	110BBBBB	110

At this point, the addresses AAAAA and BBBBB are stored in the Address 0 and Address 1 registers respectively, and are available to be read by the microprocessor. Thus, it is not necessary to store any address information elsewhere. Also, with the information stored in the Address 0 and Address 1 registers, processor intervention is not required to recognize addressing by the controller. Only in Mode 3, where secondary addresses are passed through, must the processor intervene in the addressing sequence.

Command Pass Through Register

CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

COMMAND PASS THROUGH (5R)

The Command Pass Through Register is used to transfer undefined 8-bit remote message codes from the GPIB to the microprocessor. When the CPT feature is enabled (bit B0 in Auxiliary Register B), any message not decoded by the 8291 becomes an undefined command. When Mode 3 addressing is used secondary addresses are also passed through the CPT Register. In either case, the 8291 will holdoff the handshake until the microprocessor reads this register and issues the VSCMD auxiliary command.

The CPT and APT interrupts flag the availability of undefined commands and secondary addresses in the CPT Register. The details of these interrupts are explained in the section on Interrupt Registers.

An added feature of the 8291 is its ability to handle undefined secondary commands following undefined primaries. Thus, the number of available commands for future IEEE 488 definition is increased; one undefined primary command followed by a sequence of as many as 32 secondary commands can be processed. The IEEE-488 Standard does not permit users to define their own commands, but upgrades of the standard are thus provided for.

The recommended use of the 8291's undefined command capabilities is for a controller-configured Parallel Poll. The PPC message is an undefined primary command typically followed by PPE, an undefined secondary command. For details on this procedure, refer to the section on Parallel Poll Protocol.

Auxiliary Mode Register

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

AUX MODE (5W)

CNT0—2:CONTROL BITS
COM0—4:COMMAND BITS

The Auxiliary Mode Register contains a three-bit control field and a five-bit command field. It is used for several purposes on the 8291:

1. To load "hidden" auxiliary registers on the 8291.
2. To issue commands from the microprocessor to the 8291.
3. To preset an internal counter used to generate T1, delay in the Source Handshake function, as defined in IEEE 488.

Table 5 summarizes how these tasks are performed with the Auxiliary Mode Register. Note that the three control bits determine how the five command bits are interpreted.

TABLE 5

CODE		COMMAND
CONTROL BITS	COMMAND BITS	
000	0CCCC	Execute auxiliary command CCCC
001	0FFFF	Preset internal counter to match external clock frequency of FFFF MHz (FFFF - binary representation of 1 to 8 MHz)
100	DDDDD	Write DDDDD into auxiliary register A
101	0DDDD	Write DDDD into auxiliary register B
011	$USP_3P_2P_1$	Enable/disable parallel poll either in response to remote messages (PPC followed by PPE or PPD) or as a local lpe message. (Enable if U = 0, disable if U = 1.)

AUXILIARY COMMANDS

Auxiliary commands are executed by the 8291 whenever 0000CCCC is written into the Auxiliary Mode Register, where CCCC is the 4-bit command code.

4-Bit Code	Description
0000	Immediate Execute pon — This command resets the 8291 to a power up state (local pon message as defined in IEEE 488). The following conditions constitute the power up state: 1. All talkers and listeners are disabled. 2. No interrupt status bits are set. The 8291 is designed to power up in certain states as specified in the IEEE 488 state diagrams. Thus, the following states are in effect in the power up state: SIDS, AIDS, TIDS, LIDS, NPRS, LOCS, and PPIS. The "0000" pon is an immediate execute command (a pon pulse). It is also used to release the "initialize" state generated by either an external reset pulse or the "0010" Chip Reset command.
0010	Chip Reset (Initialize) — This command has the same effect as a pulse applied to the Reset pin. (Refer to the section on Reset Procedure.)
0011	Finish Handshake — This command finishes a handshake that was stopped because of a holdoff on RFD or DAV. (Refer to Auxiliary Register A.)
0100	Trigger — A "Group Execute Trigger" is forced by this command. It has the same effect as a GET command issued by the controller-in-charge of the GPIB, but does not cause a GET interrupt.
0101	rtl† — This command corresponds to the local rtl message as defined in IEEE 488. The 8291 will go to a local state if local lockout is not in effect.
0110	Send EOI — The EOI line of the 8291 may be asserted with this command. The command causes EOI to go true with the next byte transmitted. The EOI line is then cleared upon completion of the handshake for that byte.
0111, 1111	Non-Valid/Valid Secondary Address or Command (VSCMD) — This command informs the 8291 that the secondary address received by the microprocessor was valid or invalid (0111 → invalid, 1111 → valid). If Mode 3 addressing is used, the processor must field each extended address and respond to it, or the GPIB will hang up. Note that the COM3 bit is the invalid/valid flag. The valid (1111) command is also used to tell the 8291 to continue from the command-pass-through state (immediate execute command).
0001, 1001	Parallel Poll Flag (local "ist" message) — This command sets (1001) or clears (0001) the parallel poll flag. A "1" is sent over the assigned data line (PPR-Parallel Poll Response true) only if the parallel poll flag matches the sense bit from the lpe local message (or indirectly from the PPE message). For a more complete description of the Parallel Poll features and procedures refer to the section on Parallel Poll Protocol.

†See section on 8291A compatibility.

INTERNAL COUNTER

The internal counter determines the delay time allowed for the settling of data on the DIO lines. This delay time is defined as T_1 in IEEE 488 and appears in the Source Handshake state diagram between SDYS and STRS. As such, DAV is asserted T_1 *after* the DIO lines are driven. Consequently, T_1 is a major factor in determining the data transfer rate of the 8291 over the GPIB (T_1 = TWRDV2-TWRD15).

When open-collector transceivers are used for connection to the GPIB, T_1 is defined by IEEE 488 to be 2μsec. By writing 0010FFFF into the Auxiliary Mode Register, the counter is preset to match a f_C MHz clock input, where FFFF is the binary representation of N_F ($1 \leq N_F \leq 8$, $N_F = (FFFF)_2$). When $N_F = f_C$, a 2μsec T_1 delay will be generated before each DAV asserted.

$$T_{1(\mu sec)} = \frac{2N_F}{f_C} + t_{SYNC}, \quad 1 \leq N_F \leq 8$$

t_{SYNC} is a synchronization error, greater than zero and smaller than the larger of T clock high and T clock low. (For a 50% duty cycle clock, t_{SYNC} is less than half the clock cycle).

If it is necessary that T_1 be different from 2μsec, N_F may be set to a value other than f_C. In this manner, data transfer rates may be programmed for a given system. In small systems, for example, where transfer rates exceeding GPIB specifications are required, one may set $N_F < f_C$ and decrease T_1.

When tri-state transceivers are used, IEEE 488 allows a higher transfer rate (lower T_1). Use of the 8291 with such transceivers is enabled by setting B2 in Auxiliary Register B. In this case, setting $N_F = f_C$ causes a T_1 delay of 2μsec to be generated for the first byte transmitted — all subsequent bytes will have a delay of 500 nsec.

$$T_1(\text{High Speed})\ \mu sec = \frac{N_F}{2f_C} + t_{SYNC}$$

Thus, setting $N_F = 1$ using a 8 MHz clock will generate for a 50% duty cycle clock (t_{SYNC}<63 nsec):

$$T_{1(HS)} = \frac{1}{2x8} + 0.063 = 125 \text{ nsec max.}$$

AUXILIARY REGISTER A

Auxiliary Register A is a "hidden" 5-bit register which is used to enable some of the 8291 features. Whenever a 100 $A_4A_3A_2A_1A_0$ byte is written into the Auxiliary Register, it is loaded with the data $A_4A_3A_2A_1A_0$. Setting the respective bits to "1" enables the following features:

A_0 — RFD/DAV Holdoff on all Data: If the 8291 is listening, RFD will not be sent true until the "finish handshake" auxiliary command is issued by the microprocessor. If the 8291 is talking, DAV is not sent true until the "finish handshake" command is given. In both cases, the holdoff will be in effect for each data byte.

A_1 — RFD/DAV Holdoff on End: This feature enables the holdoff on EOI or EOS (if enabled). However, no holdoff will be in effect on any other data bytes.

A_2 — End on EOS Received: Whenever the byte in the Data In Register matches the byte in the EOS Register, the End interrupt bit will be set in the Interrupt Status 1 Register.

A_3 — Output EOI on EOS Sent: Any occurrence of data in the Data Out Register matching the EOS Register causes the EOI line to be sent true along with the data.

A_4 — EOS Binary Compare: Setting this bit causes the EOS Register to function as a full 8-bit word. When it is not set, the EOS Register is a 7-bit word (for ASCII characters).

If $A_0 = A_1 = 1$, a special "continuous Acceptor Handshake cycling" mode is enabled. This mode should be used only in a controller system configuration, where both the 8291 and the 8292 are used. It provides a continuous cycling through the Acceptor Handshake state diagram, requiring no local messages from the microprocessor; the rdy local message is automatically generated when in ANRS. As such, the 8291 Acceptor Handshake serves as the controller Acceptor Handshake. Thus, the controller cycles through the Acceptor Handshake without delaying the data transfer in progress. When the tcs local message is executed, the 8291 is taken out of the "continuous AH cycling" mode, the GPIB hangs up in ANRS, and a BI interrupt is generated to indicate that control may be taken. A simpler procedure may be used when a "tcs on end of block" is executed; the 8291 may stay in "continuous AH cycling". Upon the end of a block (EOI or EOS received), a holdoff is generated, the GPIB hangs up in ANRS, and control may be taken.

AUXILIARY REGISTER B

Auxiliary Register B is a "hidden" 4-bit register which is used to enable some of the features of the 8291. Whenever a 1010$B_3B_2B_1B_0$ is written into the Auxiliary Mode Register, it is loaded with the data $B_3B_2B_1B_0$. Setting the respective bits to "1" enables the following features:

B_0 — Enable Undefined Command Pass Through: This feature allows any commands not recognized by the 8291 to be handled in software. If enabled, this feature will cause the 8291 to holdoff the handshake when an undefined command is received. The microprocessor must then read the command from the Command Pass Through Register and send the VSCMD auxiliary command. Until the VSCMD command is sent, the handshake holdoff will be in effect.

B_1 — Send EOI in SPAS: This bit enables EOI to be sent with the status byte; EOI is sent true in Serial Poll Active State. Otherwise, EOI is sent false in SPAS.

B_2 — Enable High Speed Data Transfer: This feature may be enabled when tri-state external transceivers are used. The data transfer rate is limited by T_1 (delay time generated in the Source Handshake function), which is defined according to the type of transceivers used. When the "High Speed" feature is enabled, T_1 = 2 microseconds is generated for the first byte transmitted after each true to false transition of ATN. For all subsequent bytes, T_1 = 500 nanoseconds. Refer to the Internal Counter section for an explanation of T_1 duration as a function of B_2 and of clock frequency.

B_3 — Enable Active Low Interrupt: Setting this bit causes the polarity of the INT pin to be reversed, providing an output signal compatible with Intel's MCS-48™. Interrupt registers are not affected by this bit.

PARALLEL POLL PROTOCOL

Writing a 011USP$_3$P$_2$P$_1$ into the Auxiliary Mode Register will enable (U = 0) or disable (U = 1) the 8291 for a parallel poll. When U = 0, this command is the "lpe" (local poll enable) local message as defined in IEEE 488. The "S" bit is the sense in which the 8291 is enabled; only if the Parallel Poll Flag ("ist" local message) matches this bit will the Parallel Poll Response, PPR$_N$, be sent true (Response = $\overline{S \oplus ist}$). The bits P$_3$P$_2$P$_1$ specify which of the eight data lines PPR$_N$ will be sent over. Thus, once the 8291 has been configured for Parallel Poll, whenever it senses both EOI and ATN true, it will automatically compare its PP flag with the sense bit and send PPR$_N$ true or false according to the comparison.

If a PP$_2$* implementation is desired, the "lpe" and "ist" local messages are all that are needed. Typically, the user will configure the 8291 for Parallel Poll immediately after initialization. During normal operation the microprocessor will set or clear the Parallel Poll Flag (ist) according to the device's need for service. Consequently the 8291 will be set up to give the proper response to IDY (EOI • ATN) without directly involving the microprocessor.

If a PP1* implementation is desired, the undefined command features of the 8291 must be used. In PP1, the 8291 is indirectly configured for Parallel Poll by the active controller on the GPIB. The sequence at the 8291 being enabled or disabled remotely is as follows:

1. The PPC message is received and is loaded into the Command Pass Through Register as an undefined command. A CPT interrupt is sent to the microprocessor, the handshake is automatically held off.
2. The microprocessor reads the CPT Register and sends VSCMD to the 8291, releasing the handshake.
3. Having received an undefined primary command, the 8291 is set up to receive an undefined secondary command (the PPE or PPD message). This message is also received into the CPT Register, the handshake is held off, and the CPT interrupt is generated.
4. The microprocessor reads the PPE or PPD message and writes the command into the Auxiliary Mode Register (bit 7 should be cleared first). Finally, the microprocessor sends VSCMD and the handshake is released.

*As defined in IEEE Standard 488.

End of Sequence (EOS) Register

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0

EOS REGISTER

The EOS Register and its features offer an alternative to the "Send EOI" auxiliary command. A seven or eight bit byte (ASCII or binary) may be placed in the register to flag the end of a block or read. The type of EOS byte to be used is selected in Auxiliary Register bit A_4.

If the 8291 is a listener, and the "End on EOS Received" is enabled at bit A_2, then an End interrupt is generated in the Interrupt Status 1 Register whenever the byte in the Data-In Register matches the byte in the EOS Register.

If the 8291 is a talker, and the "Output EOI on EOS Sent" is enabled at bit A_3, then the EOI line is sent true with the next data byte whenever the contents of the Data Out Register match the EOS register.

Reset Procedure

The 8291 is reset to an initialization state either by a pulse applied to its Reset pin, or by a reset auxiliary command (02H written into the Auxiliary Command Register). The following conditions are caused by a reset pulse (or local reset command):

1. A "pon" local message as defined by IEEE 488 is held true until the initialization state is released.
2. The Interrupt Status Registers are cleared (not Interrupt Enable Registers).
3. Auxiliary Registers A and B are cleared.
4. The Serial Poll Mode Register is cleared.
5. The Parallel Poll Flag is cleared.
6. The EOI bit in the Address Status Register is cleared.
7. N_F in the Internal Counter is set to 8 MHz. This setting causes the longest possible t_1 delay to be generated in the Source Handshake (16 μsec for 1 MHz clock).
8. The rdy local message is sent.

The initialization state is released by an "immediate execute pon" command (00H written into the Auxiliary Command Register).

The suggested initialization sequence is:

1. Apply a reset pulse or send the reset auxiliary command.
2. Set the desired initial conditions by writing into the Interrupt Enable, Serial Poll Mode, Address Mode, Address 0/1, and EOS Registers. Auxiliary Registers A and B, and the internal counter should also be initialized.
3. Send the "immediate execute pon" auxiliary command to release the initialization state.
4. If a PP$_2$ Parallel Poll implementation is to be used the "lpe" local message may be sent, enabling the 8291 for a Parallel Poll Response on an assigned line. (Refer to the section on Parallel Poll Protocol.)

Using DMA

The 8291 may be connected to the Intel® 8237 or 8257 DMA Controllers for DMA operation. The DREQ pin of the 8291 requests a DMA byte transfer from the 8237. It is set by BO or BI flip flops, enabled by the DMAO and DMAI bits in the Interrupt Enable 2 Register. (After reading, the INT1 register BO and BI interrupts will be cleared but not BO and BI in DREQ equation.)

The $\overline{DACK}$ pin is driven by the 8237 in response to the DMA request. When $\overline{DACK}$ is true (active low) it sets $\overline{CS}$ = RS0 = RS1 = RS2 = 0 such that the $\overline{RD}$ and $\overline{WR}$ signals sent by the 8237 refer to the Data In and Data Out Registers. Also, the DMA request line is reset by $\overline{DACK}$.

DMA input sequence:

1. A data byte is accepted from the GPIB by the 8291.
2. A BI interrupt is generated and DREQ is set.
3. $\overline{DACK}$ is asserted by the 8237 and DREQ is reset.
4. $\overline{RD}$ is driven by the 8237 and the contents of the Data In Register are transferred to MCS™ bus.
5. The 8291 sends RFD true on the GPIB and proceeds with the Acceptor Handshake protocol.

DMA output sequence:

1. A BO interrupt is generated (indicating that the Data Out Register is empty) and DREQ is asserted.
2. $\overline{DACK}$ is asserted by the 8237 and DREQ is reset.
3. $\overline{WR}$ is driven by the 8237 and a byte is transferred from the MCS bus into the Data Out Register.
4. The 8291 sends DAV true on the GPIB and proceeds with the Source Handshake protocol.

It should be noted that each time the device is addressed, the Address Status Register should be read, and the 8237 should be initialized accordingly. (Refer to the 8237 or 8257 Data Sheets.)

System Configuration

Microprocessor Bus Connection

The 8291 is 8080, 8048, 8085, 8088, and 8086 compatible. The three address pins (RS_0, RS_1, RS_2) should be connected to the non-multiplexed address bus (for example: A_8, A_9, A_{10}). In case of 8080, any address lines may be used.

External Transceivers Connection

The 8293 GPIB Transceiver interfaces the 8291 directly to the IEEE-488 bus. The 8291 and two 8293's can be configured as a talker/listener (see Figure 2) or as with the 8292 as a talker/listener/controller (see Figure 3). Absolutely no active or passive external components are required to comply with the complete IEEE-488 electrical specification.

8291 to 8291A Software Compatibility

Intel will be improving the 8291 by manufacturing an 8291A. To maintain software compatibility between the 8291 and the 8291A, the following precautions should be taken in the 8291 software:

1. BO interrupt indicates that the 8291 is ready to talk and needs a byte to output via the source handshake. The software should ensure that BO is true before writing a byte to the Data Out Register (even for the first byte after being addressed to talk).

2. SPASC interrupt should not be used during a Serial Poll sequence to determine when the Status Byte has been issued after a Service Request.

   Before setting rsv, SPAS in register 2 should be zero. After setting rsv, the processor should poll the SRQS bit in register 3, and when it is clear the Status Byte has been issued. The processor should then write an rsv local message clearing rsv.

   The definition of the SPASC interrupt will change in the 8291A. SPASC (Serial Poll Active State *Change*) in the 8291 is set by a transition into or out of SPAS. SPASC (Serial Poll Active State *Complete*) in the 8291A will be set only by the actual transfer of a Status Byte (APRSxSTRSxSPAS).

3. The 8291 rtl local message is set by the rtl Auxiliary Command and is cleared automatically by the 8291. The 8291A will have a Set rtl Auxiliary Command (1101) and a Clear rtl Auxiliary Command (0101). Thus, the 8291 programmer should write a Set rtl Auxiliary Command followed by a Clear rtl Auxiliary Command which will have the effect of writing two consecutive rtl commands.

4. User's software can distinguish between the 8291 and the 8291A as follows:
   a) pon (00H to register 5)
   b) RESET (02H to register 5)
   c) Read Interrupt Status Register 1. If BO interrupt is set, the device is the 8291. If BO is clear, it is the 8291A.

   This can be used to set a flag in the user's software which will permit special routines to be executed for each device. It could be included as part of a normal initialization procedure as the first step after a chip reset.

The 8291A will be a significant improvement over the 8291. Users should plan to convert to this product when it is available.

DEVICE ELECTRICAL CHARACTERISTICS

D.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5V ± 10%

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL}=2mA (4mA for TR1 pin)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA (-150μA for SRQ pin)
V_{OH-INT}	Interrupt Output High Voltage	2.4		V	I_{OH}=-400μA
		3.5		V	I_{OH}=-50μA
I_{IL}	Input Leakage		10	μA	V_{IN}=0V to V_{CC}
I_{LOL}	Output Leakage Current		-10	μA	V_{OUT}=0.45V
I_{LOH}	Output Leakage Current		10	μA	V_{OUT}=V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	T_A=0°C

A.C. CHARACTERISTICS

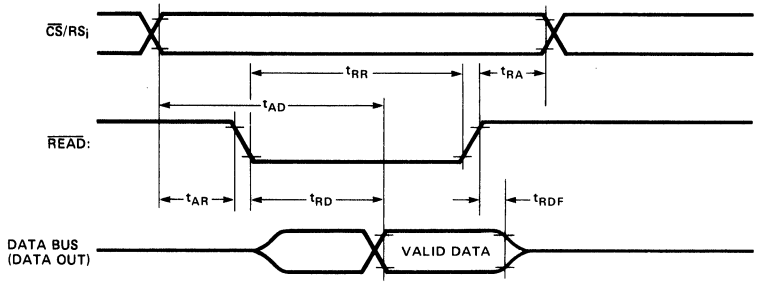
V_{CC} = 5V ± 10%, Commercial: T_A = 0°C to 70°C

Symbol	Parameter	Min.	Max.	Unit
t_{AR}	Address Stable Before $\overline{READ}$	0		nsec[1]
t_{RA}	Address Hold After $\overline{READ}$	0		nsec[1]
t_{RR}	$\overline{READ}$ width	140		nsec[2]
t_{AD}	Address Stable to Data Valid		250	nsec[1]
t_{RD}	$\overline{READ}$ to Data Valid		100	nsec[2]
t_{RDF}	Data Float After $\overline{READ}$	0	60[2]	nsec
t_{AW}	Address Stable Before $\overline{WRITE}$	0		nsec[1]
t_{WA}	Address Hold After $\overline{WRITE}$	0		
t_{WW}	$\overline{WRITE}$ Width	170		nsec[1]
t_{DW}	Data Set Up Time to the Trailing Edge of $\overline{WRITE}$	150		nsec[1]
t_{WD}	Data Hold Time After $\overline{WRITE}$	0		nsec[1]
t_{AKRQ}	$\overline{DACK}$↓ to DREQ↓		130	nsec
t_{DKDA6}	$\overline{DACK}$↓ to Up Data Valid		200	nsec

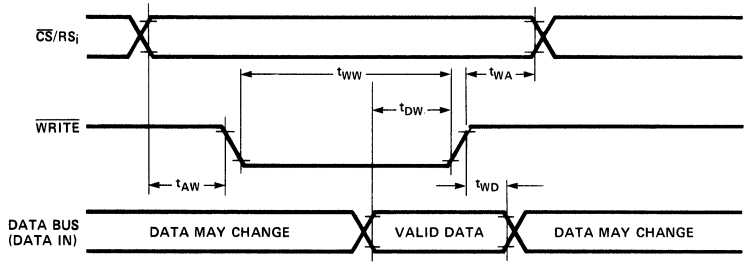
Notes:
1. 8080 System C_{Lmax} = 100pF; C_{Lmin} = 15pF; 3 MHz clock.
2. 8085 System C_L = 150pF; 4 MHz clock.

TIMING WAVEFORMS

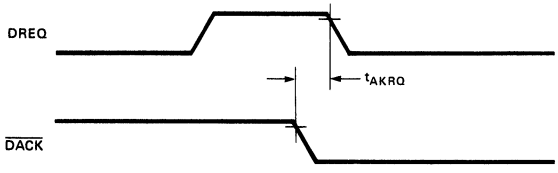
READ



WRITE



DMA



GPIB TIMINGS[1]

Symbol	Parameter	Max.	Unit	Test Conditions
TEOT13	$\overline{EOI}$↓ to TR1↑	135	nsec	PPSS, $\overline{ATN}$=0.45V
TEODI6	$\overline{EOI}$↓ to $\overline{DIO}$ Valid	155	nsec	PPSS, $\overline{ATN}$=0.45V
TEOT12	$\overline{EOI}$↓ to TR1↓	155	nsec	PPSS, $\overline{ATN}$=0.45V
TATND4	$\overline{ATN}$↓ to $\overline{NDAC}$↓	155	nsec	TACS, AIDS
TATT14	$\overline{ATN}$↓ to TR1↓	155	nsec	TACS, AIDS
TATT24	$\overline{ATN}$↓ to TR2↓	155	nsec	TACS, AIDS
TDVND3-C	$\overline{DAV}$↓ to $\overline{NDAC}$↑	650	nsec	AH, CACS
TNDDV1	$\overline{NDAC}$↑ to $\overline{DAV}$↑	350	nsec	SH, STRS
TNRDV2	$\overline{NRFD}$↑ to $\overline{DAV}$↓	350	nsec	SH, T1 True
TNDDR1	$\overline{NDAC}$↑ to DREQ↑	400	nsec	SH
TDVDR3	$\overline{DAV}$↓ to DREQ↑	600	nsec	AH, LACS, $\overline{ATN}$=2.4V
TDVND2-C	$\overline{DAV}$↑ to $\overline{NDAC}$↓	350	nsec	AH,LACS
TDVNR1-C	$\overline{DAV}$↑ to $\overline{NRFD}$↑	350	nsec	AH, LACS, rdy=True
TRDNR3	$\overline{RD}$↓ to $\overline{NRFD}$↑	500	nsec	AH, LACS
TWRDI5	$\overline{WR}$ to $\overline{DIO}$ Valid	250	nsec	SH, TACS, RS = 0.4V
TWRDV2	$\overline{WR}$↑ to $\overline{DAV}$↓	830 + t_{SYNC}	nsec	High Speed Transfers Enabled, $N_F = f_C$, $t_{SYNC} = 1/2 \cdot f_C$

Notes:
1. All GPIB timings are at the pins of the 8291.

Appendix A

MODIFIED STATE DIAGRAMS

Figure A.1 presents the interface function state diagrams. It is derived from IEEE Std. state diagrams, with the following changes:

A. The 8291 supports the complete set of IEEE-488 interface functions except for the controller. These include: SH1, AH1, T5, TE5, L3, LE3, SR1, RL1, PP1, DC1, DT1, and C0.

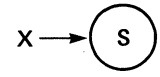
B. Addressing modes included in T,L state diagrams.

Note that in Mode 3, MSA, OSA are generated only after secondary address validity check by the microprocessor (APT interrupt).

C. In these modified state diagrams, the IEEE-488 convention of low true logic is followed. Thus, DAV is logically true at <0.8V and is equivalent to pin 36 on the 8291.

D. All remote multiline messages decoded are conditioned by ACDS. The multiplication by ACDS is not drawn to simplify the diagrams.

E. The symbol



indicates:

1. When event X occurs, the function will return to state S.
2. X overrides any other transition condition in the function.

Statement 2 simplifies the diagram, avoiding the explicit use of $\overline{X}$ to condition all transitions from S to other states.

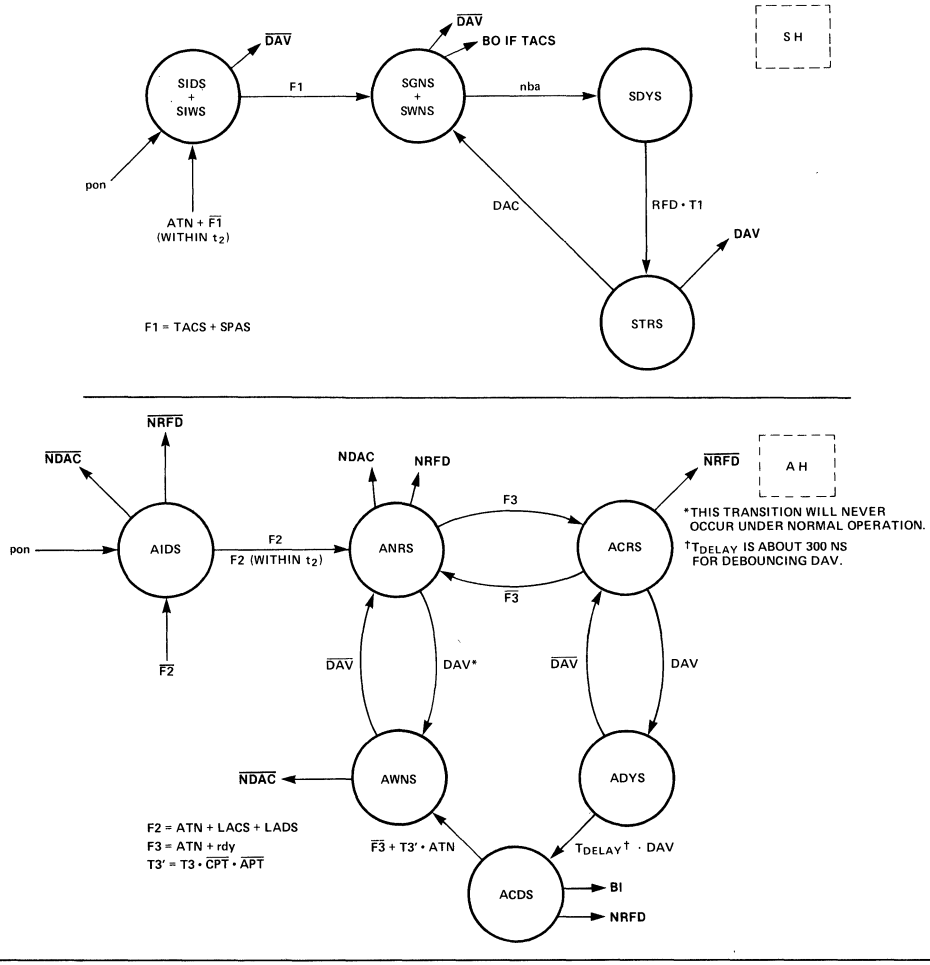


Figure A.1. 8291 State Diagrams (Continued next page)

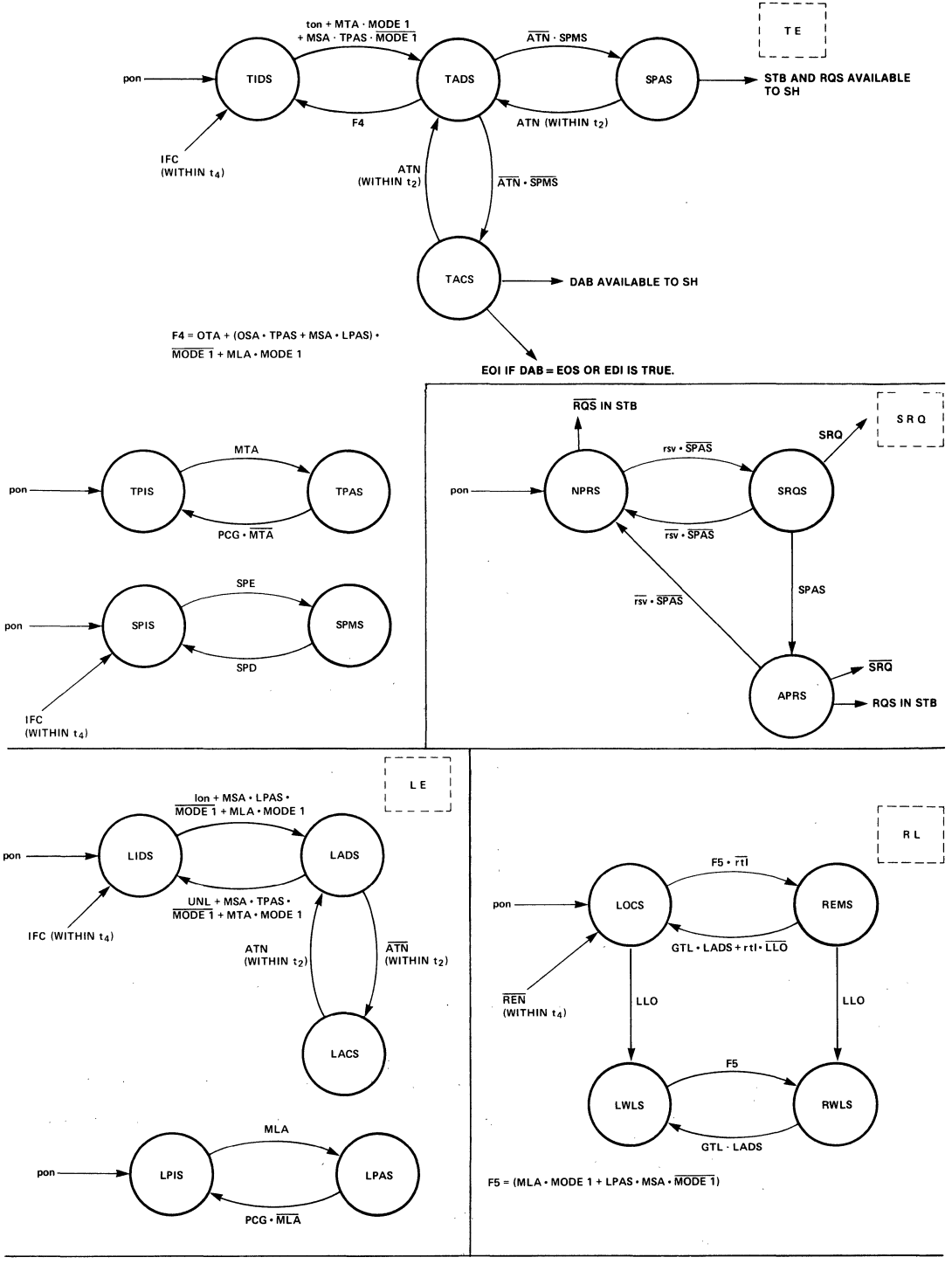


Figure A.1. 8291 State Diagrams (Continued next page)

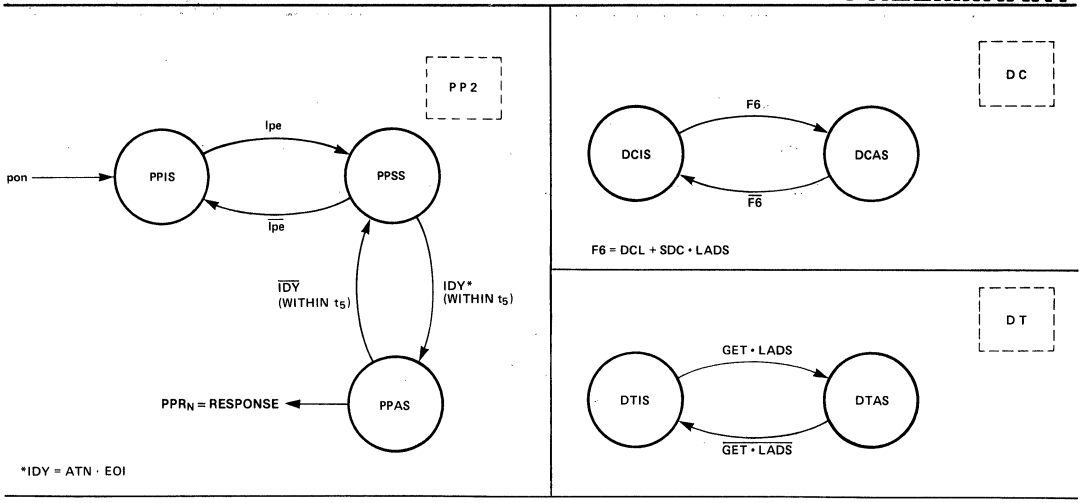


Figure A.1. 8291 State Diagrams

Appendix B

IEEE 488 TIME VALUES

Time Value Identifier*	Function (Applies to)	Description	Value
T_1	SH	Settling Time for Multiline Messages	≥ 2μs†
t_2	LC,$\overline{IC}$,SH,AH,T,L	Response to ATN	≤ 200ns
T_3	AH	Interface Message Accept Time ‡‡	> 0 δ
t_4	T,TE,L,LE,C,CE	Response to IFC or REN False	< 100μs
t_5	PP	Response to ATN+EOI	≤ 200ns
T_6	C	Parallel Poll Execution Time	≥ 2μs
T_7	C	Controller Delay to Allow Current Talker to see ATN Message	≥ 500ns
T_8	C	Length of IFC or REN False	> 100μs
T_9	C	Delay for EOI**	≥ 1.5μs††

* Time values specified by a lower case t indicate the maximum time allowed to make a state transition. Time values specified by an upper case T indicate the minimum time that a function must remain in a state before exiting.

† If three-state drivers are used on the DIO, DAV, and EOI lines, T_1 may be:
1. ≥ 1100ns
2. Or ≥ 700ns if it is known that within the controller ATN is driven by a three-state driver.
3. Or ≥ 500ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte must be sent in accordance with (1) or (2).
4. Or ≥ 350ns for all subsequent bytes following the first sent after each false transition of ATN under conditions specified in Section 5.2.3 and warning note. See IEEE Standard 488.

‡‡ Time required for interface functions to accept, not necessarily respond to interface messages.

δ Implementation independent.

** Delay required for EOI, NDAC, and NRFD signal lines to indicate valid states.

†† ≥ 600ns for three-state drivers.

Appendix C
THE THREE WIRE HANDSHAKE

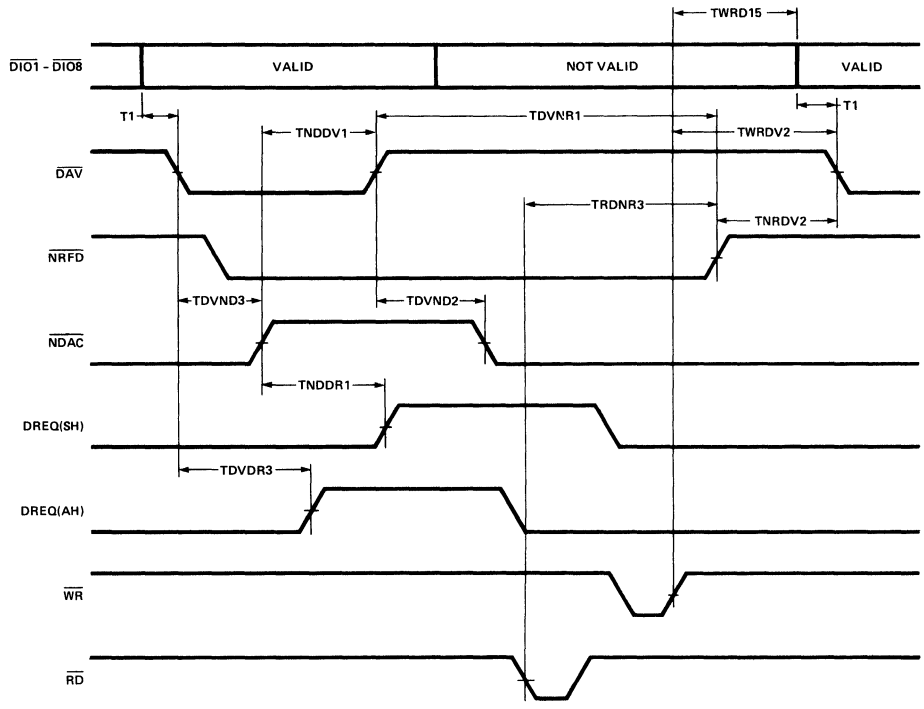
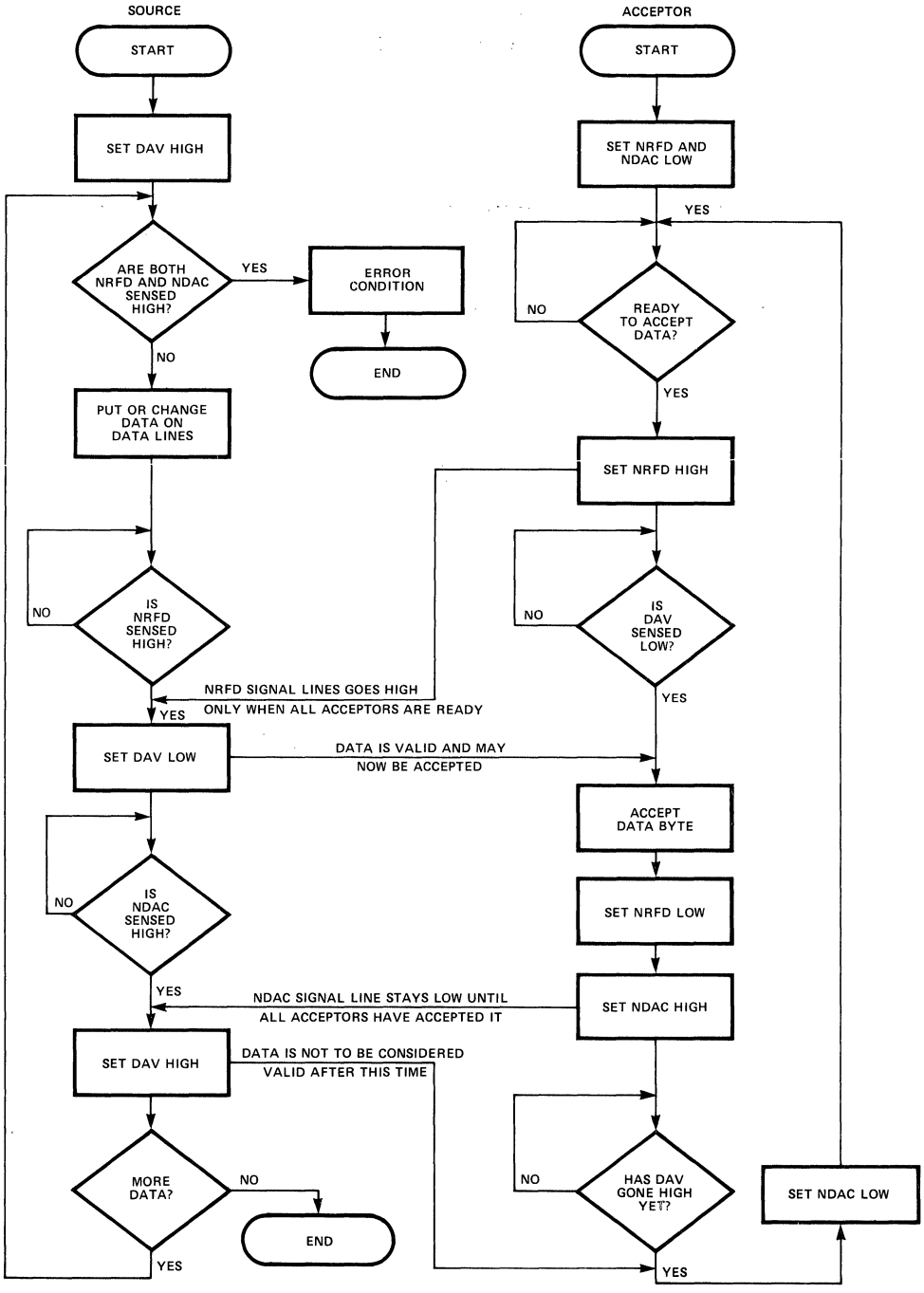


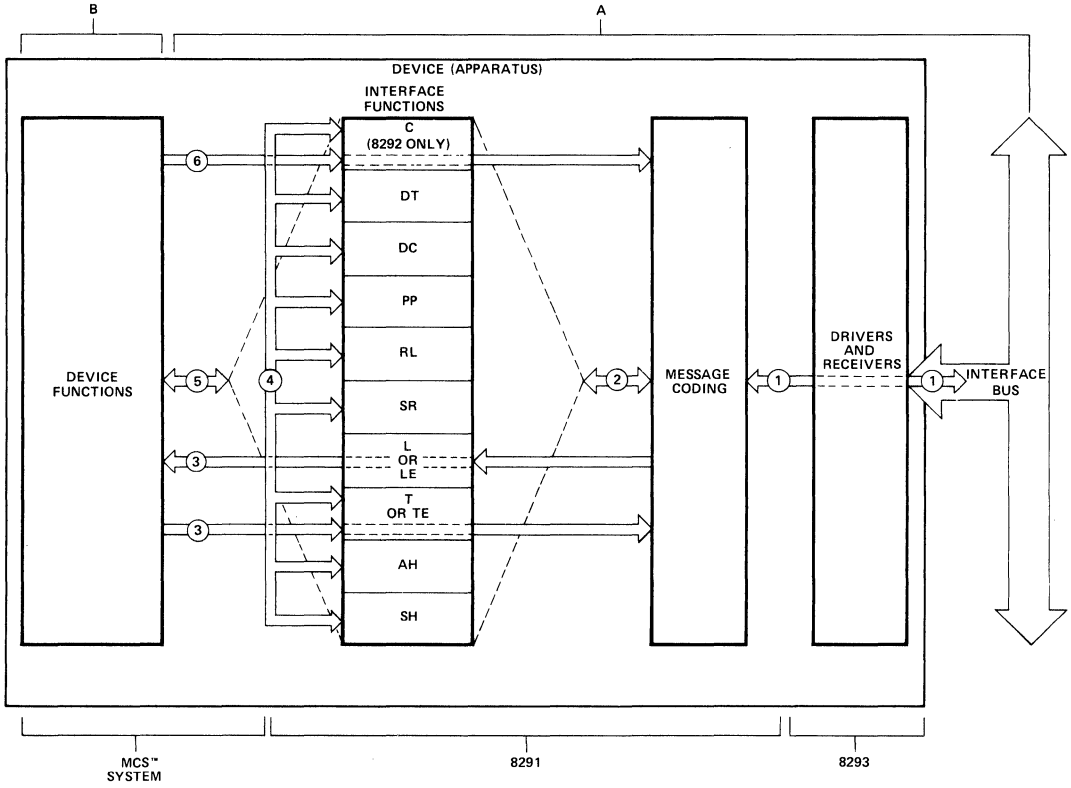
Figure C-1. 3-Wire Handshake Timing at 8291.



FLOW DIAGRAM OUTLINES SEQUENCE OF EVENTS DURING TRANSFER OF DATA BYTE. MORE THAN ONE LISTENER AT A TIME CAN ACCEPT DATA BECAUSE OF LOGICAL AND CONNECTION OF NRFD AND NDAC LINES.

Figure C.2. Handshake Flowchart.

Appendix D
FUNCTIONAL PARTITIONS



A – CAPABILITY DEFINED BY THE 488-1978 STANDARD.
B – CAPABILITY DEFINED BY THE DESIGNER.
1 – INTERFACE BUS SIGNAL LINES.
2 – REMOTE INTERFACE MESSAGES TO AND FROM INTERFACE FUNCTIONS.
3 – DEVICE DEPENDENT MESSAGES TO AND FROM DEVICE FUNCTIONS.
4 – STATE LINKAGES BETWEEN INTERFACE FUNCTIONS.
5 – LOCAL MESSAGES BETWEEN DEVICE FUNCTIONS AND INTERFACE FUNCTIONS (MESSAGES TO INTERFACE FUNCTIONS ARE DEFINED, MESSAGES FROM INTERFACE FUNCTIONS EXIST ACCORDING TO THE DESIGNER'S CHOICE).
6 – CONTROL MESSAGES (8292 ONLY).

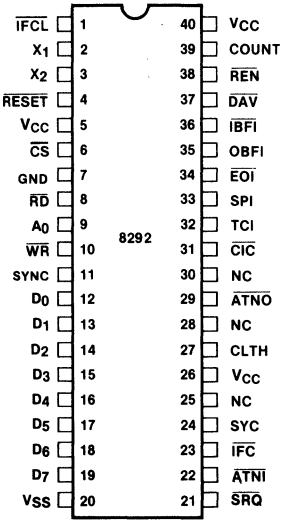
Figure D.1. Functional Partition Within a Device.

8292
GPIB CONTROLLER

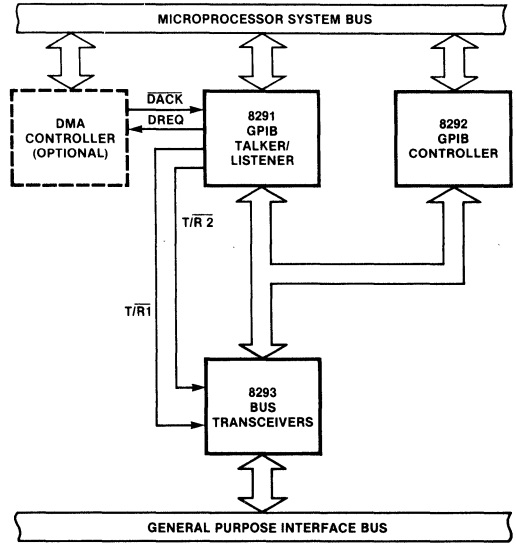
- **Complete IEEE Standard 488 Controller Function**
- **Interface Clear (IFC) Sending Capability Allows Seizure of Bus Control and/or Initialization of the Bus**
- **Responds to Service Requests (SRQ)**
- **Sends Remote Enable (REN), Allowing Instruments to Switch to Remote Control**
- **Complete Implementation of Transfer Control Protocol**
- **Synchronous Control Seizure Prevents the Destruction of Any Data Transmission in Progress**
- **Connects with the 8291 to Form a Complete IEEE Standard 488 Interface Talker/Listener/Controller**

The 8292 GPIB Controller is a microprocessor-controlled chip designed to function with the 8291 GPIB Talker/Listener to implement the full IEEE Standard 488 controller function, including transfer control protocol. The 8292 is a pre-programmed Intel® 8041A.

PIN CONFIGURATION



8291, 8292 SYSTEM DIAGRAM



00741C

PIN DESCRIPTION

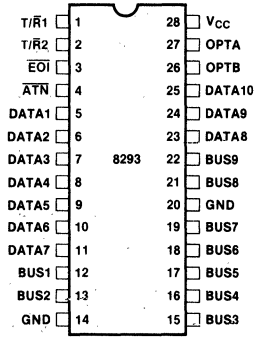
Symbol	I/O	Pin No.	Function
$\overline{IFCL}$	I	1	IFC Received (latched) — The 8292 monitors the IFC Line (when not system controller) through this pin.
X_1, X_2	I	2, 3	Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
$\overline{RESET}$	I	4	Used to initialize the chip to a known state during power on.
$\overline{CS}$	I	6	Chip Select Input — Used to select the 8292 from other devices on the common data bus.
$\overline{RD}$	I	8	I/O write input which allows the master CPU to read from the 8292.
A_0	I	9	Address Line — Used to select between the data bus and the status register during read operations and to distinguish between data and commands written into the 8292 during write operations.
$\overline{WR}$	I	10	I/O read input which allows the master CPU to write to the 8292.
SYNC	O	11	8041A instruction cycle synchronization signal; it is an output clock with a frequency of XTAL ÷ 15.
D_0-D_7	I/O	12-19	8 bidirectional lines used for communication between the central processor and the 8292's data bus buffers and status register.
V_{SS}	P.S.	7, 20	Circuit ground potential.
$\overline{SRQ}$	I	21	Service Request — One of the IEEE control lines. Sampled by the 8292 when it is controller in charge. If true, SPI interrupt to the master will be generated.
$\overline{ATNI}$	I	22	Attention In — Used by the 8292 to monitor the GPIB ATN control line. It is used during the transfer control procedure.
$\overline{IFC}$	I/O	23	Interface Clear — One of the GPIB management lines, as defined by IEEE Std. 488-1978, places all devices in a known quiescent state.
SYC	I	24	System Controller — Monitors the system controller switch.
CLTH	O	27	CLEAR LATCH Output — Used to clear the $\overline{IFCR}$ latch after being recognized by the 8292. Usually low (except after hardware Reset), it will be pulsed high when $\overline{IFCR}$ is recognized by the 8292.
$\overline{ATNO}$	O	29	Attention Out — Controls the ATN control line of the bus through external logic for tcs and tca procedures. (ATN is a GPIB control line, as defined by IEEE Std. 488-1978.)
V_{CC}	P.S.	5, 26, 40	+5V supply input. ±10%.
COUNT	I	39	Count Input — When enabled by the proper command the internal counter will count external events through this pin. High to low transition will increment the internal counter by one. The pin is sampled once per three internal instruction cycles (7.5μsec sample period when using 6 MHz XTAL). It can be used for byte counting when connected to NDAC, or for block counting when connected to the EOI.
$\overline{REN}$	O	38	The Remote Enable bus signal selects remote or local control of the device on the bus. A GPIB bus management line, as defined by IEEE Std. 488-1978.
$\overline{DAV}$	I/O	37	DAV Handshake Line — Used during parallel poll to force the 8291 to accept the parallel poll status bits. It is also used during the tcs procedure.
$\overline{IBFI}$	O	36	Input Buffer Not Full — Used to interrupt the central processor while the input buffer of the 8292 is empty. This feature is enabled and disabled by the interrupt mask register.
OBFI	O	35	Output Buffer Full — Used as an interrupt to the central processor while the output buffer of the 8292 is full. The feature can be enabled and disabled by the interrupt mask register.
$\overline{EOI2}$	I/O	34	End Or Identify — One of the GPIB management lines, as defined by IEEE Std. 488-1978. Used with ATN as Identify Message during parallel poll.
SPI	O	33	Special Interrupt — Used as an interrupt on events not initiated by the central processor.
TCI	O	32	Task Complete Interrupt — Interrupt to the control processor used to indicate that the task requested was completed by the 8292 and the information requested is ready in the data bus buffer.
CIC	O	31	Controller In Charge — Controls the S/R input of the SRQ bus transceiver. It can also be used to indicate that the 8292 is in charge of the GPIB bus.

8293
GPIB TRANSCEIVER

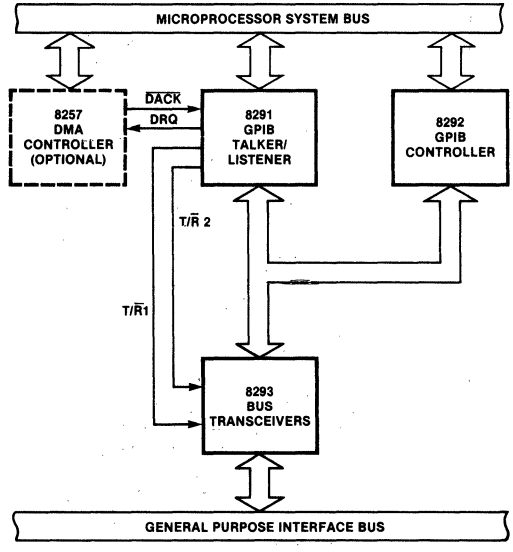
- Nine Open-collector or Three-state Line Drivers
- 48 mA Sink Current Capability on Each Line Driver
- Nine Schmitt-type Line Receivers
- High Capacitance Load Drive Capability
- Single 5V Power Supply
- 28-Pin Package
- Low Power HMOS Design
- On-chip Decoder for Mode Configuration
- Power Up/Power Down Protection to Prevent Disrupting the IEEE Bus
- Connects with the 8291 and 8292 to Form an IEEE Standard 488 Interface Talker/Listener/Controller with no Additional Components
- Only Two 8293's Required per GPIB Interface
- On-Chip IEEE-488 Bus Terminations

The Intel® 8293 GPIB Transceiver is a high current, non-inverting buffer chip designed to interface the 8291 GPIB Talker/Listener or the 8292 GPIB Controller with the 8291 to the IEEE Standard 488-1978 Instrumentation Interface Bus. Each GPIB interface would contain two 8293 Bus Transceivers. In addition, the 8293 can also be used as a general purpose bus driver.

PIN CONFIGURATION



8291, 8292, 8293 SYSTEM DIAGRAM



PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
BUS1-BUS9	I/O	12, 13, 15-19, 21, 22	These are the IEEE-488 bus interface driver/receivers. Using the mode select pins, they can be configured differently to allow direct connections between the 8291 GPIB Talker/Listener and the 8292 GPIB Controller.
DATA1-DATA10	I/O	5-11, 23-25	These are the pins to be connected to the 8291 and 8292 to interface with the GPIB bus. Their use is programmed by the two mode select pins, OPTA and OPTB. All these pins are TTL compatible.
T/$\overline{R}$1	I	1	Transmit receive 1; this pin controls the direction for NDAC, NRFD, DAV, and DIO1-DIO8. Input is TTL compatible.
T/$\overline{R}$2	I	2	Transmit receive 2; this pin controls the direction for EOI. Input is TTL compatible.
$\overline{EOI}$	I/O	3	End or Identify; this pin indicates the end of a multiple byte transfer or, in conjunction with ATN, addresses the device during a polling sequence. It connects to the 8291 and is switched between transmit and receive by T/$\overline{R}$2. This pin is TTL compatible.
$\overline{ATN}$	O	4	Attention; this pin is used by the 8291 to monitor the GPIB ATN control line. It specifies how data on the DIO lines is to be interpreted. This output is TTL compatible.
OPTA	I	27	These two pins are to control the function of the 8293. A truth table of how this programs the various modes is in Table 1.
OPTB	I	26	
V_{CC}	P.S.	28	Positive power supply (5V ± 10%).
GND	P.S.	14, 20	Circuit ground potential.

Table 1. 8293 Mode Selection Pin Mapping

Pin Name	Pin No.	IEEE Implementation Name			
		Mode 0	Mode 1	Mode 2	Mode 3
OPTA	27	0	1	0	1
OPTB	26	0	0	1	1
DATA1	5	$\overline{IFC}$	$\overline{DIO8}$	$\overline{IFC}$	$\overline{DIO8}$
BUS1	12	IFC*	DIO8*	IFC*	DIO8*
DATA2	6	$\overline{REN}$	$\overline{DIO7}$	$\overline{REN}$	$\overline{DIO7}$
BUS2	13	REN*	DIO7*	REN*	DIO7*
DATA3	7	NC	$\overline{DIO6}$	$\overline{EOI2}$	$\overline{DIO6}$
BUS3	15	EOI*	DIO6*	EOI*	DIO6*
DATA4	8	$\overline{SRQ}$	$\overline{DIO5}$	$\overline{SRQ}$	$\overline{DIO5}$
BUS4	16	SRQ*	DIO5*	SRQ*	DIO5*
DATA5	9	$\overline{NRFD}$	$\overline{DIO4}$	$\overline{NRFD}$	$\overline{DIO4}$
BUS5	17	NRFD*	DIO4*	NRFD*	DIO4*
DATA6	10	$\overline{NDAC}$	$\overline{DIO3}$	$\overline{NDAC}$	$\overline{DIO3}$
BUS6	18	NDAC*	DIO3*	NDAC*	DIO3*
DATA7	11	T/$\overline{R}$IO1	NC	$\overline{ATNI}$	$\overline{ATNO}$
DATA8	23	T/$\overline{R}$IO2	$\overline{DIO2}$	$\overline{ATNO}$	$\overline{DIO2}$
BUS7	19	ATN*	DIO2*	ATN*	DIO2*
DATA9	24	$\overline{GIO1}$	$\overline{DAV}$	$\overline{CIC}$	$\overline{DAV}$
BUS8	21	GIO1*	DAV*	CLTH	DAV*
DATA10	25	$\overline{GIO2}$	$\overline{DIO1}$	$\overline{IFCL}$	$\overline{DIO1}$
BUS9	22	GIO2*	DIO1*	SYC	DIO1*
T/$\overline{R}$1	1	T/$\overline{R}$1	T/$\overline{R}$1	T/$\overline{R}$1	T/$\overline{R}$1
T/$\overline{R}$2	2	T/$\overline{R}$2	NC	T/$\overline{R}$2	$\overline{IFCL}$
$\overline{EOI}$	3	$\overline{EOI}$	$\overline{EOI}$	$\overline{EOI}$	$\overline{EOI}$
$\overline{ATN}$	4	$\overline{ATN}$	$\overline{ATN}$	$\overline{ATN}$	$\overline{ATN}$

***Note:** These pins are the IEEE-488 bus non-inverting driver/receivers. They include all the bus terminations required by the Standard and may be connected directly to the GPIB bus connector.

GENERAL DESCRIPTION

The 8293 is a bidirectional transceiver. It was designed to interface the Intel 8291 GPIB Talker/Listener and the Intel® 8292 GPIB Controller to the IEEE Standard 488-1978. Instrumentation Bus (also referred to as the GPIB Bus). The Intel GPIB Bus Transceiver meets or exceeds all of the electrical specifications defined in the IEEE Standard 488-1978, Section 3.3-3.5, including the required bus termination specifications.

The 8293 can be hardware programmed to one of four modes of operation. These modes allow the 8293 to be configured to support both a Talker/Listener/Controller environment and Talker/Listener environment. In addition, the 8293 can be used as a general purpose three-state (push-pull) or open-collector bus transceiver with nine receiver/drivers. Two modes are used to support a Talker/Listener environment (see Figure 1), and to support a Talker/Listener/Controller environment (see Figure 2). Mode 1 is the general purpose mode.

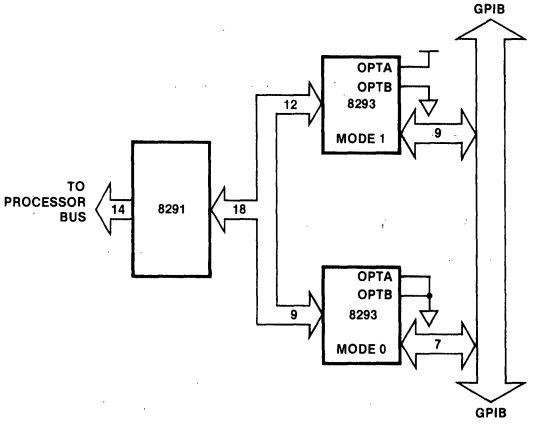


Figure 1. Talker/Listener Configuration

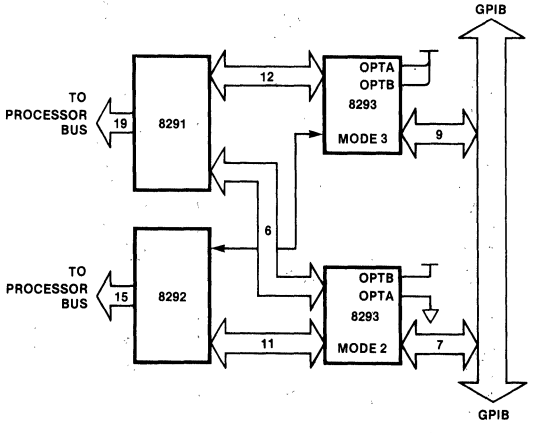


Figure 2. Talker/Listener/Controller Configuration

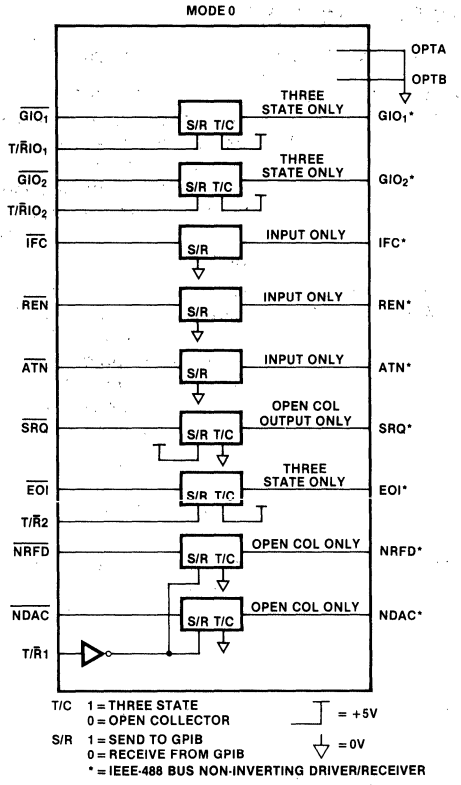


Figure 3. Talker/Listener Control Configuration

MODE 0 PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
T/R̄1	I	1	Transmit receive 1; direction control for NDAC and NRFD. If T/R̄1 is high, then NDAC* and NRFD* are receiving. Input is TTL compatible.
NDAC	I/O	10	Not Data Accepted; processor GPIB bus handshake control line; used to indicate the condition of acceptance of data by device(s). It is TTL compatible.
NDAC*	I/O	18	Not Data Accepted; IEEE GPIB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When an output, it is an open-collector driver with 48 mA sinking capability.
NRFD	I/O	9	Not Ready For Data; processor GPIB handshake control line; used to indicate the condition of readiness of device(s) to accept data. This pin is TTL compatible.

Symbol	I/O	Pin No.	Function
NRFD*	I/O	17	Not Ready For Data; IEEE GPIB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When an output, it is an open-collector driver with a 48 mA current sinking capability.
T/$\overline{R}$2	I	2	Transmit receive 2; direction control for EOI. If T/$\overline{R}$2 is high, EOI* is sending. Input is TTL compatible.
$\overline{EOI}$	I/O	3	End or Identify; processor GPIB bus control line; is used by a talker by indicate the end of a multiple byte transfer. This pin is TTL compatible.
EOI*	I/O	15	End or Identify; IEEE GPIB bus control line; is used by a talker to indicate the end of a multiple byte transfer. This pin is a three-state (push-pull) driver capable of sinking 48 mA and a TTL compatible receiver with hysteresis.
$\overline{SRQ}$	I	8	Service Request; processor GPIB bus control line; used by a device to indicate the need for service and to request an interruption of the current sequence of events on the GPIB. It is a TTL compatible input.
SRQ*	O	16	Service Request; IEEE GPIB bus control line; it is an open collector driver capable of sinking 48 mA.
$\overline{REN}$	O	6	Remote Enable; processor GPIB bus control line; used by a controller (in conjunction with other messages) to select between two alternate sources of device programming data (remote or local control). This output is TTL compatible.
REN*	I	13	Remote Enable; IEEE GPIB bus control line. This input is a TTL compatible Schmitt-trigger.
$\overline{ATN}$	O	4	Attention; processor GPIB bus control line; used by the 8291 to determine how data on the DIO signal lines are to be interpreted. This is a TTL compatible output.
ATN*	I	19	Attention; IEEE GPIB bus control line; this input is a TTL compatible Schmitt-trigger.
$\overline{IFC}$	O	5	Interface Clear; processor GPIB bus control line; used by a controller to place the interface system into a known quiescent state. It is a TTL compatible output.
IFC*	I	12	Interface Clear; IEEE GPIB bus control line. This input is a TTL compatible Schmitt-trigger.
T/$\overline{R}$IO1 T/$\overline{R}$IO2	I I	11 23	Transmit receive General IO; direction control for the two spare transceivers. Input is TTL compatible.
$\overline{GIO1}$ $\overline{GIO2}$	I/O I/O	24 25	General IO; this is the TTL side of the two spare transceivers. These pins are TTL compatible.
GIO1* GIO2*	I/O I/O	21 22	General IO; these are spare three-state (push-pull) drivers/ Schmitt-trigger receivers. The drivers can sink 48 mA.

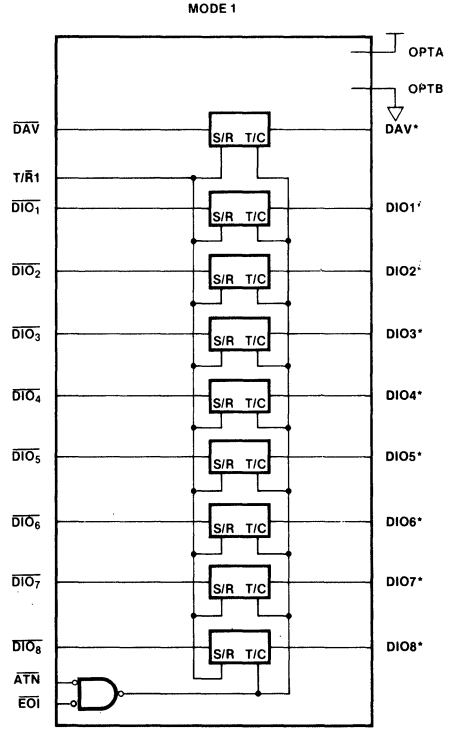


Figure 4. Talker/Listener Data Configuration

MODE 1 PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
T/$\overline{R}$1	I	1	Transmit receive 1; controls the direction for DAV and the DIO lines. If T/$\overline{R}$1 is high, then all these lines are sending information to the IEEE GPIB lines. This input is TTL compatible.
$\overline{EOI}$	I	3	End of Sequence and Attention; processor GPIB control lines. These two control signals are ANDed together to determine whether all the transceivers in the 8293 are three-state (push-pull) or open-collector. When both signals are low (true), then the controller is performing a parallel poll and the transceivers are all open-collector. These inputs are TTL compatible.
$\overline{ATN}$	I	4	
$\overline{DAV}$	I/O	24	Data Valid; processor GPIB bus handshake control line; used to indicate the condition (availability and validity) of information on the DIO signals. It is TTL compatible.
DAV*	I/O	21	Data Valid; IEEE GPIB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When DAV* is an output, it can sink 48 mA.
$\overline{DIO1}$-$\overline{DIO8}$	I/O	25, 23, 10, 9, 8, 7, 6, 5	Data Input/Output; processor GPIB bus data lines; used to carry message and data bytes in a bit-parallel byte-serial form controlled by the three handshake signals. These lines are TTL compatible.
DIO1* DIO8*	I/O	22, 19, 18, 17, 16, 15, 13, 12	Data Input/Output; IEEE GPIB bus data lines. They are TTL compatible Schmitt-triggers when used for input and can sink 48 mA when used for output. See $\overline{ATN}$ and $\overline{EOI}$ description for output mode.

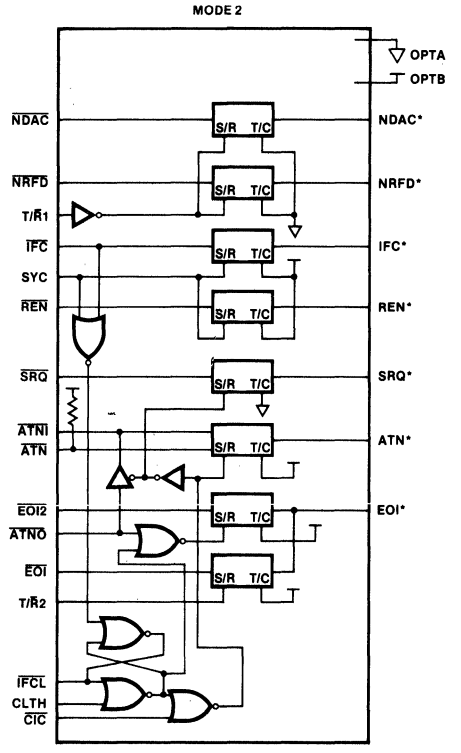


Figure 5. Talker/Listener/Controller Control Configuration

MODE 2 PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
T/$\overline{R}$1	I	1	Transmit receive 1; direction control for NDAC and NRFD. If T/$\overline{R}$1 is high, then NDAC and NRFD are receiving. Input is TTL compatible.
$\overline{NDAC}$	I/O	10	Not Data Accepted; processor GPIB bus handshake control line; used to indicate the condition of acceptance of data by device(s). This pin is TTL compatible.
NDAC*	I/O	18	Not Data Accepted; IEEE GPIB bus handshake control line. It is a TTL compatible Schmitt-trigger when used for input and an open-collector driver with a 48 mA current sink capability when used for output.

Symbol	I/O	Pin No.	Function
$\overline{\text{NRFD}}$	I/O	9	Not Ready For Data; processor GPIB bus handshake control line; used to indicate the condition of readiness of device(s) to accept data. This pin is TTL compatible.
NRFD*	I/O	17	Not Ready For Data; IEEE GPIB bus handshake control line. It is a TTL compatible Schmitt-trigger when used for input and an open-collector driver with a 48 mA current sink capability when used for output.
SYC	I	22	System Controller; used to monitor the system controller switch and control the direction for IFC and REN. This pin is a TTL compatible input.
$\overline{\text{REN}}$	I/O	6	Remote Enable; processor GPIB control line; used by the active controller (in conjunction with other messages) to select between two alternate sources of device programming data (remote or local control). This pin is TTL compatible.
REN*	I/O	13	Remote Enable; IEEE GPIB bus control line. When used as an input, this is a TTL compatible Schmitt-trigger. When an output, it is a three-state driver with a 48 mA current sinking capability.
$\overline{\text{IFC}}$	I/O	5	Interface Clear; processor GPIB bus control line; used by the active controller to place the interface system into a known quiescent state. This pin is TTL compatible.
IFC*	I/O	12	Interface Clear; IEEE GPIB bus control line. This is a TTL compatible Schmitt-trigger when used for input and a three-state driver capable of sinking 48 mA current when used for output.
$\overline{\text{CIC}}$	I	24	Controller in Charge; used to control the direction of the SRQ and to indicate that the 8292 is in charge of the bus. $\overline{\text{CIC}}$ is a TTL compatible input.
CLTH	I	21	Clear Latch; used to clear the IFC Received latch after it has been recognized by the 8292. Normally low (except after a hardware reset), it will be pulsed low when IFC Received is recognized by the 8292. This input is TTL compatible.
$\overline{\text{IFCL}}$	O	25	IFC Received Latched; the 8292 monitors the IFC line when it is not the active controller through this pin.
$\overline{\text{SRQ}}$	I/O	8	Service Request; processor GPIB control line; indicates the need for attention and requests the active controller to interrupt the current sequence of events on the GPIB bus. This pin is TTL compatible.
SRQ*	I/O	16	Service Request; IEEE GPIB bus control line. When used as an input, this pin is a TTL compatible Schmitt-trigger. When used as an output, it is an open-collector driver with a 48 mA current sinking capability.
T/$\overline{\text{R}}$2	I	2	Transmit receive 2; controls the direction for EOI. This input is TTL compatible.
$\overline{\text{ATNO}}$	I	23	Attention Out; processor GPIB bus control line; used by the 8292 for ATN control of the IEEE bus during "take control synchronously" operations. A low on this input causes ATN to be asserted if $\overline{\text{CIC}}$ indicates that this 8292 is in charge. $\overline{\text{ATNO}}$ is a TTL compatible input.
$\overline{\text{ATNI}}$	O	11	Attention In; processor GPIB bus control line; used by the 8292 to monitor the ATN line. This output is TTL compatible.
$\overline{\text{ATN}}$	O	4	Attention; processor GPIB bus control live; used by the 8292 to monitor the ATN line. This output is TTL compatible.
ATN*	I/O	19	Attention; IEEE GPIB bus control line; used by a controller to specify how data on the DIO signal lines are to be interpreted and which devices must respond to data. When used as an output, this pin is a three-state driver capable of sinking 48 mA current. As an input, it is a TTL compatible Schmitt-trigger.
$\overline{\text{EOI2}}$	I/O	7	End or Identify 2; processor GPIB bus control line; used in conjunction with ATN by the active controller (the 8292) to execute a polling sequence. This pin is TTL compatible.

Symbol	I/O	Pin No.	Function
$\overline{EOI}$	I/O	3	End or Identify; processor GPIB bus control line; used by a talker to indicate the end of a multiple byte transfer sequence. This pin is TTL compatible.
EOI*	I/O	15	End or Identify; IEEE GPIB bus control line; used by a talker to indicate the end of a multiple byte transfer sequence or, by a controller in conjunction with ATN, to execute a polling sequence. When an output, this pin can sink 48 mA current. When an input, it is a TTL compatible Schmitt-trigger.

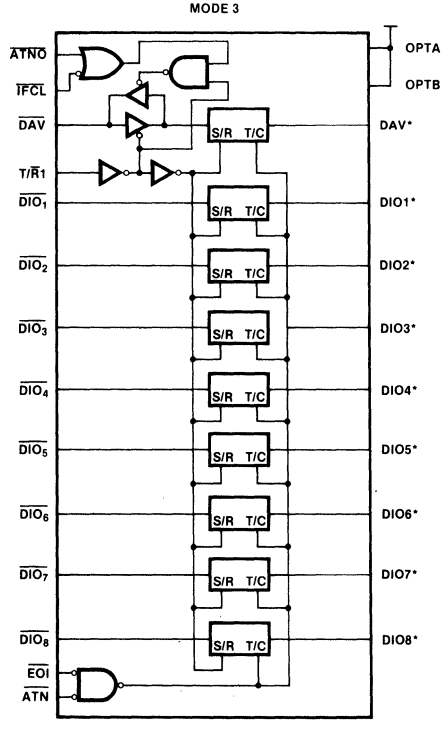


Figure 6. Talker/Listener/Controller Data Configuration

MODE 3 PIN DESCRIPTION

Symbol	I/O	Pin No.	Function
T/$\overline{R}$1	I	1	Transmit receive 1; controls the direction for DAV and the DIO lines. If T/$\overline{R}$1 is high, then all these lines are sending information to the IEEE GPIB lines. This input is TTL compatible.
$\overline{EOI}$	I	3	End of Sequence and Attention; processor GPIB control lines. These two control lines are ANDed together to determine whether all the transceivers in the 8293 are push-pull or open-collector. When both signals are low (true), then the controller is performing a parallel poll and the transceivers are all open-collector. These inputs are TTL compatible.
$\overline{ATN}$	I	4	
$\overline{ATNO}$	I	23	Attention Out; processor GPIB control line; used by the 8292 during "take control synchronously" operations. This pin is TTL compatible.
$\overline{IFCL}$	I	2	Interface Clean Latched; used to make DAV received after the system controller asserts IFC. This input is TTL compatible.
$\overline{DAV}$	I/O	24	Data Valid; processor GPIB handshake control line; used to indicate the condition (availability and validity) of information on the DIO signals. This pin is TTL compatible.
DAV*	I/O	21	Data Valid; IEEE GPIB handshake control line. When an input, this pin is a TTL compatible Schmitt-trigger. When DAV* is an output, it can sink 48 mA.
$\overline{DIO1}$-$\overline{DIO8}$	I/O	25, 23, 10, 9, 8, 7, 6, 5	Data Input/Output; processor GPIB bus data lines; used to carry message and data bytes in a bit-parallel byte-serial form controlled by the three handshake signals. These lines are TTL compatible.
DIO1*-DIO8*	I/O	22, 19, 18, 17, 16, 15, 13, 12	Data Input/Output; IEEE GPIB bus data lines. They are TTL compatible Schmitt-triggers when used for input and can sink 48 mA when used for output.

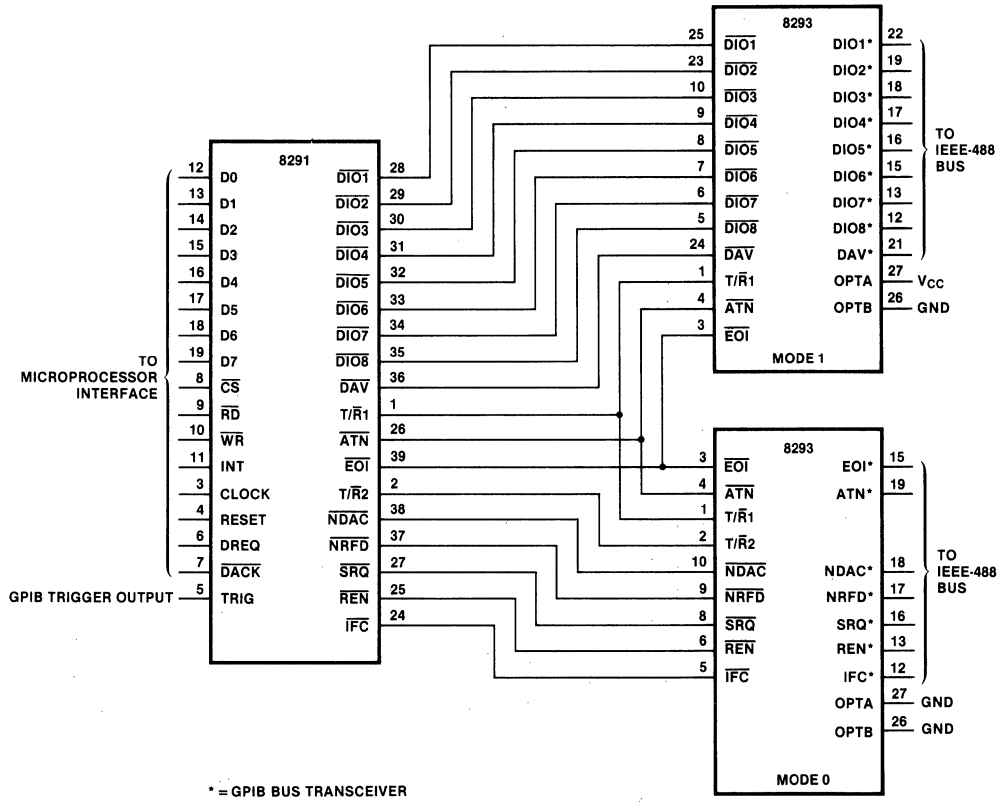


Figure 7. 8291 and 8293 System Configuration

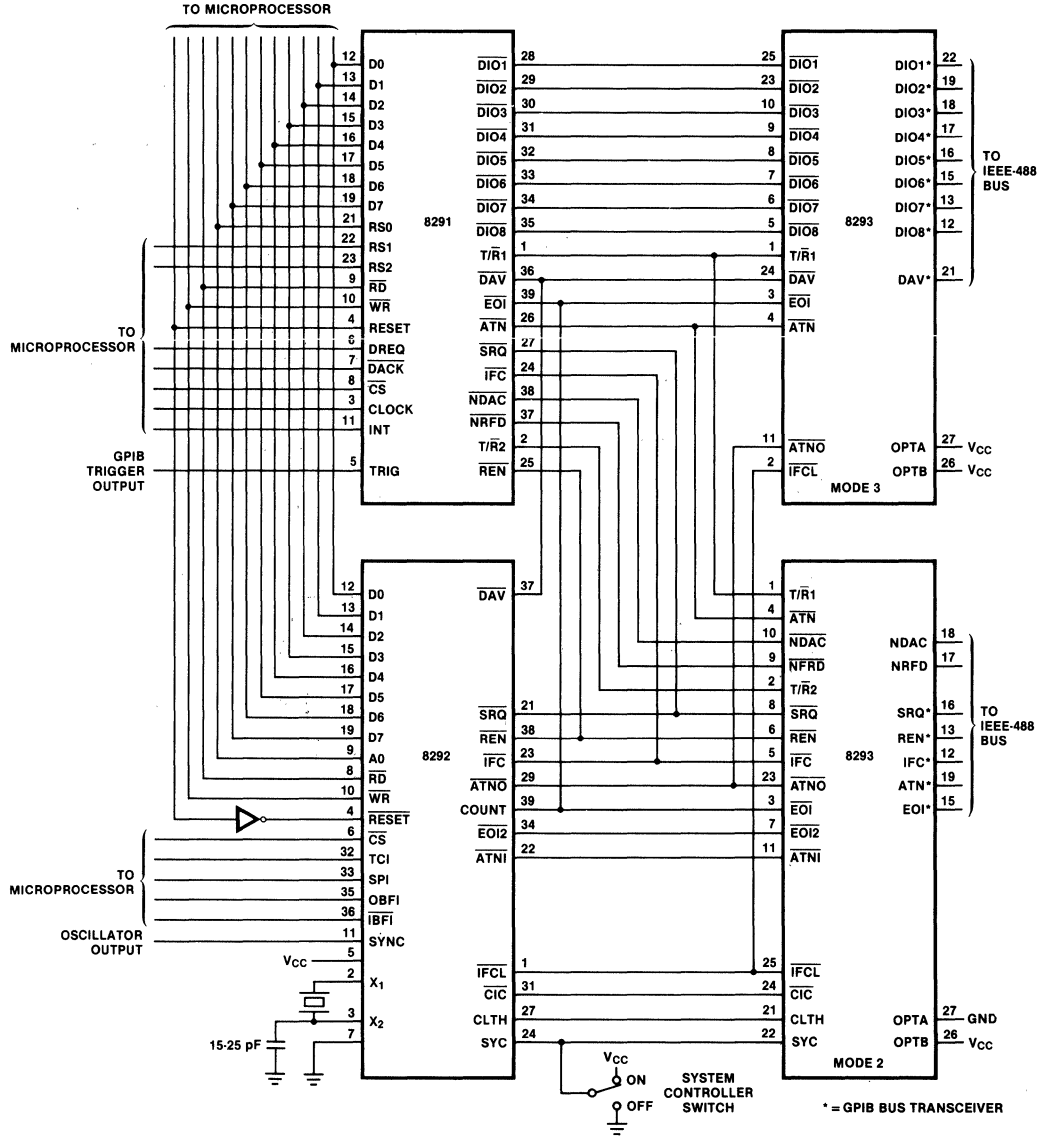


Figure 8. 8291, 8292, and 8293 System Configuration

Absolute Maximum Ratings*

Ambient Temperature Under Bias.....0°C to 70°C
Storage Temperature.....−65°C to +150°C
Voltage on any Pin with
Respect to Ground.....−1.0V to +7V
Power Dissipation.....1 Watt

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

D.C. and Operating Characteristics

$T_A = 0°C$ to 70°C; $V_{CC} = 5.0V \pm 10\%$; GND = 0V

SYMBOL	PARAMETER	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.	MAX.		
V_{IL1}	Input Low Voltage (GPIB Bus Pins)			0.8	V	
V_{IL2}	Input Low Voltage (Option Pins)	−0.1		0.1	V	
V_{IL3}	Input Low Voltage (All Others)			0.8	V	
V_{IH1}	Input High Voltage (GPIB Bus Pins)	2.0			V	
V_{IH2}	Input High Voltage (Option Pins)	4.5		5.5	V	
V_{IH3}	Input High Voltage (All Others)	2.0			V	
V_{OL1}	Output Low Voltage (GPIB Bus Pins)			0.5	V	$I_{OL} = 48$ mA
V_{OL2}	Output Low Voltage (All Others)			0.5	V	$I_{OL} = 16$ mA
V_{OH1}	Output High Voltage (GPIB Bus Pins)	2.4			V	$I_{OH} = -5.2$ mA
V_{OH2}	Output High Voltage (All Others)	2.4			V	$I_{OH} = -400\ \mu A$
V_{IH4}	Receiver Input Hysteresis	400	600		mV	
V_{IT}	Receiver Input Threshold High to Low	0.8	1.0		V	
	Receiver Input Threshold Low to High		1.6	2.0		
I_{LI1}	Low Input Load Current (GPIB Bus Pins)	−3.2		0.0	mA	$V_{IL} = 0.8V$
I_{LI2}	Low Input Load Current (All Others)			10	μA	$V_{IL} = 0.8V$
I_{PD}	Bus Power Down Leakage Current			10	μA	$V_{CC} = 0V$
I_{CC}	Power Supply Current			100	mA	

Capacitance

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance		5	10	pF	$V_{IN} = V_{CC}$
C_{OUT}	Output Capacitance		10	20	pF	$V_{OUT} = V_{CC}$

A.C. Characteristics

$T_A = 0°C$ to $70°C$; $V_{CC} = 5.0V \pm 10\%$; GND = 0V

SYMBOL	PARAMETER	TYP.*	MAX.	UNITS
t_{PLH1}	Driver Propagation Delay (Low to High)	20	35	ns
t_{PHL1}	Driver Propagation Delay (High to Low)	17	30	ns
t_{PLH2}	Receiver Propagation Delay (Low to High)	22	35	ns
t_{PHL2}	Receiver Propagation Delay (High to Low)	18	30	ns
t_{PHZ1}	Driver Enable Delay (High to 3-State)	20	35	ns
t_{PZH1}	Driver Enable Delay (3-State to High)	15	30	ns
t_{PLZ1}	Driver Enable Delay (Low to 3-State)	20	35	ns
t_{PZL1}	Driver Enable Delay (3-State to Low)	15	30	ns
t_{PHZ2}	Receiver Enable Delay (High to 3-State)	25	40	ns
t_{PZH2}	Receiver Enable Delay (3-State to High)	20	35	ns
t_{PLZ2}	Receiver Enable Delay (Low to 3-State)	25	40	ns
t_{PZL2}	Receiver Enable Delay (3-State to Low)	20	35	ns

*Typical @ $T_A = 25°C$.

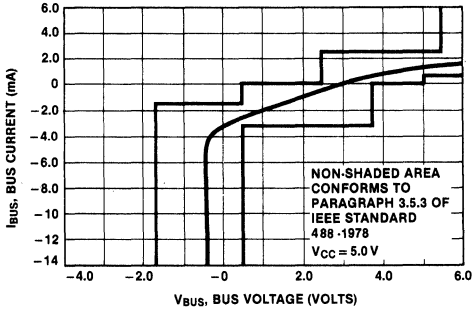


Figure 9. Typical Bus Load Line

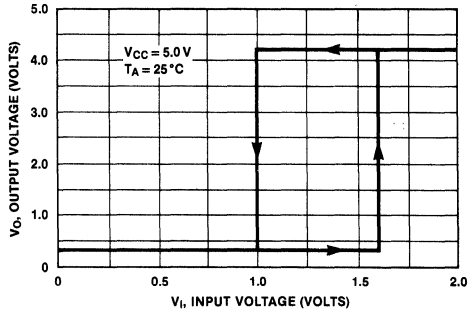


Figure 10. Typical Receiver Hysteresis Characteristics

OUTPUT LOADING TEST CIRCUITS

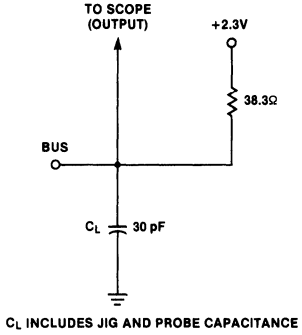


Figure 11. Data Input to Bus Output (Driver)

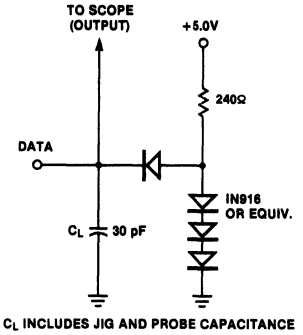


Figure 12. Bus Input to Data Output (Receiver)

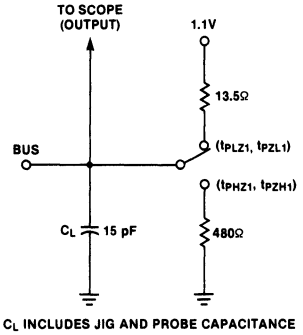


Figure 13. Send/Receive Input to Bus Output (Driver)

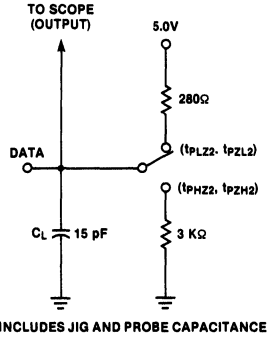
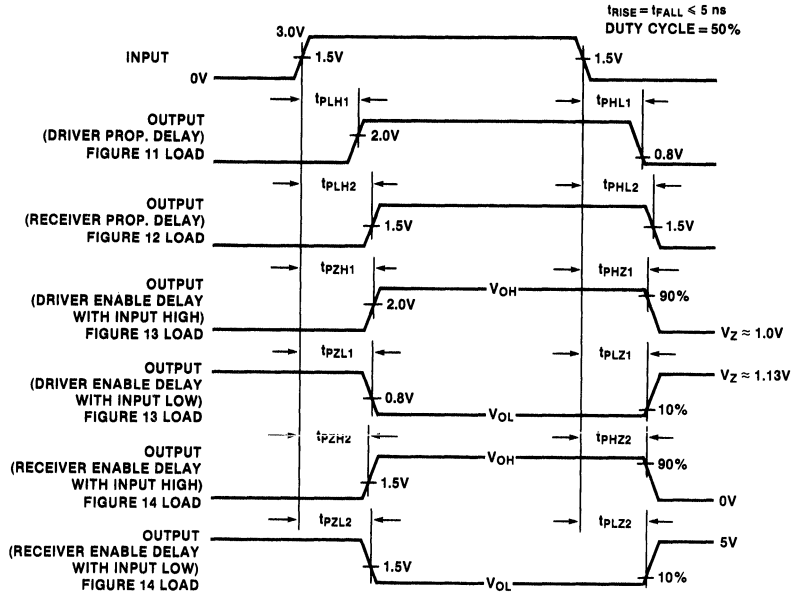


Figure 14. Send/Receive Input to Data Output (Receiver)

8293 WAVEFORMS



8294
DATA ENCRYPTION UNIT

- **Certified by National Bureau of Standards**
- **80 Byte/Sec Data Conversion Rate**
- **64-Bit Data Encryption Using 56-Bit Key**
- **DMA Interface**
- **3 Interrupt Outputs to Aid in Loading and Unloading Data**
- **7-Bit User Output Port**
- **Single 5V ± 10% Power Supply**
- **Peripheral to MCS-86™, MCS-85™, MCS-80™ and MCS-48™ Processors**
- **Implements Federal Information Processing Data Encryption Standard**
- **Encrypt and Decrypt Modes Available**

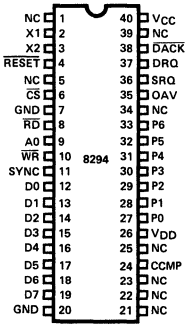
DESCRIPTION

The Intel® 8294 Data Encryption Unit (DEU) is a microprocessor peripheral device designed to encrypt and decrypt 64-bit blocks of data using the algorithm specified in the Federal Information Processing Data Encryption Standard. The DEU operates on 64-bit text words using a 56-bit user-specified key to produce 64-bit cipher words. The operation is reversible: if the cipher word is operated upon, the original text word is produced. The algorithm itself is permanently contained in the 8294; however, the 56-bit key is user-defined and may be changed at any time.

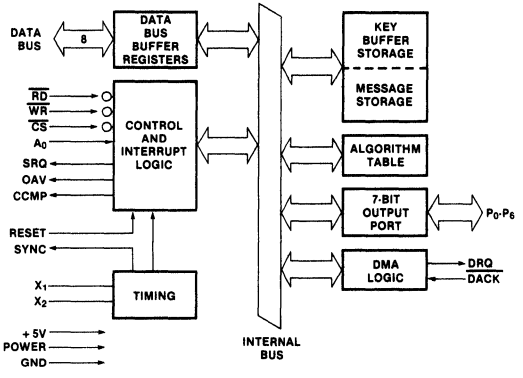
The 56-bit key and 64-bit message data are transferred to and from the 8294 in 8-bit bytes by way of the system data bus. A DMA interface and three interrupt outputs are available to minimize software overhead associated with data transfer. Also, by using the DMA interface two or more DEUs may be operated in parallel to achieve effective system conversion rates which are virtually any multiple of 80 bytes/second. The 8294 also has a 7-bit TTL compatible output port for user-specified functions.

Because the 8294 implements the NBS encryption algorithm it can be used in a variety of Electronic Funds Transfer applications as well as other electronic banking and data handling applications where data must be encrypted.

PIN CONFIGURATION



BLOCK DIAGRAM



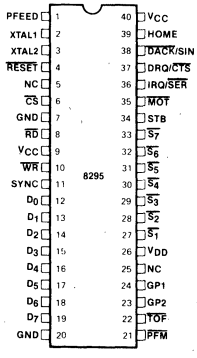
8295
DOT MATRIX PRINTER CONTROLLER

- **Interfaces Dot Matrix Printers to MCS-48™, MCS-80/85™, MCS-86™ Systems**
- **40 Character Buffer On Chip**
- **Serial or Parallel Communication with Host**
- **DMA Transfer Capability**
- **Programmable Character Density (10 or 12 Chararcters/Inch)**
- **Programmable Print Intensity**
- **Single or Double Width Printing**
- **Programmable Multiple Line Feeds**
- **3 Tabulations**
- **2 General Purpose Outputs**

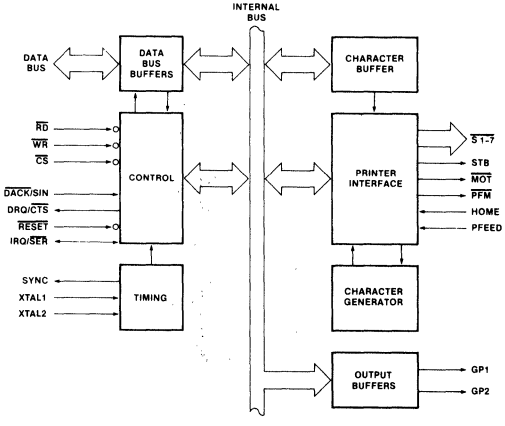
The Intel® 8295 Dot Matrix Printer Controller provides an interface for microprocessors to the LRC 7040 Series dot matrix impact printers. It may also be used as an interface to other similar printers.

The chip may be used in a serial or parallel communication mode with the host processor. In parallel mode, data transfers are based on polling, interrupts, or DMA. Furthermore, it provides internal buffering of up to 40 characters and contains a 7 x 7 matrix character generator accommodating 64 ASCII characters.

PIN CONFIGURATION



BLOCK DIAGRAM



8041A/8641A/8741A
UNIVERSAL PERIPHERAL INTERFACE
8-BIT MICROCOMPUTER

- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **One 8-Bit Status and Two Data Registers for Asynchronous Slave-to-Master Interface**
- **DMA, Interrupt, or Polled Operation Supported**
- **1024 x 8 ROM/EPROM, 64 x 8 RAM, 8-Bit Timer/Counter, 18 Programmable I/O Pins**
- **Fully Compatible with MCS-48™, MCS-80™, MCS-85™, and MCS-86™ Microprocessor Families**
- **Interchangeable ROM and EPROM Versions**
- **3.6 MHz 8741A-8 Available**
- **Expandable I/O**
- **RAM Power-Down Capability**
- **Over 90 Instructions: 70% Single Byte**
- **Single 5V Supply**

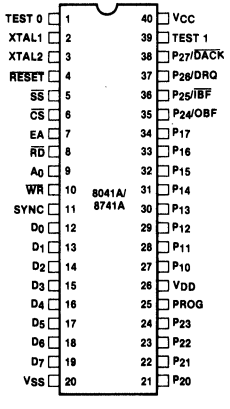
The Intel® 8041A/8741A is a general purpose, programmable interface device designed for use with a variety of 8-bit microprocessor systems. It contains a low cost microcomputer with program memory, data memory, 8-bit CPU, I/O ports, timer/counter, and clock in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in MCS-48™, MCS-80™, MCS-85™, MCS-86™, and other 8-bit systems.

The UPI-41A™ has 1K words of program memory and 64 words of data memory on-chip. To allow full user flexibility the program memory is available as ROM in the 8041A version or as UV-erasable EPROM in the 8741A version. The 8741A and the 8041A are fully pin compatible for easy transition from prototype to production level designs. The 8641A is a one-time programmable (at the factory) 8741A which can be ordered as the first 25 pieces of a new 8041A order. The substitution of 8641A's for 8041A's allows for very fast turnaround for initial code verification and evaluation results.

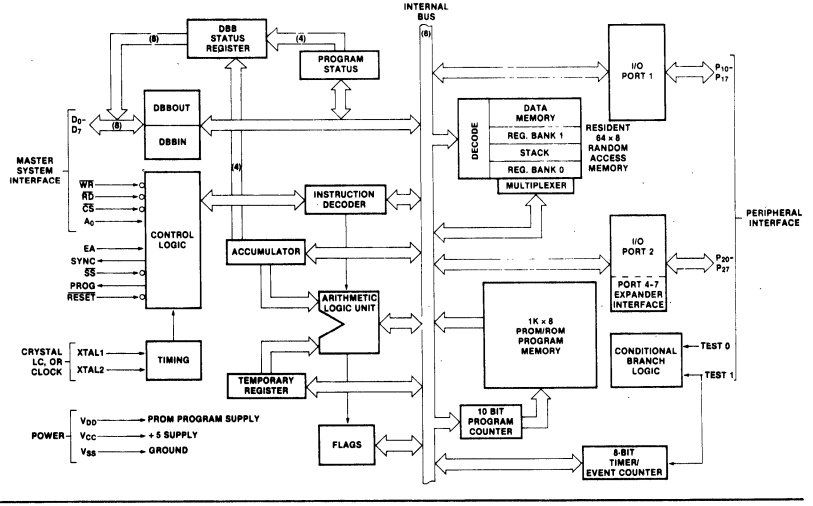
The device has two 8-bit, TTL compatible I/O ports and two test inputs. Individual port lines can function as either inputs or outputs under software control. I/O can be expanded with the 8243 device which is directly compatible and has 16 I/O lines. An 8-bit programmable timer/counter is included in the UPI device for generating timing sequences or counting external inputs. Additional UPI features include: single 5V supply, low power standby mode (in the 8041A), single-step mode for debug (in the 8741A), and dual working register banks.

Because it's a complete microcomputer, the UPI provides more flexibility for the designer than conventional LSI interface devices. It is designed to be an efficient controller as well as an arithmetic processor. Applications include keyboard scanning, printer control, display multiplexing and similar functions which involve interfacing peripheral devices to microprocessor systems.

PIN CONFIGURATION



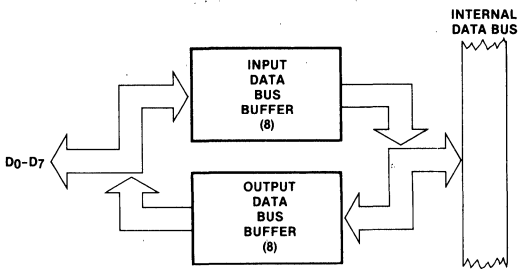
BLOCK DIAGRAM



00188A

UPI-41A™ FEATURES AND ENHANCEMENTS

1. Two Data Bus Buffers, one for input and one for output. This allows a much cleaner Master/Slave protocol.



2. 8 Bits of Status

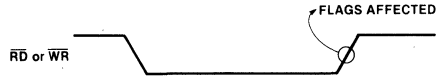
ST_7	ST_6	ST_5	ST_4	F_1	F_0	IBF	OBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

ST_4-ST_7 are user definable status bits. These bits are defined by the "MOV STS, A" single byte, single cycle instruction. Bits 4-7 of the accumulator are moved to bits 4-7 of the status register. Bits 0-3 of the status register are not affected.

MOV STS, A Op Code: 90H

1	0	0	1	0	0	0	0
D_7							D_0

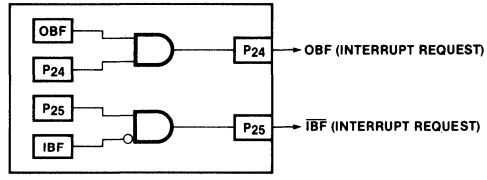
3. $\overline{RD}$ and $\overline{WR}$ are edge triggered. IBF, OBF, F_1 and INT change internally after the trailing edge of $\overline{RD}$ or $\overline{WR}$.



4. P_{24} and P_{25} are port pins or Buffer Flag pins which can be used to interrupt a master processor. These pins default to port pins on Reset.

   If the "EN FLAGS" instruction has been executed, P_{24} becomes the OBF (Output Buffer Full) pin. A "1" written to P_{24} enables the OBF pin (the pin outputs the OBF Status Bit). A "0" written to P_{24} disables the OBF pin (the pin remains low). This pin can be used to indicate that valid data is available from the UPI-41A (in Output Data Bus Buffer).

   If "EN FLAGS" has been executed, P_{25} becomes the $\overline{IBF}$ (Input Buffer Full) pin. A "1" written to P_{25} enables the $\overline{IBF}$ pin (the pin outputs the inverse of the IBF Status Bit). A "0" written to P_{25} disables the $\overline{IBF}$ pin (the pin remains low). This pin can be used to indicate that the UPI-41A is ready for data.



DATA BUS BUFFER INTERRUPT CAPABILITY

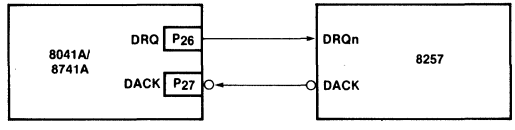
EN FLAGS Op Code: 0F5H

1	1	1	1	0	1	0	1
D_7							D_0

5. P_{26} and P_{27} are port pins or DMA handshake pins for use with a DMA controller. These pins default to port pins on Reset.

   If the "EN DMA" instruction has been executed, P_{26} becomes the DRQ (DMA ReQuest) pin. A "1" written to P_{26} causes a DMA request (DRQ is activated). DRQ is deactivated by DACK·RD, DACK·WR, or execution of the "EN DMA" instruction.

   If "EN DMA" has been executed, P_{27} becomes the $\overline{DACK}$ (DMA ACKnowledge) pin. This pin acts as a chip select input for the Data Bus Buffer registers during DMA transfers.



DMA HANDSHAKE CAPABILITY

EN DMA Op Code: 0E5H

1	1	1	0	0	1	0	1
D_1							D_0

00188A

PIN DESCRIPTION

Signal	Description
$D_0 - D_7$ (BUS)	Three-state, bidirectional DATA BUS BUFFER lines used to interface the UPI-41A to an 8-bit master system data bus.
P_{10}-P_{17}	8-bit, PORT 1 quasi-bidirectional I/O lines.
P_{20}-P_{27}	8-bit, PORT 2 quasi-bidirectional I/O lines. The lower 4 bits (P_{20}-P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4-7 access. The upper 4 bits (P_{24}-P_{27}) can be programmed to provide Interrupt Request and DMA Handshake capability. Software control can configure P_{24} as OBF (Output Buffer Full), P_{25} as $\overline{IBF}$ (Input Buffer Full), P_{26} as DRQ (DMA Request), and P_{27} as $\overline{DACK}$ (DMA ACKnowledge).
$\overline{WR}$	I/O write input which enables the master CPU to write data and command words to the UPI-41A INPUT DATA BUS BUFFER.
$\overline{RD}$	I/O read input which enables the master CPU to read data and status words from the OUTPUT DATA BUS BUFFER or status register.
$\overline{CS}$	Chip select input used to select one UPI-41A out of several connected to a common data bus.
A_0	Address input used by the master processor to indicate whether byte transfer is data or command.
TEST 0, TEST 1	Input pins which can be directly tested using conditional branch instructions.
	T_1 also functions as the event timer input (under software control). T_0 is used during PROM programming and verification in the 8741A.
XTAL1, XTAL2	Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	Output signal which occurs once per UPI-41A instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	External access input which allows emulation, testing and PROM/ROM verification.
PROG	Multifunction pin used as the program pulse input during PROM programming.
	During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	Input used to reset status flip-flops and to set the program counter to zero.
	$\overline{RESET}$ is also used during PROM programming and verification.
$\overline{SS}$	Single step input used in the 8741A in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	+5V main power supply pin.
V_{DD}	+5V during normal operation. +25V during programming operation. Low power standby pin in ROM version.
V_{SS}	Circuit ground potential.

UPI™ INSTRUCTION SET

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A,Rr	Add register to A	1	1
ADD A,@Rr	Add data memory to A	1	1
ADD A,#data	Add immediate to A	2	2
ADDC A,Rr	Add register to A with carry	1	1
ADDC A,@Rr	Add data memory to A with carry	1	1
ADDC A,#data	Add immed. to A with carry	2	2
ANL A,Rr	AND register to A	1	1
ANL A,@Rr	AND data memory to A	1	1
ANL A,#data	AND immediate to A	2	2
ORL A,Rr	OR register to A	1	1
ORL A,@Rr	OR data memory to A	1	1
ORL A,#data	OR immediate to A	2	2
XRL A,Rr	Exclusive OR register to A	1	1
XRL A,@Rr	Exclusive OR data memory to A	1	1
XRL A,#data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A,Pp	Input port to A	1	2
OUTL Pp,A	Output A to port	1	2
ANL Pp,#data	AND immediate to port	2	2
ORL Pp,#data	OR immediate to port	2	2
IN A,DBB	Input DBB to A, clear IBF	1	1
OUT DBB,A	Output A to DBB, set OBF	1	1
MOV STS,A	A_4-A_7 to Bits 4-7 of Status	1	1
MOVD A,Pp	Input Expander port to A	1	2
MOVD Pp,A	Output A to Expander port	1	2
ANLD Pp,A	AND A to Expander port	1	2
ORLD Pp,A	OR A to Expander port	1	2
DATA MOVES			
MOV A,Rr	Move register to A	1	1
MOV A,@Rr	Move data memory to A	1	1
MOV A,#data	Move immediate to A	2	2
MOV Rr,A	Move A to register	1	1
MOV @Rr,A	Move A to data memory	1	1
MOV Rr,#data	Move immediate to register	2	2
MOV @Rr,#data	Move immediate to data memory	2	2
MOV A,PSW	Move PSW to A	1	1
MOV PSW,A	Move A to PSW	1	1
XCH A,Rr	Exchange A and register	1	1
XCH A,@Rr	Exchange A and data memory	1	1
XCHD A,@Rr	Exchange digit of A and register	1	1
MOVP A,@A	Move to A from current page	1	2
MOVP3, A,@A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A,T	Read Timer/Counter	1	1
MOV T,A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1

00188A

Mnemonic	Description	Bytes	Cycles
CONTROL			
EN DMA	Enable DMA Handshake Lines	1	1
EN I	Enable IBF Interrupt	1	1
DIS I	Disable IBF Interrupt	1	1
EN FLAGS	Enable Master Interrupts	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ Rr, addr	Decrement register and jump	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 Flag = 1	2	2
JF1 addr	Jump on F1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumulator Bit	2	2

APPLICATIONS

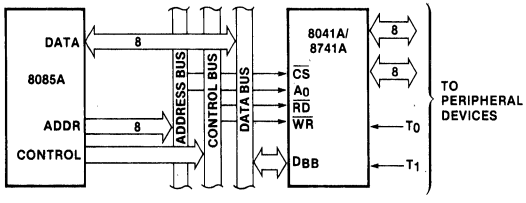


Figure 1. 8085A-8041A Interface

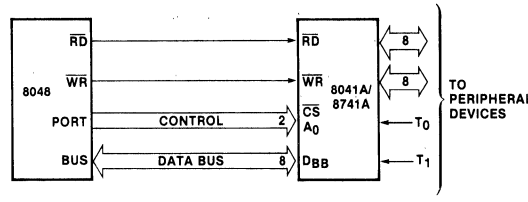


Figure 2. 8048-8041A Interface

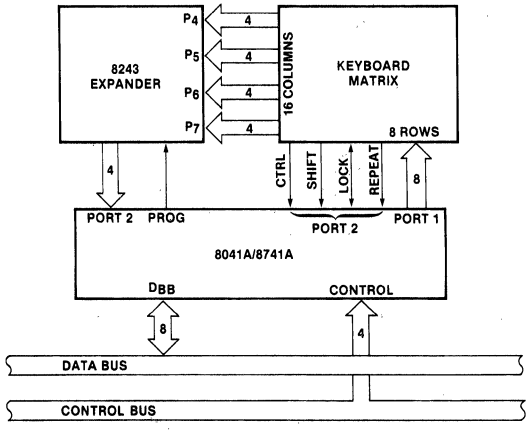


Figure 3. 8041A-8243 Keyboad Scanner

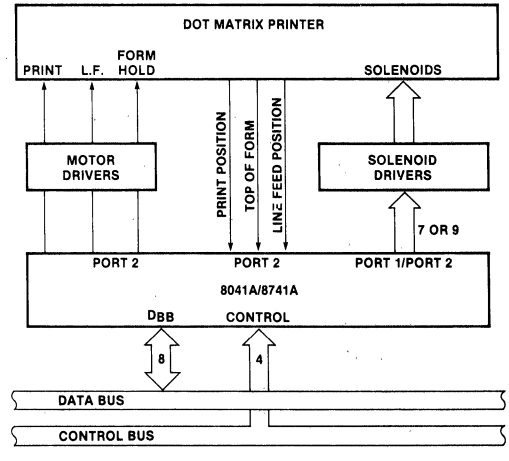


Figure 4. 8041A Matrix Printer Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin With Respect to Ground ..... 0.5V to +7V
Power Dissipation ..... 1.5 Watt

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

D.C. AND OPERATING CHARACTERISTICS

$T_A = 0°C$ to 70°C, $V_{SS} = 0V$, 8041A: $V_{CC} = V_{DD} = +5V \pm 10\%$, 8741A: $V_{CC} = V_{DD} = +5V \pm 5\%$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)	−0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)	2.2	V_{CC}		
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)	3.8	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0-D_7)		0.45	V	$I_{OL} = 2.0$ mA
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	$I_{OL} = 1.6$ mA
V_{OL2}	Output Low Voltage (Prog)		0.45	V	$I_{OL} = 1.0$ mA
V_{OH}	Output High Voltage (D_0-D_7)	2.4		V	$I_{OH} = -400\ \mu A$
V_{OH1}	Output High Voltage (All Other Outputs)	2.4		V	$I_{OH} = -50\ \mu A$
I_{IL}	Input Leakage Current (T_0, T_1, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A_0, EA)		±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OZ}	Output Leakage Current (D_0-D_7, High Z State)		±10	μA	$V_{SS} + 0.45 \le V_{IN} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.5	mA	$V_{IL} = 0.8V$
I_{LI1}	Low Input Load Current ($\overline{RESET}$, $\overline{SS}$)		0.2	mA	$V_{IL} = 0.8V$
I_{DD}	V_{DD} Supply Current		15	mA	Typical = 5 mA
$I_{CC} + I_{DD}$	Total Supply Current		125	mA	Typical = 60 mA

A.C. CHARACTERISTICS

$T_A = 0°C$ to 70°C, $V_{SS} = 0V$, 8041A: $V_{CC} = V_{DD} = +5V \pm 10\%$, 8741A: $V_{CC} = V_{DD} = +5V \pm 5\%$

DBB READ

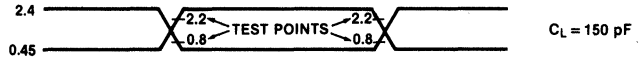
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	$\overline{CS}$, A_0 Setup to $\overline{RD}$↓	0		ns	
t_{RA}	$\overline{CS}$, A_0 Hold After $\overline{RD}$↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	250		ns	
t_{AD}	$\overline{CS}$, A_0 to Data Out Delay		225	ns	$C_L = 150$ pF
t_{RD}	$\overline{RD}$↓ to Data Out Delay		225	ns	$C_L = 150$ pF
t_{DF}	$\overline{RD}$↑ to Data Float Delay		100	ns	
t_{CY}	Cycle Time (Except 8741A-8)	2.5	15	μs	6.0 MHz XTAL
t_{CY}	Cycle Time (8741A-8)	4.17	15	μs	3.6 MHz XTAL

DBB WRITE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	$\overline{CS}$, A_0 Setup to $\overline{WR}$↓	0		ns	
t_{WA}	$\overline{CS}$, A_0 Hold After $\overline{WR}$↑	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	250		ns	
t_{DW}	Data Setup to $\overline{WR}$↑	150		ns	
t_{WD}	Data Hold After $\overline{WR}$↑	0		ns	

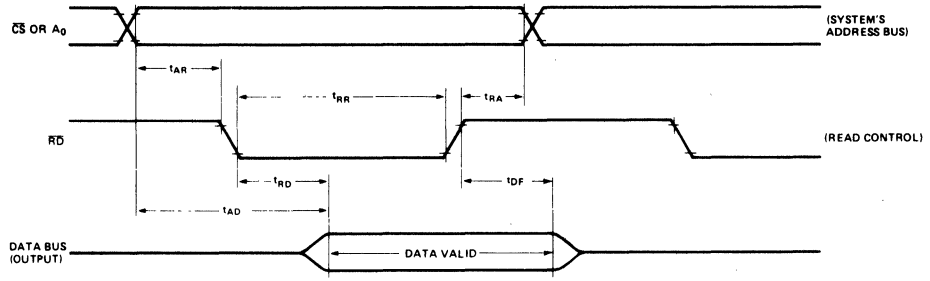
00188A

INPUT AND OUTPUT WAVEFORMS FOR A.C. TESTS

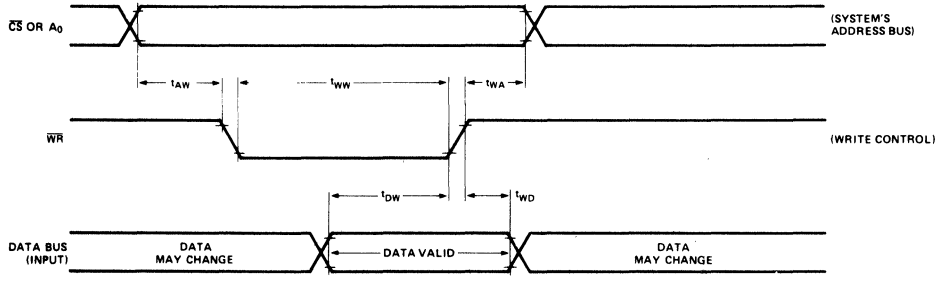


WAVEFORMS

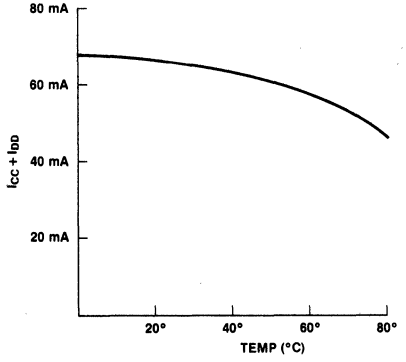
1. READ OPERATION—DATA BUS BUFFER REGISTER.



2. WRITE OPERATION—DATA BUS BUFFER REGISTER.



TYPICAL 8041/8741A CURRENT



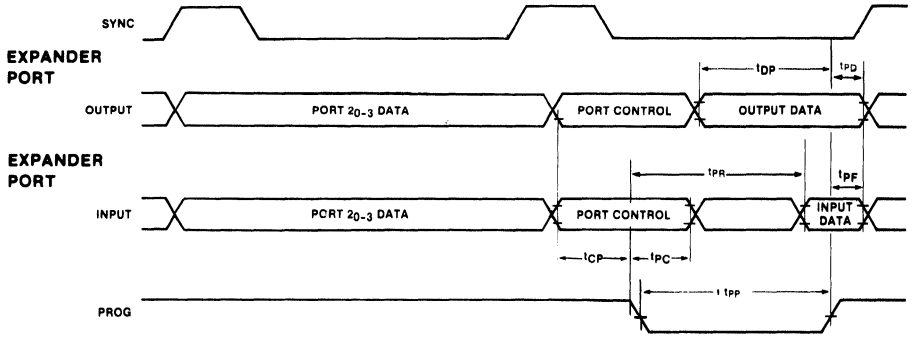
00188A

A.C. CHARACTERISTICS—PORT 2

$T_A = 0°C$ to $70°C$, 8041A: $V_{CC} = +5V \pm 10\%$, 8741A: $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of PROG	110		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	100		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
t_{PF}	Input Data Hold Time	0	150	ns	
t_{DP}	Output Data Setup Time	250		ns	
t_{PD}	Output Data Hold Time	65		ns	
t_{PP}	PROG Pulse Width	1200		ns	

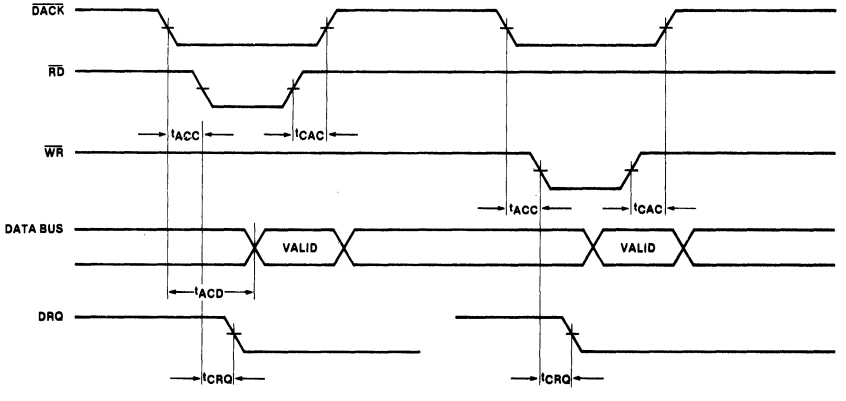
PORT 2 TIMING



A.C. CHARACTERISTICS—DMA

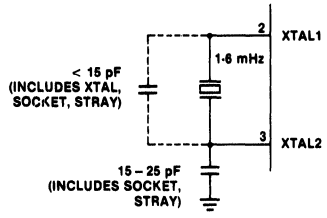
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{ACC}	$\overline{DACK}$ to $\overline{WR}$ or $\overline{RD}$	0		ns	
t_{CAC}	$\overline{RD}$ or $\overline{WR}$ to $\overline{DACK}$	0		ns	
t_{ACD}	$\overline{DACK}$ to Data Valid		225	ns	$C_L = 150$ pF
t_{CRQ}	$\overline{RD}$ or $\overline{WR}$ to DRQ Cleared		200	ns	

WAVEFORMS—DMA



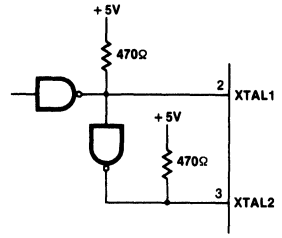
00188A

CRYSTAL OSCILLATOR MODE



CRYSTAL SERIES RESISTANCE SHOULD BE <75Ω AT 6 MHz; <180Ω AT 3.6 MHz.

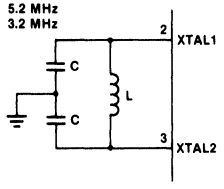
DRIVING FROM EXTERNAL SOURCE



BOTH XTAL1 AND XTAL2 SHOULD BE DRIVEN.
RESISTORS TO V_{CC} ARE NEEDED TO ENSURE V_{IH} = 3.8V
IF TTL CIRCUITRY IS USED.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz



$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$

$$C' = \frac{C + 3C_{PP}}{2}$$

$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE.

PROGRAMMING, VERIFYING, AND ERASING THE 8741A EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a missocketed 8741A will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. A_0 = 0V, $\overline{CS}$ = 5V, EA = 5V, $\overline{RESET}$ = 0V, TEST0 = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS and PROG floating.
2. Insert 8741A in programming socket
3. TEST 0 = 0v (select program mode)
4. EA = 23V (activate program mode)
5. Address applied to BUS and P20-1
6. $\overline{RESET}$ = 5v (latch address)
7. Data applied to BUS
8. V_{DD} = 25v (programming power)
9. PROG = 0v followed by one 50ms pulse to 23V
10. V_{DD} = 5v
11. TEST 0 = 5v (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0v
14. $\overline{RESET}$ = 0v and repeat from step 5
15. Programmer should be at conditions of step 1 when 8741A is removed from socket.

00188A

8741A Erasure Characteristics

The erasure characteristics of the 8741A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8741A in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8741A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8741A window to prevent unintentional erasure.

The recommended erasure procedure for the 8741A is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of 15 w-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8741A should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

A.C. TIMING SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{\text{RESET}}$ ↑	4tcy			
t_{WA}	Address Hold Time After $\overline{\text{RESET}}$ ↑	4tcy			
t_{DW}	Data in Setup Time to PROG ↑	4tcy			
t_{WD}	Data in Hold Time After PROG ↓	4tcy			
t_{PH}	$\overline{\text{RESET}}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD} Setup Time to PROG ↑	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	mS	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{\text{RESET}}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{\text{RESET}}$ Setup Time Before EA ↑.	4tcy			

Note: If TEST 0 is high, t_{DO} can be triggered by $\overline{\text{RESET}}$ ↑.

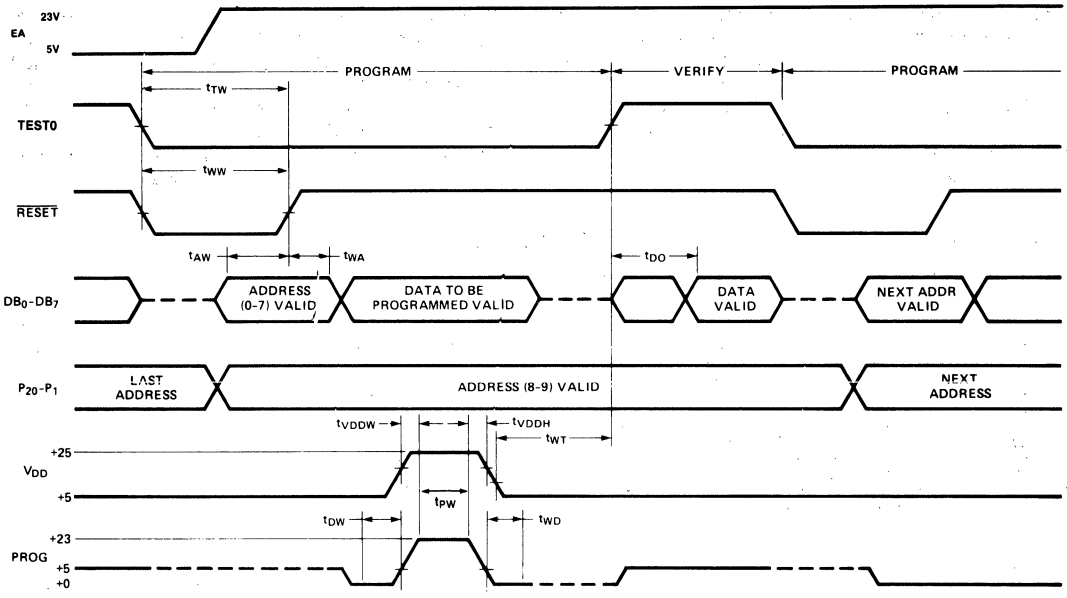
D.C. SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

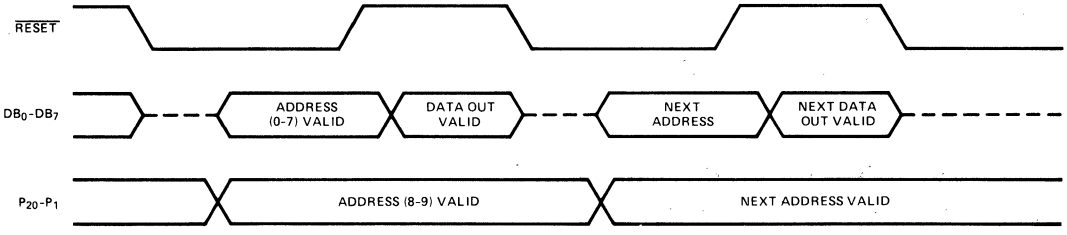
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. **PROG MUST FLOAT IF EA IS LOW (i.e., ≠23V), OR IF T0 = 5V FOR THE 8741A. FOR THE 8041A PROG MUST ALWAYS FLOAT.**
2. **XTAL1 AND XTAL 2 DRIVEN BY 3.6 MHz CLOCK WILL GIVE 4.17 μsec t_{CY}. THIS IS ACCEPTABLE FOR 8741A-8 PARTS AS WELL AS STANDARD PARTS.**
3. **A0 MUST BE HELD LOW (i.e., = 0V) DURING PROGRAM/VERIFY MODES.**

The 8741A EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP series) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

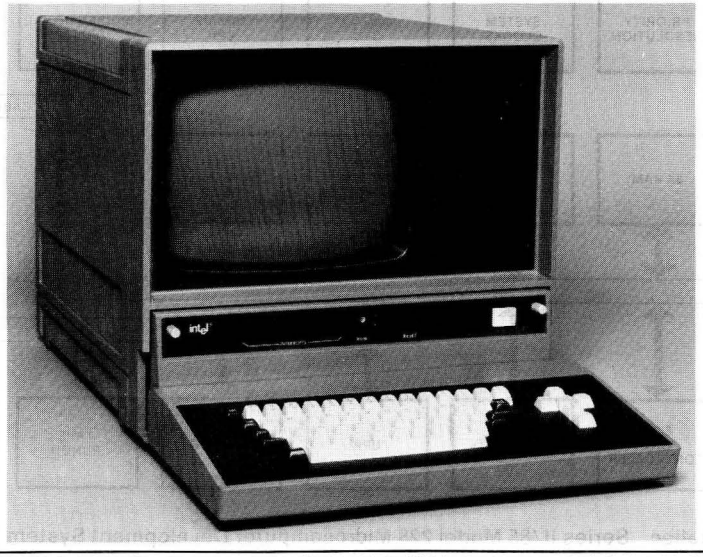
Support Products

10

MODEL 225
INTELLEC® SERIES II/85
MICROCOMPUTER DEVELOPMENT SYSTEM

- **Complete microcomputer development system for MCS®-86, MCS®-85, MCS®-80, and MCS®-48 microprocessor families**
- **High performance 8085A-2 CPU, 64K bytes RAM memory, and 4K bytes ROM memory**
- **Self-test diagnostic capability**
- **Built-in interfaces for high speed paper tape reader/punch, printer, and universal PROM programmer**
- **Integral 250K byte floppy disk drive with total storage capacity expandable to over 2M bytes of floppy disk storage and 7.3M bytes of hard disk storage**
- **Powerful ISIS-II Disk Operating System with relocating macroassembler, linker, locater, and CRT based editor CREDIT**
- **Supports PL/M, FORTRAN, BASIC, PASCAL and COBOL high level languages**
- **Software compatible with previous Intellec® systems**

The Intellec Series II/85 Model 225 Microcomputer Development System is a performance enhanced, complete microcomputer development system integrated into one compact package. The Model 225 includes a CPU with 64K bytes of RAM, 4K bytes of ROM, a 2000-character CRT, detachable full ASCII keyboard with cursor controls and upper/lower case capability, and a 250K-byte floppy disk drive. Powerful ISIS-II Disk Operating System software allows the Model 225 to be used quickly and efficiently for assembling and debugging programs for Intel's MCS-86, MCS-85, MCS-80, or MCS-48 microprocessor families. ISIS-II performs all file handling operations for the user, leaving him free to concentrate on the details of his own application. When used with an optional in-circuit emulator (ICE™) module, the Model 225 provides all of the hardware and software development tools necessary for the rapid development of a microcomputer-based product. Optional storage peripherals provide over 2 million bytes of floppy disk, and 7.3 million of hard disk storage capacity.



121599-001 Rev. A

FUNCTIONAL DESCRIPTION

Hardware Components

The Intellec Series II/85 Model 225 is a highly-integrated microcomputer development system consisting of a CRT chassis with a 6-slot cardcage, power supply, fans, cables, single floppy disk drive, and two printed circuit cards. A separate, full ASCII keyboard is connected with a cable. A block diagram of the Model 225 is shown in Figure 1.

CPU Cards — The master CPU card contains its own microprocessor, memory, I/O, interrupt and bus interface circuitry implemented with Intel's high technology LSI components. Known as the integrated processor card (IPC), it occupies the first slot in the cardcage. A second slave CPU card is responsible for all remaining I/O control including the CRT and keyboard interface. This card, mounted on the rear panel, also contains its own microprocessor, RAM and ROM memory, and I/O interface logic, thus, in effect, creating a dual processor environment. Known as the I/O controller (IOC), the slave CPU card communicates with the IPC over an 8-bit bidirectional data bus.

Expansion — Five remaining slots in the cardcage are available for system expansion. Additional expansion of 4 slots can be achieved through the addition of an Intellec Series II expansion chassis.

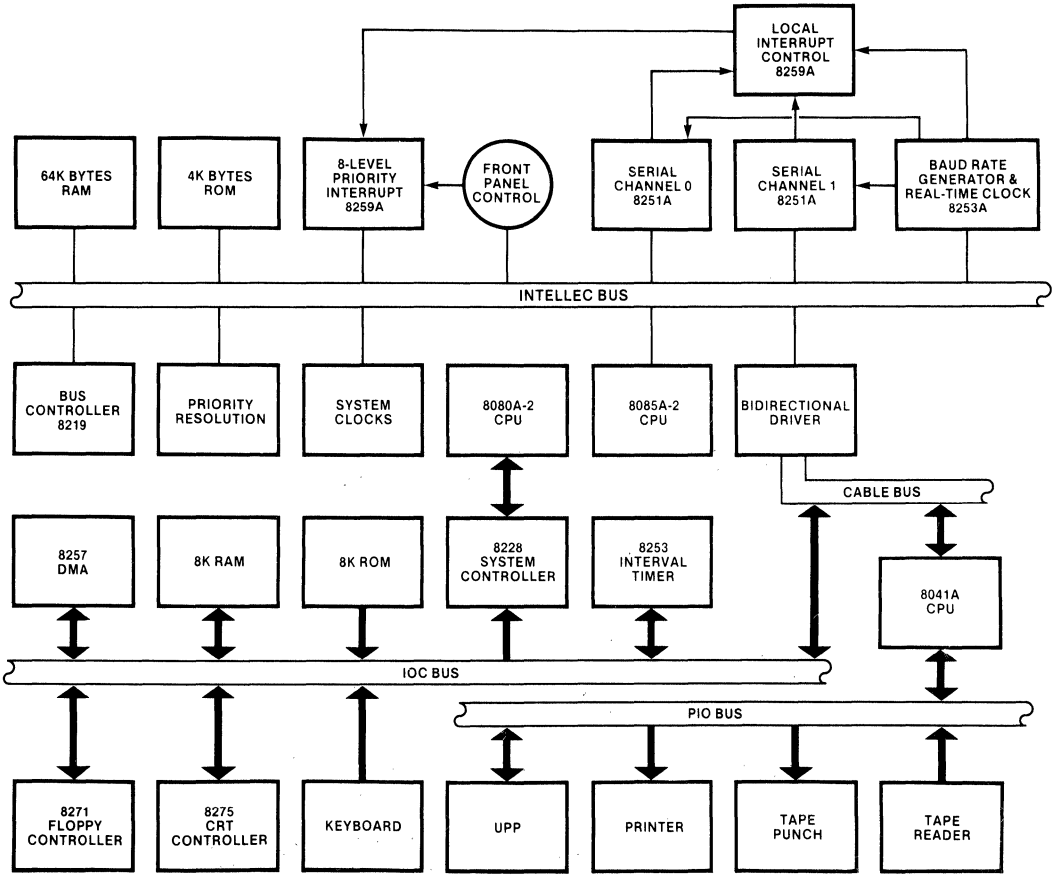


Figure 1. Intellec Series II/85 Model 225 Microcomputer Development System Block Diagram

System Components

The heart of the IPC is an Intel NMOS 8-bit microprocessor, the 8085A-2, running at 4.0 MHz. 64K bytes of RAM memory are provided on the board using 16K RAMs. 4K of ROM is provided, preprogrammed with system bootstrap "self-test" diagnostics and the Intellec Series II/85 System Monitor. The eight-level vectored priority interrupt system allows interrupts to be individually masked. Using Intel's versatile 8259A interrupt controller, the interrupt system may be user programmed to respond to individual needs.

Input/Output

IPC Serial Channels — The I/O subsystem in the Model 225 consists of two parts: the IOC card and two serial channels on the IPC itself. Each serial channel is RS232 compatible and is capable of running asynchronously from 110 to 9600 baud or synchronously from 150 to 56K baud. Both may be connected to a user defined data set or terminal. One channel contains current loop adapters. Both channels are implemented using Intel's 8251A USART. They can be programmed to perform a variety of I/O functions. Baud rate selection is accomplished through an Intel 8253 interval timer. The 8253 also serves as a real-time clock for the entire system. I/O activity through both serial channels is signaled to the system through a second 8259A interrupt controller, operating in a polled mode nested to the primary 8259A.

IOC Interface — The remainder of system I/O activity takes place in the IOC. The IOC provides interface for the CRT, keyboard, and standard Intellec peripherals including printer, high speed paper tape reader/punch, and universal PROM programmer. The IOC contains its own independent microprocessor, an 8080A-2. The CPU controls all I/O operations as well as supervising communications with the IPC. 8K bytes of ROM contain all I/O control firmware. 8K bytes of RAM are used for CRT screen refresh storage. These do not occupy space in Intellec Series II main memory since the IOC is a totally independent microcomputer subsystem.

Integral CRT

Display — The CRT is a 12-inch raster scan type monitor with a 50/60 Hz vertical scan rate and 15.5kHz horizontal scan rate. Controls are provided for brightness and contrast adjustments. The interface to the CRT is provided through an Intel 8275 single-chip programmable CRT controller. The master processor on the IPC transfers a character for display to the IOC, where it is stored in RAM. The CRT controller reads a line at a time into its line buffer through an Intel 8257 DMA controller and then feeds one character at a time to the character generator to produce the video signal. Timing for the CRT control is provided by an Intel 8253 interval timer. The screen display is formatted as 25 rows of 80 characters. The full set of ASCII characters is displayed, including lower case alphas.

Keyboard — The keyboard interfaces directly to the IOC processor via an 8-bit data bus. The keyboard contains an Intel UPI-41™ Universal Peripheral Interface, which scans the keyboard, encodes the characters, and buffers the characters to provide N-key rollover. The keyboard itself is a high quality typewriter style keyboard containing the full ASCII character set. An upper/lower case switch allows the system to be used for document preparation. Cursor control keys are also provided.

Peripheral Interface

A UPI-41 Universal Peripheral Interface on the IOC board provides interface for other standard Intellec peripherals including a printer, high speed paper tape reader, high speed paper tape punch, and universal PROM programmer. Communication between the IPC and IOC is maintained over a separate 8-bit bidirectional data bus. Connectors for the four devices named above, as weli as the two serial channels, are mounted directly on the IOC itself.

Control

User control is maintained through a front panel, consisting of a power switch and indicator, reset/boot switch, run/halt light, and eight interrupt switches and indicators. The front panel circuit board is attached directly to the IPC, allowing the eight interrupt switches to connect to the primary 8259A, as well as to the Intellec Series II bus.

Integral Floppy Disk Drive

The integral floppy disk is controlled by an Intel 8271 single chip, programmable floppy disk controller. It transfers data via an Intel 8257 DMA controller between an IOC RAM buffer and the diskette. The 8271 handles reading and writing of data, formatting diskettes, and reading status, all upon appropriate commands from the IOC microprocessor.

MULTIBUS™ Interface Capability

All Intellec Series II/85 models implement the industry standard MULTIBUS protocol. The MULTIBUS protocol enables several bus masters, such as CPU and DMA devices, to share the bus and memory by operating at different priority levels. Resolution of bus exchanges is synchronized by a bus clock signal derived independently from processor clocks. Read/write transfers may take place at rates up to 5 MHz. The bus structure is suitable for use with any Intel microcomputer family.

SPECIFICATIONS

Host Processor (IPC)

8085A-2 based, operating at 4.0 MHz.
RAM — 64K on the CPU card
ROM — 4K (2K in monitor, 2K in boot/diagnostic)
Bus — MULTIBUS™ bus, maximum transfer rate of 5 MHz
Clocks — Host processor, crystal controlled at 4.0 MHz, bus clock, crystal controlled at 9.8304 MHz

I/O Interfaces

Two Serial I/O Channels, RS232C, at 110-9600 baud (asynchronous) or 150-56K baud (synchronous). Baud rates and serial format fully programmable using Intel 8251A USARTs. Serial Channel 1 additionally provided with 20 mA current loop. Parallel I/O interfaces provided for paper tape punch, paper tape reader, printer, and UPP-103 Universal PROM Programmer.

Interrupts

8-level, maskable, nested priority interrupt network initiated from front panel or user selected devices.

Direct Memory Access (DMA)

Standard capability on MULTIBUS interface; implemented for user selected DMA devices through optional DMA module—maximum transfer rate of 5 MHz.

Memory Access Time

RAM — 470 ns max
PROM — 540 ns max

Integral Floppy Disk Drive

Floppy Disk System Capacity —
250K bytes (formatted)
Floppy Disk System Transfer Rate —
160K bits/sec
Floppy Disk System Access Time —
Track to Track: 10 ms max
Average Random Positioning: 260 ms
Rotational Speed: 360 rpm
Average Rotational Latency: 83 ms
Recording Mode: FM

Physical Characteristics

CHASSIS

Width — 17.37 in. (44.12 cm)
Height — 15.81 in. (40.16 cm)
Depth — 19.13 in. (48.59 cm)
Weight — 73 lb. (33 kg)

KEYBOARD

Width — 17.37 in. (44.12 cm)
Height — 3.0 in. (7.62 cm)
Depth — 9.0 in. (22.86 cm)
Weight — 6 lb. (3 kg)

Electrical Characteristics

DC POWER SUPPLY

Volts Supplied	Amps Supplied	Typical System Requirements
+ 5 ± 5%	30.0	17.0
+12 ± 5%	2.5	1.1
−12 ± 5%	0.3	0.1
−10 ± 5%	1.0	0.08
+15 ± 5%*	1.5	1.5
+24 ± 5%*	1.7	1.7

*Not available on bus.

AC REQUIREMENTS FOR MAINFRAME

110V, 60 Hz — 5.9 Amp
220V, 50 Hz — 3.0 Amp

Environmental Characteristics

Operating Temperature — 16°C to 32°C (61°F to 90°F)
Humidity — 20% to 80%

Equipment Supplied

Model 225 Chassis including:
- Integrated Processor Card (IPC)
- I/O Controller Board (IOC)
- CRT
- ROM-Resident System Monitor

Detachable keyboard
ISIS-II System Diskette with MCS-80/MCS-85 Macroassembler
ISIS-II CREDIT Diskette CRT-Based Text Editor

Documentation Supplied

A Guide to Microcomputer Development Systems, 9800558

Intellec® Series II Model 22X/23X Installation Manual, 9800559

ISIS-II System User's Guide, 9800306

Intellec® Series II Hardware Reference Manual, 9800556

8080/8085 Assembly Language Programming Manual, 9800301

ISIS-II 8080/8085 Assembler Operator's Manual, 9800292

Intellec® Series II Systems Monitor Source Listing, 9800605

Intellec® Series II Schematic Drawings, 9800554

ISIS-II CREDIT (CRT-Based Text Editor) User's Guide, 9800902

Additional manuals may be ordered from any Intel sales representative or distributor office, or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Part Number	Description
MDS-225*	Intellec® Series II/85 Model 225 Microcomputer Development System (110V/60 Hz)
MDS-226*	Intellec® Series II/85 Model 226 Microcomputer Development System (220V/50 Hz)

***"MDS"** is an ordering code only, and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corp.

INTELLEC PROMPT 48
MCS-48 MICROCOMPUTER DESIGN AID

Complete low cost design aid and EPROM programmer for revolutionary MCS-48 single component computers

Simplifies microcomputing, allowing user to enter, run, debug, and save machine language programs with calculator-like ease

Utilizes two removable 8-bit MCS-48 CPUs
- **— 8748 CPU with erasable, reprogrammable on-chip program memory**
- **— 8035 CPU with off-chip program memory**

1K-byte erasable, reprogrammable on-chip (8748), expandable program memory, 1K-byte RAM in PROMPT system

64 bytes RAM on-chip, expandable register memory

256 bytes expandable RAM data memory in PROMPT system

27 on-chip TTL compatible expandable I/O lines

On-chip clock, internal timer/event counter, two vectored interrupts, eight level stack control

Single +5V DC system power requirement

Integral keyboard and displays (no teletypewriter or CRT terminal required)

Extensive PROMPT 48 monitor, allowing system I/O, bus, and memory expansion

Includes comprehensive design library

The Intellec Prompt 48 MCS-48 Microcomputer Design Aid is a low cost, fully-assembled design aid for the revolutionary 8748 single component microcomputer. PROMPT 48 simplifies the programming of MCS-48 systems — programs may be entered and debugged with calculator-like ease on the large, informative display and keyboard panel. The comprehensive design library with tutorial manual is ideal for newcomers to microcomputing. PROMPT 48's panel connector allows easy access to I/O ports and system bus. Thus users can expand program memory beyond the 1K bytes provided internally.

FEATURES

Single Component Computer

The 8748 is the first microcomputer fully integrated on one component. All elements of a computing system are provided, including CPU, RAM, I/O, timer, interrupts, and erasable, reprogrammable nonvolatile program memory.

Programming Socket

PROMPT's programming socket programs this revolutionary "smart PROM"—the 8748—in a highly reliable, convenient manner. A fail-safe interlock ensures the device is properly inserted before applying programming pulses. Each location may be individually programmed, one byte at a time. A read-before-write programming algorithm prevents device damage by inadvertently programming unerased memory.

MCS-48 Processors

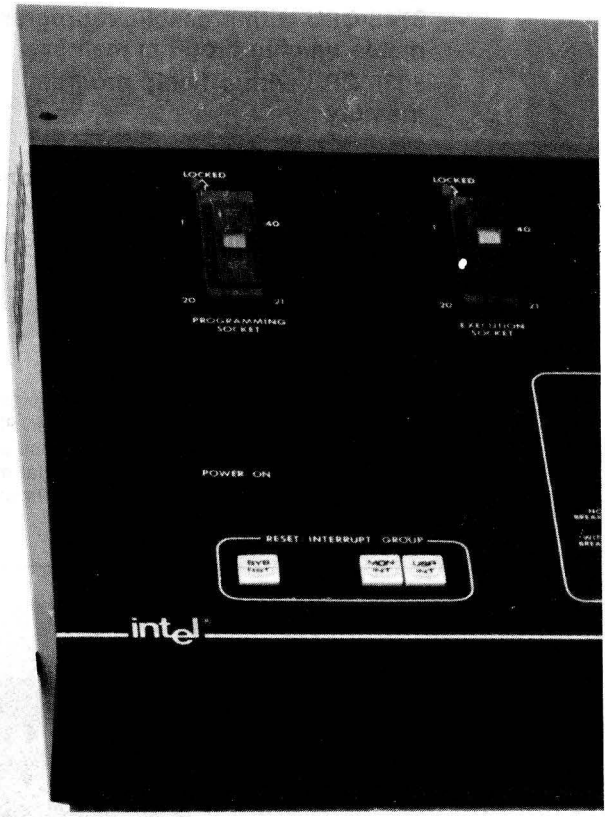
The execution socket accepts either an 8035 or an 8748 MCS-48 processor. Both are supplied with each PROMPT 48, and either can serve as heart of the PROMPT system. There are no processors within the PROMPT 48 mainframe, which instead contains monitor ROM and RAM, user RAM, peripherals, drivers, and sophisticated control circuitry. Once a processor is seated in the execution socket and power is applied, the PROMPT system comes to life. Various access modes may be selected such as program execution from PROMPT system RAM, or from on-chip PROM. Thus programs may first be executed from PROMPT RAM with the 8035 processor. When debugging is complete, the 8035 (execution socket) processor can program the 8748 (programming socket) processor. Finally, a programmed 8748 processor may be exercised by itself from the execution socket. The execution socket processor runs either monitor or user programs.

System Monitor

The system reset command initializes the PROMPT system and enters the monitor. The monitor interrupt command exits a user program gracefully, preserving system status and entering the monitor. The user interrupt command causes an interrupt only if the PROMPT system is running a user program. A comprehensive system monitor resides in four 1K-byte read only memories. It drives the PROMPT keyboard and displays and responds to commands and functions. The top 16 bytes of on-chip program memory must be used by the PROMPT system to switch between monitor and user programs. It requires one level of the MCS-48 eight-level stack.

Commands

PROMPT 48's commands are grouped and color-coded to simplify access to the 8748's separate program and data memory. Registers, data memory, or program memory, may be examined and modified with the examine and modify commands. Then either the next or previous register and memory locations may be accessed with one keystroke. Programs may be exercised in three modes. The go no break (GO NO BREAK) runs in real time. The go with break (GO WITH BREAK) mode is not real time — after each instruction the MCS-48 program counter is compared against pending breakpoints. If no break is encountered, execution resumes. The go single step (GO SINGLE STEP) mode exercises one instruction at a time. Commands are like sentences, with parameters separated by ⊡ NEXT. Each command ends with ⊡ EXECUTE/END. In addition to the PROMPT basic commands, thirteen functions simplify programming. Each is started merely by pressing a hex data/function key and entering parameters as required, as shown in Table 1.



Cable Interface

An optional cable, PROMPT-SER, directly connects the PROMPT system to virtually any terminal via a rear access slot.

Key	Function	Operation
2	Port 2 map	Allows specification of direction of each pin on port 2. Port 2 is multiplexed to address external program memory and expand I/O. Thus it must be buffered; the P2 map command establishes the direction of buffering.
3	Program EPROM	Programs 8748 EPROMs.
4	Byte search (with optional mask)	Sweeps through register, data, or program memory searching for byte matches. Starting and ending memory addresses are specified.
5	Word search (with optional mask)	Sweeps through register, data, or program memory searching for word matches. Starting and ending memory addresses are specified.
6	Hex calculator	Computes hexadecimal sums and differences.
7	8748 program for debug	Similar to program EPROM, but ensures that the top of program memory contains monitor re-entry code for debugging.
8	Compare	Verifies any portions of EPROM program memory against PROMPT memory.
9	Move memory	Allows blocks of register, data, or program memory to be moved.
A	Access	Specifies one of six access modes for PROMPT 48. For example EPROM, PROMPT RAM, or external program memory, and a variety of input/output options may be selected.
B	Breakpoint	Allows any or all of the eight breakpoints to be set and cleared.
C	Clear	Clears portions of register, data, or program memory.
D	Dump	Dumps register, data, or program memory to PROMPT's serial channel: for example, a teletypewriter paper tape punch.
E	Enter	Enters (reads) register, data, or program memory from PROMPT's serial channel.
F	Fetch	Fetches programs from EPROM to PROMPT RAM.

Table 1. PROMPT 48 Commands and Functions

Access

Easy access to the pins of the executing processor is provided via the I/O ports and bus connector. Only the EA external access, SS single step, and X1, X2 clock inputs are reserved for the PROMPT system.

Expansion

Program or data memory may be expanded beyond that provided on-chip or in the PROMPT system. I/O ports may be expanded, as with the 8243, or peripheral controllers may be memory-mapped. The I/O ports and Bus connector allows the execution socket processor to be directly interfaced to prototype systems, yet be controlled from the PROMPT panel.

Control

The command/function group panel keyboard and displays completely control PROMPT 48—a teletypewriter or CRT terminal is not needed. A hyphen prompting character appears whenever a command or function can be entered. Addresses and data are shown whenever examining registers and memory. Parameters for commands and functions are also shown.

FUNCTIONAL DESCRIPTION

"PROMPT" stands for PROgraMming Tool. It is a programmer for 8748 EPROMs, and a versatile aid for debugging MCS-48 programs. Programs can be entered via its integral panel keyboard, programming socket, or serial channel. Almost any terminal can be interfaced to the serial channel, including a teletypewriter, CRT, or an Intellec microcomputer development system. Intellec PROMPT 48 simplifies the programming of MCS-48 systems. Like the 8748 it is radically new, highly integrated, and expandable. Like the MCS-48 family, it is low cost, and ideal for small applications and programs. It is a design aid, not a development system with sophisticated software and peripherals.

MCS-48 Processors

PROMPT 48 comes complete with two of Intel's revolutionary MCS-48 processors: an 8748-4 Single Component 8-Bit Microcomputer and and 8035-4 Single Component 8-Bit Microcomputer. Advances in n-channel MOS technology allow Intel, for the first time to integrate into one 40-pin component all computer functions:

8-bit CPU
1K x 8-bit EPROM/ROM program memory
64 x 8-bit RAM data memory
27 input/output lines
8-bit timer/event counter

Performance — More than 90 instructions — each one or two cycles — make the single chip MCS-48 equal in performance to most multi-chip microprocessors. The MCS-48 is an efficient controller and arithmetic processor, with extensive bit handling, binary, and BCD arithmetic instructions. These are encoded for minimum program length; 70% are single byte operation codes, and none is more than two bytes.

Flexibility — Three interchangeable, pin-compatible devices offer flexibility and low cost in development and production, as follows:

8748 — with user-programmable and erasable EPROM program memory for prototype and pre-productions systems.

8048 — with factory-programmed mask ROM memory for low-cost, high volume production.

8035 — without program memory, for use with external program memories.

Circuitry — Each MCS-48 processor operates on a single +5V supply, with internal oscillator and clock driver, and circuitry for interrupts and resets. Extra circuitry is in the 8048 ROM processor to allow low power standby operation. The 64 x 8 RAM data memory can be independently powered.

Compatibility — For systems requiring additional compatibility, the MCS-48 can be expanded with the new 8243 I/O expander, 8155 I/O and 256-byte RAM, 8755 I/O and 2K-byte EPROM, or 8355 I/O and 2K ROM devices. MCS-48 processors readily interface to MCS-80/85 peripherals and standard memories.

Memory Capacity

PROMPT 48 is a complete, fully assembled and powered microcomputer system including program memory, data memory, I/O, and system monitor beyond that available on MCS-48 single component computers. 1K bytes of PROMPT system RAM serve as "writable program memory" — a ROM simulator for the program memory on each MCS-48 computer. 256 bytes of PROMPT system RAM serve as "external data memory," beyond the 64 register bytes on each MCS-48 computer. Users may further expand program or data memory via the panel I/O ports and bus connector.

Programming

Programs written first in assembly language, are entered in machine language and debugged with calculator-like ease on the large, informative display and keyboard panel. Most MCS-48 operations can be specified with only two keystrokes. Once entered, routines can be exercised one instruction (single step) or many instructions at a time. The principal MCS-48 register — the accumulator — is displayed while single stepping. Programs can be executed in real time (GO NO BREAK) or with as many as eight different breakpoints (GO WITH BREAK).

Control

PROMPT 48 can be fully controlled either by the panel keyboard and displays, or remotely by a serial channel. Thus a teletypewriter or CRT can be used but neither is required.

Access

The PROMPT panel I/O ports and bus connector allow easy access to all MCS-48 pins except those reserved for control by the PROMPT system, namely EA external access, SS single step, and X1, X2 clock inputs.

Optional Expansion

PROMPT 48 may be expanded beyond the resources on both the MCS-48 single component computer and the PROMPT system. External program and data memory may be interfaced and input/out ports added with the 8243 I/O expander.

Documentation

The PROMPT 48 manual includes chapters for the reader with little or no programming experience. Topics treated range from number systems to microcomputer hardware design. A novel, unifying set of tutorial diagrams — MICROMAPS — simplify microcomputer concepts. PROMPT's handy, pocket-sized reference cardlet can be affixed to the mainframe. Programming pads aid in the organization and documentation of programs. These features, plus a comprehensive design library of manuals, articles, and application notes, make the Intellec PROMPT 48 ideal for the newcomer to microcomputing.

SPECIFICATIONS

Timing

Basic Instruction — 2.5 μs
Cycle Time — t_{CY} = 2.5 μs
Clock — 6 MHz ± 0.1%

Memory Bytes

The 8748 contains 64 bytes of register memory, no external data memory, and 1024 bytes of RAM program memory. The PROMPT system provides 256 bytes of external data memory, and 1024 bytes of RAM program memory. PROMPT RAM program memory can be used in place of the on-chip EPROM program memory; thus programs less than 1024 bytes may be designed. For larger programs additional memory can be directly interfaced to the MCS-48 bus via the PROMPT panel I/O ports and bus connector.

Memory Configuration

Memory	Maximum	On Chip	In PROMPT 48
Register	64	64	0
Data	3328	0	256
Program	4096	1024 EPROM	1024 RAM

I/O Ports

All MCS-48 I/O ports are accessible on the PROMPT panel connector.

Bus — A true bidirectional 8-bit port with associated strobes. If the bidirectional feature is not needed, bus can serve as either a statically latched output port or a non-latching input port. Input and output lines cannot be mixed.

Ports 1 and 2 — Data written to these 8-bit ports is latched and remains unchanged until written. As inputs these lines are not latching. The lines of ports 1 and 2 are called quasibidirectional. A special output structure allows each line of port 1 and half of port 2 to serve as an input, an output, or both. Any mix of input, output, and both lines is allowed.

T0, T1, and INT — Three pins that can serve as inputs. T0 can be designated as a clock output. Input/output can be expanded via the PROMPT panel connector with a special I/O expander (8243) or standard peripherals.

Reset and Interrupts

Reset — initializes the PROMPT system and enters the monitor.

Monitor Interrupt — exits a user program gracefully, preserving system status and entering the monitor.

User Interrupt — causes an interrupt only if the PROMPT system is running a user program.

The processor traps to location 3_{16}. The MCS-48 timer/event counter is not used by the PROMPT system and is available to the user. Either timer flag or interrupt will signal when overflow has occurred. The timer interrupt can be used only in the go-no-break (real time) mode.

EPROM Programming

PROMPT 48 provides a programming socket to directly program 8748s. Programs are loaded into the PROMPT RAM program memory via keyboard. EPROM, teletypewriter, or other serial interface. A fail-safe interlock ensures programming pulses are applied only if the device is properly inserted. Inadvertant reprogramming is prevented by a read-before-write programming algorithm. Each location may be individually programmed, one byte at a time.

Panel I/O Ports and Bus Connectors

All MCS-48 pins, except five, are accessible on the I/O ports and bus connector. The five reserved for PROMPT system control are EA external access, SS single step, X1, X2 crystal inputs, and 5V. Due to internal buffering of the MCS-48 bus, access times will be negligibly degraded by the PROMPT system. Since MCS-48 processors do not communicate internal address gate status, bus data must be driven out if neither PSEN nor RD is asserted.

System Devices

Both user programs and the PROMPT monitor enjoy access to system devices: serial I/O, panel displays, and keyboard. These are memory-mapped to program memory addresses beyond 2K.

Serial I/O — The serial I/O port (data 820_{16}, control 821_{16}) is defined by software and jumpers for 110 baud, 20 mA current loop, but can easily be jumpered for other baud rates and RS232C levels. Asynchronous or synchronous transmission, data format, control characters, and parity can be programmed.

Panel Displays — Eight display ports (data 810-817_{16}) allow each of the panel displays to be written from user programs. Data written on a display device will time out after a fixed interval. Displays must be refreshed on a polled or interrupt-driven basis. User programs can call software drivers which provide this capability.

Keyboard — Software is used to debounce the panel keyboard (data 810_{16}). The monitor's input routines (see Software Drivers) provide this debouncing and can be called from user programs.

Commands

Single step / With break / No break } Go

Examine/modify { Register / Data / Program } Memory

Open previous/clear/entry ⊡ Next ☐ Execute/End

Functions

2 Port 2 map
3 Program EPROM (8748)
4 Search (R, D or P)* memory for 1 byte, optional mask
5 Search (R, D or P) memory for 2 bytes, optional mask
6 Hexadecimal calculator +, −
7 8748 program EPROM for debug
8 Compare EPROM with memory
9 Move memory (R, D or P)
A Access
B Breakpoint
C Clear memory (R, D or P)
D Dump memory (R, D or P)
E Enter (read) memory (R, D or P)
F Fetch EPROM program memory

Note
*R, D, or P is register, data, or program.

Software Drivers

Panel Keyboard In — KBIN, KDBIN
Panel Display Out — DGS6, DGOUT, HXOUT, BLK, REFS, ENREF
Serial Channel — CI, CO, RI, PO, CSTS

Connectors

Serial I/O — 3M 3462-0001 Flat Crimp/AMP 88106-1 Flat Crimp/ TI H312113 Solder/AMP 1-583485-5 Solder.

Panel I/O Ports and Bus Connector — 3M 3425 Flat Crimp. A complete cable set including wirewrap header for prototyping is included with each PROMPT.

Equipment Supplied

PROMPT 48 mainframe with two MCS-48 processors (8748, 8035), display/keyboard, EPROM programmer, power supply, cabinet, and ROM-based monitor.
110 V AC power cable
110 or 220 V AC
Fuse
Panel I/O ports
Bus connector cable set

Physical Characteristics

Height — 5.3 in. (13.5 cm) max
Width — 17 in. (43.2 cm)
Depth — 17 in. (43.2 cm) max
Weight — 21 lb. (9.6 kg)

Electrical Characteristics

Power Requirements — either 115 or 230V AC (± 10%) may be switch selected on the mainframe. 1.8 amps max current (at 125 V AC).
Frequency — 47-63 Hz

Environmental Characteristics

Operating Temperature — 0°C to +40°C
Non-Operating Temperature — 20°C to +65°C

Reference Manuals

9800402 — Intellec PROMPT 48 User's Manual (SUPPLIED)
9800270 — MCS-48 User's Manual (SUPPLIED)
9800255 — MCS-48 and UPI-41 Assembly Language Programming Manual (SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Part Number	Description
PROMPT-48 or PROMPT-48-220V	Intellec PROMPT 48 MCS-48 microcomputer design aid. Complete with two MCS-48 processors (8748 and 8305), EPROM programmer, integral keyboard, displays, and system monitor in ROM.
PROMPT-SER	Serial cable for connecting PROMPT to TTY, CRT

intel®

ICE-49
MCS-48 IN-CIRCUIT EMULATOR

Emulates 8049, 8048, 8748, 8039, 8035, and 8021* Microcomputers

Extends Intellec microcomputer development system debug power to user configured system via external cable and 40-pin plug, replacing system MCS-48 device

Emulates user system MCS-48 device in real time

Shares static RAM memory with user system for program debug

Provides hardware comparators for user designated break conditions

Eliminates need for extraneous debugging tools residing in user system

Collects bus, register, and MCS-48 status information on instructions emulated

Provides capability to examine and alter MCS-48 registers, memory, and flag values, and to examine pin and port values

Integrates hardware and software efforts early to save development time

The ICE-49 MCS-48 In-Circuit Emulator module is an Intellec-resident module that interfaces with any MCS-48 system. The MCS-48 family consists of the 8049, 8048, 8748, 8039, 8035, and 8021 microcomputers. The ICE-49 module interfaces with an MCS-48 system through a cable terminating in an MCS-48 pin-compatible plug which replaces the MCS-48 device in the sytem. With the ICE-49 plug in place, the designer has the capability to execute the system in real time while collecting up to 255 instruction cycles of real-time trace data. In addition, he can single step the system program to monitor more closely the program logic during execution. Static RAM memory is available through the ICE-49 module to emulate MCS-48 program and data memory. The designer can display and alter the contents of data and replacement RAM control memory, internal MCS-48 registers and flags and I/O ports. Powerful debug capability is extended into the MCS-48 system while ICE-49 debug hardware and software remain inside the Intellec system. Symbolic reference capability allows the designer to use meaningful symbols rather than absolute values when examining and modifying memory, registers, flags, and I/O ports in this system.

*EM1 emulator board is also required.

FUNCTIONAL DESCRIPTION

Debug Capability Inside User System

The ICE-49 module provides the user with the ability to debug a full prototype or production system without introducing extraneous hardware or software test tools. The module connects to the user system through the socket provided for the MCS-48 device in the user system. Intellec memory is used for the execution of the ICE-49 software. The Intellec console and file handling capabilities provide the designer with the ability to communicate with the ICE-49 module and display information on the operation of the prototype system. The ICE-49 module block diagram is shown in Figure 1.

Batch Testing

In conjunction with the ISIS-II diskette operating system, the ICE-49 module can run extensive system diagnostics without operator intervention. The designer or test engineer can define a complete diagnostic exercise, which is stored in a file on the diskette. When activated with an ISIS-II submit command, this file can instruct the ICE-49 module to execute the diagnostic routine and store the results in another file on the diskette. Results are available to the designer at his convenience. In this way, routine diagnostics and long term testing may be done without tying up valuable manpower.

Integrated Hardware/Software Development

The user prototype need consist of no more than an MCS-48 socket and timing logic to begin integration of software and hardware development efforts. Through the ICE-49 module mapping capabilities, Intellec system resources can be accessed to replace prototype memory. Hardware designs can be tested using the system software to drive the final product. Thus, the system integration phase, which can be costly when attempting to mesh completed hardware and software products, becomes a convenient two-way debug tool when begun early in the design cycle.

Real-Time Trace

The ICE-49 module captures trace information while the designer is executing programs in real time. The instructions executed, program counter, port values for bus 0, port 1 and port 2, and the values of selected MCS-48 status lines are stored for the last 255 instruction cycles executed. When retrieved for display, code is disassembled for user convenience. This provides data for determining how the user system was reacting prior to emulation break, and is available whether the break was user initiated or the result of an error condition. For more detailed information on the actions of internal registers, flags, or other system operations, the user may operate in single or multiple step sequences tailored to system debug needs.

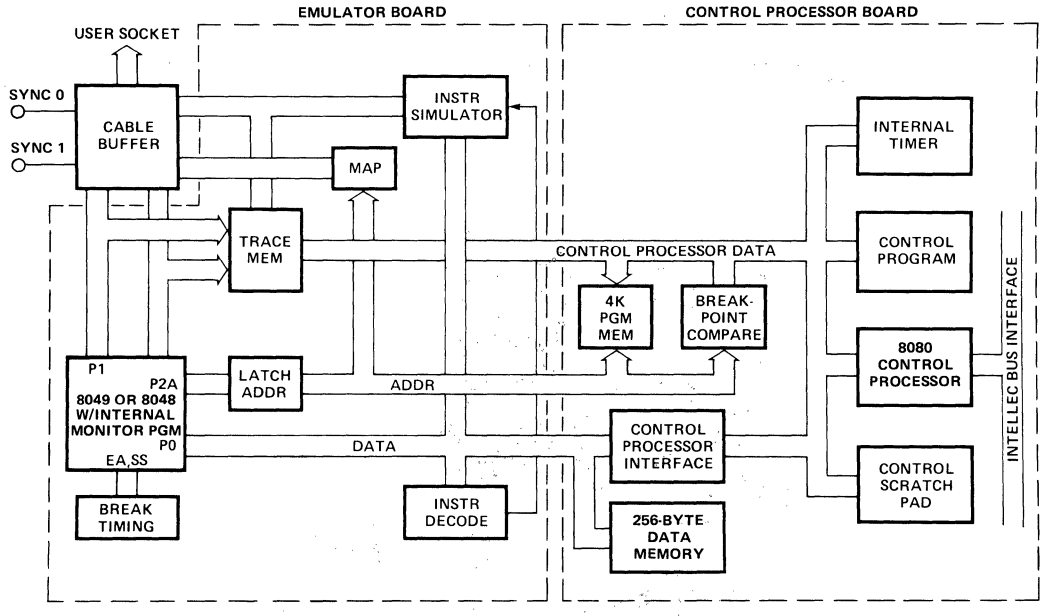


Figure 1. ICE-49 Module Block Diagram

Memory Mapping

The 8049, 8748 and 8048 contain internal program and data memory. Both program and data memory can be expanded using external memory devices.

Internal Memory — When the MCS-48 microcomputer is replaced by the ICE-49 socket in a system, the ICE-49 module supplies static RAM memory as a replacement for the internal microcomputer memory. The ICE-49 module has enough RAM memory available to emulate up to the total 4K control memory capability of the system. The ICE-49 module also provides for up to 384 bytes of data memory.

External Memory — The ICE-49 module separates replacement control memory into sixteen 256-byte blocks. Replacement external data memory consists of one 256-byte block. Each block of memory can be defined separately as supplied by the user system or supplied by the ICE-49 module. The user may assign ICE-49 equivalent memory to take the place of external memory not yet supplied in his system.

Symbolic Debugging

ICE-49 software provides symbolic definition of all MCS-48 registers, flags, and selected MCS-48 pins. Symbolically defined pseudo registers provide access to the sense of MCS-48 flip flops which enable time, counter, interrupt, and flag-0/flag-1 options. In addition, the user may reference locations in program and data memory, or their contents, symbolically. The user symbol table generated along with the object file during a program assembly may be loaded to Intellec memory for access during emulation. The user is encouraged to add to this symbol table any additional symbolic values for memory addresses, constants, or variables he may find useful during system debugging. Symbols may be substituted for numeric values in any of the ICE-49 commands. Symbolic reference is a great advantage to the system designer. He is no longer burdened with the need to recall or look up those addresses of key locations in his program that can change with each assembly. Meaningful symbols from his source program may be used instead. For example, the command:

```
GO FROM .START TILL XDATA. RSLT WRITTEN
```

begins execution of the program at the address referenced by the label START in the designers assembly program. A breakpoint is set to occur the first time the microprocessor writes to the external data memory location referenced by RSLT. The designer does not have to be concerned with the physical locations of START and RSLT. The ICE-49 software driver supplies them automatically from information stored in the symbol table.

Hardware

The ICE-49 module is a microcomputer system utilizing Intel's 8049 or 8048/8748 microcomputer as its nucleus. The 8049 provides the 8049, 8039 emulation characteristics. The 8048/8748 provides the 8748/8048/8035/8021 emulation characteristics. The ICE-49 module uses an Intel 8080 to communicate with the Intellec host processor via a common memory space. The 8080 also controls an internal ICE-49 bus for intramodule communication. ICE-49 hardware consists of two PC boards, the controller board, and the emulator board, all of which reside in the Intellec chassis. A cable interfaces the ICE-49 boards to the MCS-48 system. The cable terminates in a MCS-48 pin compatible plug which replaces any MCS-48 device in the user system. The ICE-49 module block diagram is shown in Figure 1.

Real-Time Trace

Trace Buffer — While the ICE-49 module is executing the user program, it is monitoring port, program counter, data, and status lines. Values for each instruction cycle executed are stored in a 255 x 44 real-time RAM trace buffer. A resetable timer resident on the controller board counts instruction cycles.

Controller Board

The ICE-49 module talks to the Intellec system as a peripheral device. The controller board receives commands from the Intellec system and responds through the parameter block. Three 15-bit hardware breakpoint registers are available for loading by the user. While in emulation mode, a hardware comparator is constantly monitoring address and status lines for a match to terminate an emulation. The breakpoint registers provide a signal when a match is detected. The user may disable the emulation break capability and use the signal to synchronize other debug tools. The controller board returns real-time trace data, MCS-48 register, flag, and pin values, and ICE-49 status information, to a control block in the Intellec system when emulation is terminated. This information is available to the user through the ICE-49 interrogation commands. Error conditions, when present, are automatically displayed on the Intellec system console. The controller board also contains static RAM memory, which can be used to emulate MCS-48 program and data memory in real time. 4K of memory is available in sixteen 256-byte pages to emulate MCS-48 PROM or PROM program memory. A 256-byte page of data memory is available to access in place of MCS-48 external data memory. The controller board address map directs the ICE-49 module to access either replacement ICE-49 memory or actual user system external memory in 256-byte segments based on information provided by the user.

Emulator Board

The emulator board contains the 8049* and peripheral logic required to emulate the MCS-48 device in the user system. A software selectable 6 MHz or 3 MHz clock drives the emulated MCS-48 device. This clock can be disabled and replaced with a user supplied TTL clock in the user system.

*Use 8048 with internal monitor program when emulating 8748/8048/8035/8021.

Cable Card

The cable card is included for cable driving. It transmits address and data bus information to the user system through a 40-pin connector which plugs into the user system in the socket designed for the MCS-48 device.

Software

The ICE-49 software driver is a RAM-based program which provides the user with an easy to use command language (see Table 1, Table 2, and Table 3) for defining breakpoints, initiating real-time emulation or single step operation, and interrogating and altering user system status recorded during emulation. The ICE-49 command language contains a broad range of modifiers to provide the user with maximum flexibility in defining the operation to be performed. The ICE-49 software driver is available on diskette and operates in 32K of Intellec RAM memory.

Command	Operation
Enable	Activates breakpoint and display registers for use with go and step commands.
Go	Initiates real-time emulation and allows user to specify breakpoints and data retrieval.
Step	Initiates emulation in single instruction increments. Each step is followed by register dump. User may optionally tailor other diagnostic activity to his needs.
Interrupt	Emulates user system interrupt.

Table 1. ICE-49 Emulation Commands

Command	Operation
Display	Prints contents of memory, MCS-48 device registers, I/O ports, flags, pins, real-time trace data, symbol table, or other diagnostic data on list device.
Change	Alters contents of memory, register, output port, or flag. Sets or alters breakpoints and display registers.
Map	Defines memory status.
Base	Establishes mode of display for output data.
Suffix	Establishes mode of display input data.

Table 2. ICE-49 Interrogation Commands

Command	Operation
Load	Fetches user symbol table and object code from input device.
Save	Sends user symbol table and object code to output device.
Define	Enters symbol name and value to user symbol table.
Move	Moves block of memory data to another area of memory.
List	Defines list device.
Exit	Returns program control to ISIS-II.
Evaluate	Converts expression to equivalent values in binary, octal, decimal, and hex.
Remove	Deletes symbols from symbol table.
Reset	Reinitializes ICE-49 hardware.

Table 3. ICE-49 Utility Commands

SPECIFICATIONS

ICE-49 Operating Environment

Required Hardware

Intellec microcomputer development system
System console
Intellec diskette operating system
ICE-49 Module

Required Software

System monitor
ISIS-II

Equipment Supplied

Printed circuit boards (control board, emulator board)
Interface cables and buffer module
ICE-49 software, diskette-based version (single density or double density)
8048 with internal monitor program

System Clock

Crystal controlled 6.0 MHz internal, 3.0 MHz internal or user supplied TTL external: software selectable.

Physical Characteristics

Width — 12.00 in. (30.48 cm)
Height — 6.75 in. (17.15 cm)
Depth — 0.50 in. (1.27 cm)
Weight — 8.00 lb. (3.64 kg)

Electrical Characteristics

DC Power Requirements

V_{CC} = +5V ±5%
I_{CC} = 10A max; 7.0A typ
V_{DD} = +12V ±5%
I_{DD} = 79 mA max; 45 mA typ
V_{BB} = −10V ±5%
I_{BB} = 20 mA max

EM1
8021 EMULATION BOARD

EPROM functional equivalent of 8021 — single component 8-bit microcomputer

Based on 8748 — user programmable/ erasable EPROM 8-bit computer

Operates with ICE-49™ module to provide full in-circuit debugging of 8021 prototype system

Connects to prototype system through 8021 pin compatible plug

On-card 3.0 MHz or external TTL driven clock

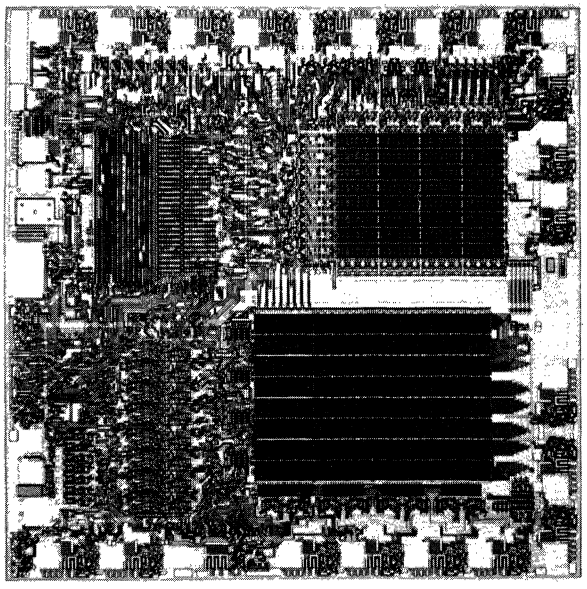
Portable 4" x 7" microcomputer circuit assembly

The EM1 emulator board is a ready-to-use 4" x 7" microcomputer circuit assembly that emulates the Intel 8021 microcomputer. A 12-inch flat-cable assembly connects the board to the 8021 socket in a prototype system. The board is designed so that it can be mounted either as a stand-alone unit, or within the prototype assembly.

The 8021 microcomputer has 1K x 8 mask-programmable ROM program memory and 64 by 8 RAM data memory. The EM1 is controlled by an Intel 8748, with 1K of EPROM program memory and a 64 byte data memory. The EPROM can be programmed and erased repeatedly during hardware and software development. The EM1 has several ancillary circuits that perform the following functions which are specific to the 8021:

Zero crossing detector
Crystal controlled clock/buffer
Port 0 simulator

For prototype debugging, the 8748 can be removed from its socket and replaced with a cable to an ICE-49 module. When used with the EM1, ICE-49 module emulates the 8021 in real-time, or single-steps the 8021 program at the user's command. A full range of capabilities for examining and modifying 8021 memory and status are supplied through ICE-49 module.



HARDWARE

The EM1 emulation board uses the 8748 to perform the emulation.

P0 Simulator

Port 0 of the 8021 is a quasi*-bidirectional port. The P0 simulator converts the data bus of the 8748 into a quasi-bidirectional port.

Crystal Control Clock Buffer

The EM1 allows user to select an on-board oscillator or a TTL clock driven from the 8021 user's prototype system via a Cambion Suitcase jumper.

Jumper	Position	State
W1	A — B	On-Board
	C — D	External TTL Clock

*A bidirectional port which serves as an input port, output port, or both even though outputs are statically latched.

Zero Cross Detection Simulator

The zero cross detection simulator enables the 8748's T1 input to detect zero-crossings. The circuitry provides a high level signal on a positive crossing and a low level signal on a negative crossing of zero to the T1 input of the 8748.

Reset Buffer

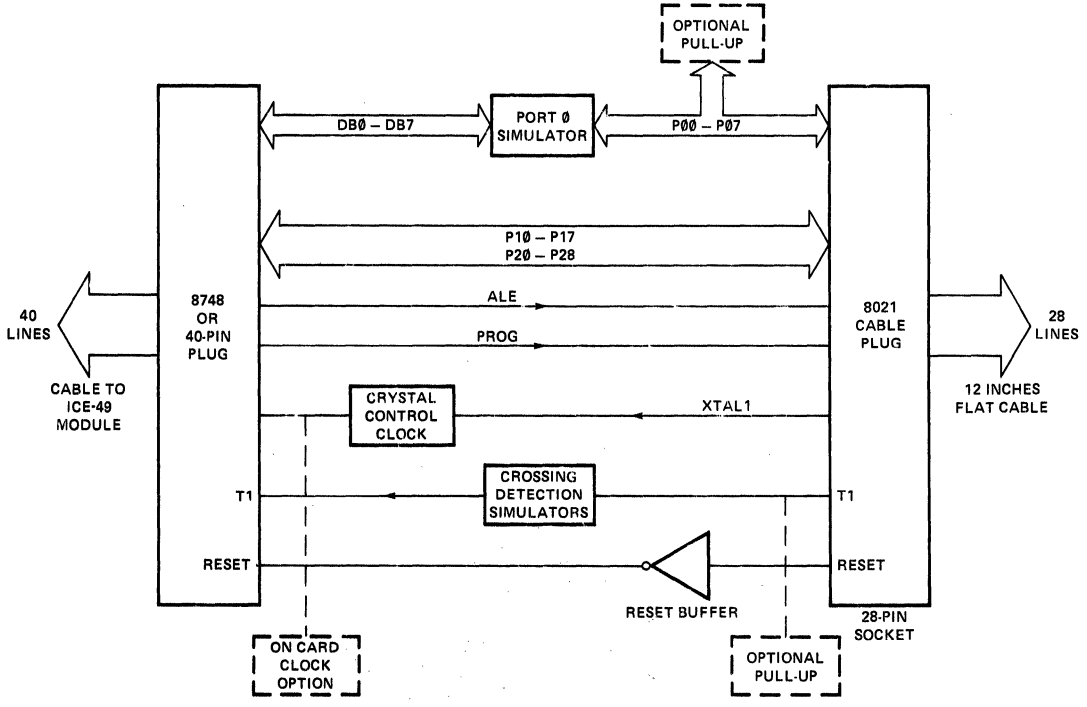
The 8021 resets on a logic HIGH level signal. However, the 8748 resets on a logic LOW level, thus an inverter is provided on the EM1 to make the two chips compatible.

Optional Pull-Ups

Resistors are provided to simulate the optional pull-up resistors on T1 input and Port 0 of the 8021. A removable resistor pack is used on Port 0. The T1 input pull up can be installed by soldering in a 50K resistor.

Software

When emulating the 8021 with EM1, the user must observe the 8021 instruction set.



SPECIFICATIONS

Operating Environment

Stand-Alone
  Required Hardware:
    EM1 emulation board

In-Circuit Emulation
  Required Hardware:
    EM1 emulation board
    Intellec Microcomputer Development System configurated with ICE-49 module

Equipment Supplied

EM1 printed circuit board

12" long flat cable terminating in 28-pin plug, pin compatible with 8021

EM1 Operator's Manual

System Clock

Crystal controlled 3.0 MHz on board or user supplied TTL external clock: hardware jumper selectable.

Physical Characteristics

Width: 7.0 in (17.78 cm)
Height: 4.0 in. (10.16 cm)
Depth: 0.75 in. (1.91 cm)
Weight: < 1.0 lbs. (0.45 kg)

Electrical Characteristics

DC Power:
  V_{CC} 5V ± 5%
  I_{CC} 300 mA (max.)

Environmental Characteristics

Operating Temperature: 0 — 55°C

Operating Humidity: up to 95% relative humidity without condensation

ORDERING INFORMATION

PART NUMBER	Description
MDS-EM1	8021 Emulation Board

EM2
8022 EMULATION BOARD

Portable 4.25″ x 2.75″ microcomputer circuit assembly

Connects directly into prototype system through Intel® 8022* pin compatible socket

Provides Intel® 8755A — 2K x 8 EPROM

EPROM functional and electrical equivalent of Intel® 8022 — single component 8-bit computer

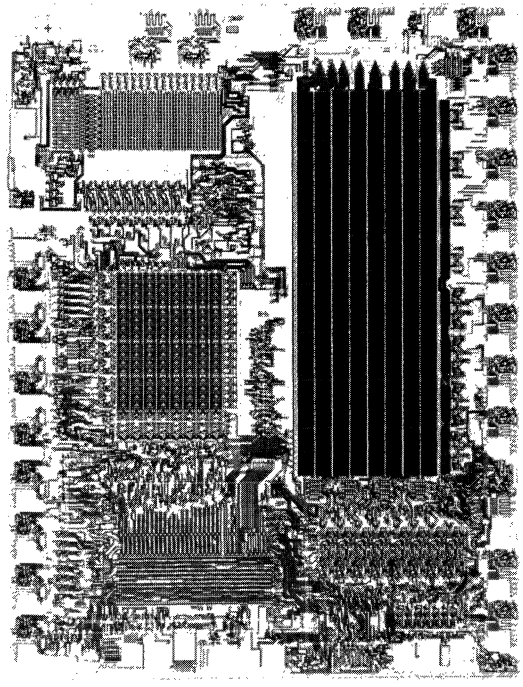
The EM2 emulator board is a ready-to-use 4.25″ x 2.75″ microcomputer circuit assembly that emulates the Intel® 8022 single chip microcomputer. The emulator board is designed to plug directly into the 8022 socket. No interfacing and interconnection cables are necessary. Power is obtained from the user's system.

The EM2 emulator board provides the user a full EPROM functional and electrical equivalent of the 8022 single component 8-bit microcomputer.

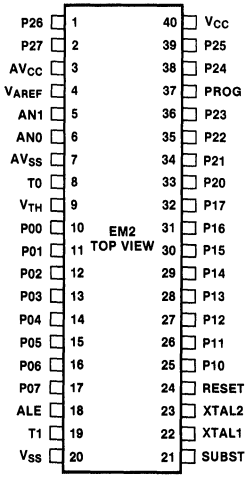
The EM2 emulator board consists of an Intel® 8022 emulator chip and an Intel® 8755A, providing the EM2 emulator board with a 2K x 8 EPROM program memory which can be programmed and erased repeatedly during hardware and software development.

The 8022E emulator chip is a modified version of the 8022 intended for use in design support systems. Instead of using resident ROM memory as the 8022, the 8022E uses an external 2K EPROM 8755A memory for program storage, allowing easy program modification.

*See Intel® 8022 Data Sheet.

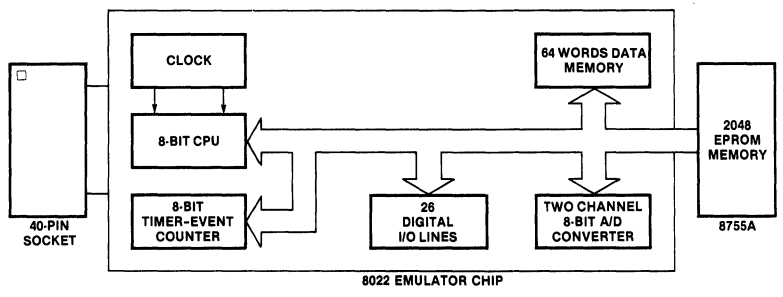


40-PIN SOCKET CONFIGURATION



□ PIN 1
SQUARE SOLDER PAD

EM2 BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential.
V_{CC}	40	+5V circuit power supply.
PROG	37	Output strobe for Intel® 8243 I/O expander.
P00–P07 Port 0	10–17	8-bit open-drain port with comparator inputs. The switching threshold is set externally by V_{TH}. Optional pull-up resistors may be added via ROM mask selection. (The emulator board has switch selection of this option.)
V_{TH}	9	Port 0 threshold reference pin.
P10–P17 Port 1	25–32	8-bit quasi-bidirectional port.
P20–P27	33–36	8-bit quasi-bidirectional port.
Port 2	38–39 1–2	P20–P23 also serve as a 4-bit I/O expander for Intel® 8243.
T0	8	Interrupt input and input pin testable using the conditional transfer instructions JT0 and JNT0. Initiates an interrupt following a low level input if interrupt is enabled. Interrupt is disabled after a reset.
T1	19	Input pin testable using the JT1 and JNT1 conditional transfer instructions. Can be designated the timer/event counter input using the STRT CNT instruction. Also serves as the zero-cross detection input to allow zero-crossover sensing of slowly moving AC inputs. Optional pull-up resistor may be added via ROM mask selection.
RESET	24	Input used to initialize the processor by clearing status flip-flops and setting the program counter to zero.
AV_{SS}	7	A/D converter GND potential. Also establishes the lower limit of the conversion range.
AV_{CC}	3	A/D +5V power supply.
SUBST	21	Substrate pin used with a bypass capacitor to stabilize the substrate voltage and improve A/D accuracy.
V_{AREF}	4	A/D converter reference voltage. Establishes the upper limit of the conversion range.
AN0, AN1	6,5	Analog inputs to A/D converter. Software selectable on-chip via SEL AN0 and SEL AN1 instructions.
ALE	18	Address Latch Enable. Signal occurring once every 30 input input clocks (once every single cycle instruction), used as an output clock.
XTAL1	22	One side of crystal, inductor, or resistor input for internal oscillator. Also input for external frequency source. (Not TTL compatible.)
XTAL2	23	Other side of timing control element. This pin is not connected when an external frequency source is used.

On the EM2 Board:

The Intel® 8755A EPROM can be programmed using any of the modules listed in Table 1.

Module	Description
UPP-103	Universal PROM Programmer. Requires UPP-955, which includes 8755A Personality Card with 40-pin adapter socket.
PROMPT-48	Intellec® MCS-48 Microcomputer Design Aid. Requires PROMPT-475 Programming Adapter.
PROMPT-80/85	Intellec® 8080/8085 Microcomputer Design Aid. Requires PROMPT-975 Programming Adapter.

Table 1. 8755A Proramming Module

The 8755A EPROM is erased when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). Sunlight and certain fluorescent lamps have wavelengths in the 3000Å to 4000Å range. If the 8755A is to be exposed to sunlight or room fluorescent lighting for extended periods, then opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure is exposure to ultraviolet light which has a wavelength of 2537Å. The integrated dose (UV intensity multiplied by exposure time) for erasure should be a minimum of 15W-sec/cm. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000μ W/em^2 power rating. Place the 8755A within one inch of the lamp during erasure. Some lamps include a filter which should be removed before erasure.

SPECIFICATIONS

Operating Environment

Intel® 8755A EPROM Programming

UPP-103
PROMPT-48
PROMPT-80/85

Intellec Microcomputer Development System

Software

8048 Assembler
ISIS-II Diskette Operating System

Equipment Supplied

EM2 Printed Circuit Board
EM2 Reference Manual

Physical Characteristics

Width: 2.75 in. (6.98 cm)
Height: 4.25 in. (10.79 cm)
Depth: 1.5 in. (3.81 cm)
Weight: 0.5 lb (0.23 kg)

Electrical Characteristics

DC Power

$V_{CC} = 5V \pm 5\%$
$I_{CC} = 300$ mA (maximum)

Environmental Characteristics

Operating Temperature — 0 to 55°C

Operating Humidity — Up to 95% relative humidity without condensation

ORDERING INFORMATION

Part Number	Description
MDS-EM2	8022 Emulation Board

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN SALES OFFICES

September 1980

ALABAMA

Intel Corp.
303 Williams Avenue, S.W.
Suite 1422
Huntsville 35801
Tel: (205) 533-9353

ARIZONA

Intel Corp.
10210 N. 25th Avenue, Suite 11
Phoenix 85021
Tel: (602) 997-9695

BFA
4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA

Intel Corp.
7670 Opportunity Rd.
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*
2000 East 4th Street
Suite 100
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel. (213) 986-9510
TWX: 910-495-2045

Intel Corp.*
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
910-338-0255

Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I
558 Valley Way
Calaveras Business Park
Milpitas 95035
Tel: (408) 946-8885

Mac-I
P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I
1321 Centinela Avenue
Suite 1
Santa Monica 90404
Tel: (213) 829-4797

Mac-I
20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO

Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

CONNECTICUT

Intel Corp.
Peacock Alley
36 Padanaram Road
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

FLORIDA

Intel Corp.
1001 N.W. 62nd Street, Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

FLORIDA (cont.)

Intel Corp.
5151 Adanson Street, Suite 203
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

GEORGIA

Intel Corp.
3300 Holcomb Bridge Rd.
Norcross 30092

ILLINOIS

Intel Corp.*
2550 Golf Road, Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

Technical Representatives
1502 North Linde Street
Bloomington 61701
Tel: (309) 829-8080

INDIANA

Intel Corp.
9101 Wesleyan Road
Suite 204
Indianapolis 46268
Tel: (317) 299-0623

IOWA

Technical Representatives, Inc.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52405
Tel: (319) 393-5510

KANSAS

Intel Corp.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

Technical Representatives, Inc.
8245 Nieman Road, Suite 100
Lenexa 66214
Tel: (913) 888-0212, 3, & 4
TWX: 910-749-6412

Technical Representatives, Inc.
360 N. Rock Road
Suite 4
Wichita 67206
Tel: (316) 681-0242

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Mesa Inc.
16021 Industrial Dr.
Gaithersburg 20760
Tel: (301) 948-4350

MASSACHUSETTS

Intel Corp.*
27 Industrial Ave.
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333

EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740
TWX: 922531

MICHIGAN

Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 810-244-4915

MINNESOTA

Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
502 Earth City Plaza
Suite 121
Earth City 63045
Tel: (314) 291-1990

Technical Representatives, Inc.*
502 Earth City Plaza
Suite 201
Earth City 63045
Tel: (314) 291-0001

Technical Representatives, Inc.*
VSW Bldg. Suite 560
406 W. 31st Street
Kansas City 64111
Tel: (816) 756-3575
TWX: 910-771-0025

NEW JERSEY

Intel Corp.*
Raritan Plaza
2nd Floor
Raritan Center
Edison 08817
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO

BFA Corporation
1704 Moon N.E., Suite 7
Las Cruces 87112
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK

Intel Corp.*
300 Motor Pkwy.
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 254-6120
TWX: 510-253-7391

Measurement Technology, Inc.
59 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

Intel Corp.
154 Huffman Mill Rd.
Burlington 27215
Tel: (919) 584-3631

OHIO

Intel Corp.*
6500 Poe Avenue
Dayton 45415
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg., No. 300
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736
TWX: 810-427-9298

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086
TWX: 910-467-8741

PENNSYLVANIA

Intel Corp.*
275 Commerce Dr.
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-661-2077

Intel Corp.*
201 Penn Center Boulevard
Suite 301W
Pittsburgh 15235
Tel: (412) 823-4970

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS

Intel Corp.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5617

Intel Corp.*
6420 Richmond Ave.
Suite 280
Houston 77057
Tel: (713) 784-3400
TWX: 910-881-2490

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

WASHINGTON

Intel Corp.
Suite 114, Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086
TWX: 910-443-3002

WISCONSIN

Intel Corp.
150 S. Sunnyslope Rd.
Brookfield 53005
Tel: (414) 784-9060

CANADA

Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W 4Y5
Tel: (416) 675-2105
TELEX: 06983574

Multilek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multilek, Inc.
Toronto
Tel: (416) 245-4622

Multilek, Inc.
Montreal
Tel: (514) 481-1350

*Field Application Location

intel
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

September 1980

ALABAMA

†Hamilton/Avnet Electronics
4812 Commercial Drive N.W.
Huntsville 35805
Tel: (205) 837-7210

†Pioneer/Huntsville
1207 Putman Drive NW
Huntsville 35805
Tel: (205) 837-9033
TWX: 810-726-2197

ARIZONA

†Hamilton/Avnet Electronics
2615 S. 21st Street
Phoenix 85034
Tel: (602) 275-7851

†Wyle Distribution Group
8155 N. 24th Avenue
Phoenix 85021
Tel: (602) 995-9185
TWX: 910-951-4282

CALIFORNIA

Arrow Electronics, Inc.
9511 Ridge Haven Court
San Diego 92123
Tel: (714) 565-4800

Arrow Electronics, Inc.
720 Palomar Avenue
Sunnyvale, California 94086
Tel: (408) 739-3011
TWX: 910-339-9371

†Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6051
TWX: 910-595-1928

Hamilton/Avnet Electronics
1175 Bordeaux Dr.
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
8917 Complex Drive
San Diego 92123
Tel: (714) 571-7923
TWX: 910-335-1216

†Hamilton/Avnet Electronics
10912 W. Washington Blvd.
Culver City 90230
Tel: (213) 558-2193
TWX: 910-340-6364 or 7073

†Hamilton Electro Sales
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4100
TWX: 910-595-2638

†Wyle Distribution Group
124 Maryland Street
El Segundo 90245
Tel: (213) 322-3826
TWX: 910-348-7140 or 7111

†Wyle Distribution Group
9525 Chesapeake Dr.
San Diego 92123
Tel: (714) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95052
Tel: (408) 727-2500
TWX: 910-338-0451 or 0296

Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 641-1611

COLORADO

†Wyle Distribution Group
6777 E. 50th Avenue
Commerce City 80022
Tel: (303) 287-9611
TWX: 910-931-0510

†Hamilton/Avnet Electronics
8765 E. Orchard Road
Suite 708
Englewood 80111
Tel: (303) 534-1212
TWX: 910-931-0510

COLORADO (cont.)

Wyle Distribution Group
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-WYLE
TWX: 910-931-0510

CONNECTICUT

†Arrow Electronics
12 Beaumont Road
Wallingford 06512
Tel: (203) 265-7741
TWX: 710-476-0162

†Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Harvey Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
TWX: 710-468-3373
TWX: 710-393-6770

FLORIDA

†Arrow Electronics
1001 N.W. 62nd Street
Suite 108
Ft. Lauderdale 33309
Tel: (305) 776-7790
TWX: 510-955-9456

†Arrow Electronics
115 Palm Bay Road, NW
Suite 10, Bldg. 200
Palm Bay 32905
Tel: (305) 725-1480
TWX: 510-959-6337

†Hamilton/Avnet Electronics
6800 Northwest 20th Ave.
Ft. Lauderdale 33309
Tel: (305) 971-2900
TWX: 510-955-3097

Hamilton/Avnet Electronics
3197 Tech. Drive North
St. Petersburg 33702
Tel: (813) 576-3930
TWX: 810-863-0374

†Pioneer/Orlando
6220 S. Orange Blossom Trail
Suite 412
Orlando 32809
Tel: (305) 859-3600
TWX: 810-850-0177

GEORGIA

Arrow Electronics
2979 Pacific Drive
Norcross 30071
Tel: (404) 449-8252
TWX: 810-757-4213

†Hamilton/Avnet Electronics
6700 I-85 Access Road, No. 11
Suite 1E
Norcross 30071
Tel: (404) 448-0800

ILLINOIS

Arrow Electronics
492 Lunt Avenue
P.O. Box 94248
Schaumburg 60172
Tel: (312) 893-9420
TWX: 910-222-1807

†Hamilton/Avnet Electronics
3901 No. 25th Avenue
Schiller Park 60176
Tel: (312) 678-6310
TWX: 910-227-0060

Pioneer/Chicago
1551 Carmen Drive
Elk Grove 60007
Tel: (312) 437-9680
TWX: 910-222-1834

INDIANA

†Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel: (317) 844-9333

Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300
TWX: 810-260-1794

KANSAS

†Hamilton/Avnet Electronics
9219 Quivira Road
Overland Park 66215
Tel: (913) 888-8900

†Component Specialties, Inc.
8369 Nieman Road
Lenexa 66214
Tel: (913) 492-3555

MARYLAND

Arrow Electronics, Inc.
4801 Benson Avenue
Baltimore 21227
Tel: (301) 247-5200

†Hamilton/Avnet Electronics
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5684
TWX: 710-862-1861

†Pioneer/Washington
9100 Gaither Road
Gaithersburg 20760
Tel: (301) 948-0710
TWX: 710-828-0545

MASSACHUSETTS

†Hamilton/Avnet Electronics
50 Tower Office Park
Woburn 01801
Tel: (617) 273-7500
TWX: 710-393-0382

†Arrow Electronics
96D Commerce Way
Woburn 01801
Tel: (617) 933-8130
TWX: 710-393-6770

Harvey/Boston
44 Hartwell Ave.
Lexington 02173
Tel: (617) 861-9200
TWX: 710-326-6617

MICHIGAN

†Arrow Electronics
3810 Varsity Drive
Ann Arbor 48104
Tel: (313) 971-8220
TWX: 810-223-6020

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
TWX: 810-242-3271

†Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

MINNESOTA

†Arrow Electronics
5230 W. 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-756-2726

†Industrial Components
5229 Edina Industrial Blvd.
Minneapolis 55435
Tel: (612) 831-2666
TWX: 910-756-3153

†Hamilton/Avnet Electronics
7449 Cahill Road
Edina 55435
Tel: (612) 941-3801
TWX: 910-576-2720

MISSOURI

†Hamilton/Avnet Electronics
13743 Shoreline Ct.
Earth City, 63045
Tel: (314) 344-1200
TWX: 910-762-0606

NEW HAMPSHIRE

†Arrow Electronics
1 Perimeter Drive
Manchester 03103
Tel: (603) 668-6968
TWX: 710-220-1684

NEW JERSEY

†Arrow Electronics
Pleasant Valley Avenue
Moorestown 08057
Tel: (215) 928-1800
TWX: 710-897-0829

†Arrow Electronics
285 Midland Avenue
Saddle Brook 07662
Tel: (201) 797-5800
TWX: 710-998-2206

†Hamilton/Avnet Electronics
1 Keystone Ave.
Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0100
TWX: 710-897-1405

†Harvey Electronics
45 Route 46
Pinebrook 07058
Tel: (201) 227-1262
TWX: 710-734-4382

Hamilton/Avnet Electronics
10 Industrial Road
Fairfield 07006
Tel: (201) 575-3390
TWX: 710-734-4438

NEW MEXICO

†Alliance Electronics Inc.
11030 Cochiti S.E.
Albuquerque 87123
Tel: (505) 292-3360
TWX: 910-989-1151

†Hamilton/Avnet Electronics
2524 Baylor Drive, S.E.
Albuquerque 87119
Tel: (505) 765-1500

NEW YORK

†Arrow Electronics
3000 South Winton Road
Rochester 14623
Tel: (716) 275-0300
TWX: 510-253-4766

†Arrow Electronics
7705 Maltlage Drive
Liverpool 13088
Tel: (315) 652-1000
TWX: 710-545-0230

Arrow Electronics
20 Oser Avenue
Hauppauge 11787
Tel: (516) 231-1000
TWX: 510-227-6623

†Hamilton/Avnet Electronics
333 Metro Park
Rochester 14623
Tel: (716) 475-9130
TWX: 510-253-5470

†Hamilton/Avnet Electronics
16 Corporate Circle
E. Syracuse 13057
Tel: (315) 437-2641

†Hamilton/Avnet Electronics
5 Hub Drive
Melville, Long Island 11746
Tel: (516) 454-6000
TWX: 510-252-0893

†Microcomputer System Technical Demonstrator Centers

intel
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

September 1980

NEW YORK (cont.)

Harvey Electronics
P.O. Box 1208
Binghampton 13902
Tel: (607) 748-8211
TWX: 510-252-0893

†Harvey Electronics
60 Crossways Park West
Woodbury 11797
Tel: (516) 921-8700
TWX: 510-221-2184

Harvey/Rochester
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
TWX: 510-253-7001

NORTH CAROLINA

Arrow Electronics
938 Burke Street
Winston-Salem 27102
Tel: (919) 725-8711
TWX: 510-922-4765

Pioneer/Carolina
106 Industrial Ave.
Greensboro 27406
Tel: (919) 273-4441
TWX: 510-925-1114

†Hamilton/Avnet Electronics
2803 Industrial Drive
Raleigh 27609
Tel: (919) 829-8030

OHIO

Arrow Electronics
7620 McEwen Road
Centerville 45459
Tel: (513) 435-5563
TWX: 810-459-1611

Arrow Electronics
6238 Cochran Rd.
Solon 44139
Tel: (216) 248-3990
TWX: 810-427-9409

Arrow Electronics
10 Knollcrest Dr.
Cincinnati 45237
Tel: (513) 761-5432
TWX: 810-461-2670

†Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 433-0610
TWX: 910-340-2531

†Hamilton/Avnet Electronics
4588 Emery Industrial Parkway
Warrensville Heights 44128
Tel: (216) 831-3500
TWX: 810-427-9452

†Pioneer/Dayton
1900 Troy Street
Dayton 45404
Tel: (513) 236-9900
TWX: 810-459-1622

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
TWX: 810-422-2210

OKLAHOMA

†Components Specialties, Inc.
7920 E. 40th Street
Tulsa 74145
Tel: (918) 664-2820
TWX: 910-845-2215

OREGON

†Almac/Stroum Electronics
8022 S.W. Nimbus, Bldg. 7
Beaverton 97005
Tel: (503) 641-9070

†Hamilton/Avnet Electronics
6024 SW Jean Rd.
Bldg. C, Suite 10
Lake Oswego 97034
Tel: (503) 635-7848

PENNSYLVANIA

†Arrow Electronics
4297 Greensburg Pike
Suite 3114
Pittsburgh 15221
Tel: (412) 351-4000

Pioneer/Pittsburgh
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
TWX: 710-795-3122

Pioneer/Delaware Valley
261 Gibraltar Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TEXAS

Arrow Electronics
13715 Gamma Road
Dallas 75234
Tel: (214) 386-7500
TWX: 910-861-5495

Arrow Electronics, Inc.
10700 Corporate Drive, Suite 100
Stafford 77477
Tel: (713) 491-4100

Component Specialties Inc.
8222 Jamestown Drive
Suite 115
Austin 78758
Tel: (512) 837-8922
TWX: 910-874-1320

†Component Specialties, Inc.
10807 Shady Trail, Suite 101
Dallas 75220
Tel: (214) 357-6511
TWX: 910-861-4999

†Component Specialties, Inc.
8585 Commerce Park Drive, suite 590
Houston 77036
Tel: (713) 771-7237
TWX: 910-881-2422

Hamilton/Avnet Electronics
2401 Rutland
Austin 78758
Tel: (512) 837-8911

†Hamilton/Avnet Electronics
2111 W. Walnut Hill Lane
Irving 75062
Tel: (214) 661-4111
TWX: 910-860-5371

†Hamilton/Avnet Electronics
3939 Ann Arbor Drive
Houston 77063
Tel: (713) 780-1771

UTAH

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel: (801) 972-2800

WASHINGTON

†Almac/Stroum Electronics
5811 Sixth Ave. South
Seattle 98108
Tel: (206) 763-2300
TWX: 910-444-2067

Arrow Electronics, Inc.
Electronics Distribution Division
1059 Andover Park East
Tukwila 98188
Tel: (206) 575-0907

†Hamilton/Avnet Electronics
14212 N.E. 21st Street
Bellevue 98005
Tel: (206) 453-5844

WASHINGTON (cont.)

†Wyle Distribution Group
1750 132nd Avenue NE
Bellevue 98005
Tel: (206) 453-8300
TWX: 910-443-2526

WISCONSIN

†Arrow Electronics
430 W. Rawson Avenue
Oak Creek 53154
Tel: (414) 764-6600
TWX: 910-338-0026

†Hamilton/Avnet Electronics
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510
TWX: 910-262-1182

CANADA

ALBERTA

†L.A. Varah Ltd.
4742 14th Street N.E.
Calgary T2D 6L7
Tel: (403) 230-1235
TWX: 018-258-97

Zentronics
9224 27th Avenue
Edmonton T6N 1B2
Tel: (403) 463-3014

Zentronics
3651 21st N.E.
Calgary T2E 6T5
Tel: (403) 230-1422

BRITISH COLUMBIA

†L.A. Varah Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068

Zentronics
550 Cambie St.
Vancouver V6B 2N7
Tel: (604) 688-2533
TWX: 04-5077-89

MANITOBA

L.A. Varah
1-1832 King Edward Street
Winnipeg R2R 0N1
Tel: (204) 633-6190
TWX: 07-55-365

Zentronics
590 Berry St.
Winnipeg R3H 0S1
Tel: (204) 775-8661

ONTARIO

†Hamilton/Avnet Electronics
3688 Rexwood Road, Units G & H
Mississauga L4V 1M5
Tel: (416) 677-7432
TWX: 610-492-8860

†Hamilton/Avnet Electronics
1735 Courtwood Crescent
Ottawa K2C 3J2
Tel: (613) 226-1700
TWX: 053-4971

†L.A. Varah, Ltd.
505 Kenora Avenue
Hamilton L8E 3P2
Tel: (416) 561-9311
TWX: 061-8349

†Zentronics
141 Catherine Street
Ottawa K2P 1C3
Tel: (613) 238-6411
TWX: 053-3636

†Zentronics
1355 Meyerside Drive
Mississauga, Ontario L5T 1C9
Tel: (416) 676-9000
Telex: 06-983-657

QUEBEC

†Hamilton/Avnet Electronics
2670 Sabourin Street
St. Laurent H4S 1M2
Tel: (514) 331-6443
TWX: 610-421-3731

Zentronics
5010 Pare Street
Montreal H4P 1P3
Tel: (514) 735-5361
TWX: 05-827-535

†Microcomputer System Technical Demonstrator Centers

intel
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

INTERNATIONAL SALES AND MARKETING OFFICES

September 1980

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

Micro Sistemas S.A.
9 De Julio 561
Cordoba
Tel: 54-51-32-880
TELEX: 51837 BICCO

AUSTRALIA

A.J.F. Systems & Components Pty. Ltd.
310 Queen Street
Melbourne
Victoria 3000
Tel:
TELEX:

Warburton Franki
Corporate Headquarters
372 Eastern Valley Way
Chatswood, New South Wales 2067
Tel: 407-3261
TELEX: AA 21299

AUSTRIA

Bacher Elektronische Geraete GmbH
Rotenmulgasse 26
A 1120 Vienna
Tel: (0222) 83 63 96
TELEX: (01) 1532

Rekirsch Elektronik Geraete GmbH
Lichtensteinstrasse 97
A1000 Vienna
Tel: (222) 347646
TELEX: 74759

BELGIUM

Inelco Belgium S.A.
Ave. des Croix de Guerre 94
B1120 Brussels
Tel: (02) 216 01 60
TELEX: 25441

BRAZIL

Icotron S.A.
0511-Av. Mutinga 3650
6 Andar
Pirituba-Sao Paulo
Tel: 261-0211
TELEX: (011) 222 ICO BR

CHILE

DIN
Av. Vic. Mc kenna 204
Casilla 6055
Santiago
Tel: 227 564
TELEX: 3520003

CHINA

C.M. Technologies
525 University Avenue
Suite A-40
Palo Alto, CA 94301

COLOMBIA

International Computer Machines
Carrera 7 No. 72-34
Apdo-Aereo 19403
Bogota 1
Tel: 211-7282
TELEX: 71314 INCO

CYPRUS

Cyprus Eltrom Electronics
P.O. Box 5393
Nicosia
Tel: 21-27982

DENMARK

STL-Lyngso Komponent A/S
Ostmarken 4
DK-2860 Soborg
Tel: (01) 67 00 77
TELEX: 22990

Scandinavian Semiconductor
Supply A/S
Nannasgade 18
DK-2200 Copenhagen
Tel: (01) 83 50 90
TELEX: 19037

FINLAND

Oy Fintronic AB
Melkonkatu 24 A
SF-00210
Helsinki 21
Tel: 0-692 6022
TELEX: 124 224 Ftron SF

FRANCE

Celdis S.A.*
53, Rue Charles Frerot
F-94250 Gentilly
Tel: (1) 581 00 20
TELEX: 200 485

Feutrier
Rue des Trois Glorieuses
F-42270 St. Priest-en-Jarez
Tel: (77) 74 67 33
TELEX: 300 0 21

Metrologie*
La Tour d'Asnieres
4, Avenue Laurent Cely
92606-Asnieres
Tel: 791 44 44
TELEX: 611 448

Tekelec Airtronic*
Cite des Bruyeres
Rue Carle Vernet
F-92310 Sevres
Tel: (1) 534 75 35
TELEX: 204552

GERMANY

Electronic 2000 Vertriebs GmbH
Neumarkter Strasse 75
D-8000 Munich 80
Tel: (089) 434061
TELEX: 522561

Jermyn GmbH
Postfach 1180
D-6077 Camberg
Tel: (06434) 231
TELEX: 484426

Kontron Elektronik GmbH
Breslauerstrasse 2
8057 Eching B
D-8000 Munich
Tel: (89) 319.011
TELEX: 522122

Neye Enatechnik GmbH
Schillerstrasse 14
D-2085 Quickborn-Hamburg
Tel: (04106) 6121
TELEX: 02-13590

GREECE

American Technical Enterprises
P.O. Box 156
Athens
Tel: 30-1-8811271
30-1-8219470

HONG KONG

Schmidt & Co.
28/F Wing on Center
Connaught Road
Hong Kong
Tel: 5-455-644
TELEX: 74766 Schmc Hx

INDIA

Micronic Devices
104/109C, Nirmal Industrial Estate
Sion (E)
Bombay 400022, India
Tel: 486-170
TELEX: 011-5947 MDEV IN

ISRAEL

Eastronics Ltd.*
11 Rozanis Street
P.O. Box 39300
Tel Aviv 61390
Tel: 475151
TELEX: 33638

ITALY

Eledra 3S S.P.A.*
Viale Elvezia, 18
I 20154 Milan
Tel: (02) 34.93.041-31.85.441
TELEX: 332332

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. Room 407
2-14-1 Asano, Kokura
Kita-Ku, Kitokyushu City 802
Tel: (093) 511-6471
TELEX: AECKY 7126-16

Hamilton-Avnet Electronics Japan Ltd.
YU and YOU Bldg. 1-4 Horidome-Cho
Nihonbashi
Tel: (03) 662-9911
TELEX: 2523774

Ryoyo Electric Corp.
Konwa Bldg.
1-12-22, Tsukiji, 1-Chome
Chuo-Ku, Tokyo 104
Tel: (03) 543-7711

Tokyo Electron Ltd.
No. 1 Higashikata-Machi
Midori-Ku, Yokohama 226
Tel: (045) 471-8811
TELEX: 781-4473

KOREA

Koram Digital
Room 909 Woonam Bldg.
7, 1-KA Bongre-Dong
Chung-Ku Seoul
Tel: 23-8123
TELEX: K23542 HANSINT

Leewood International, Inc.
C.P.O. Box 4046
112-25, Sokong-Dong
Chung-Ku, Seoul 100
Tel: 28-5927
CABLE: "LEEWOOD" Seoul

MEXICO

Proveedora Electronica, S.A. (Proesa)
Prol. Moctezuma Ote. 24
Col. Romero de Terreros
Apdo. Postal 21-139
Mexico 21, D.F.
TELEX: 017-72402 SAULME

NETHERLANDS

Inelco Nether. Comp. Sys. BV
Turfstekerstraat 63
Aalsmeer 1431 D
Tel: (2977) 28855
TELEX: 14693

Koning & Hartman
Koperwerf 30
2544 EN Den Haag
Tel: (70) 210.101
TELEX: 31528

NEW ZEALAND

W. K. McLean Ltd.
P.O. Box 18-065
Glenn Innes, Auckland, 6
Tel: 587-037
TELEX: NZ2763 KOSFY

NORWAY

Nordisk Elektronik (Norge) A/S
Postoffice Box 122
Smedsvingen 4
1364 Hvalstad
Tel: 02 78 62 10
TELEX: 17546

PORTUGAL

Ditram
Componentes E Electronica LDA
Av. Miguel Bombarda, 133
Lisboa 1
Tel: (19) 545313
TELEX: 14347 GESPIC

SINGAPORE

General Engineers Associates
Blk 3, 1003-1008, 10th Floor
P.S.A. Multi-Storey Complex
Telok Blangah/Pasir Panjang
Singapore 5
Tel: 271-3163
TELEX: RS23987 GENERCO

SOUTH AFRICA

Electronic Building Elements
Pine Square
18th Street
Hazelwood, Pretoria 0001
Tel: 789 221
TELEX: 30181SA

SPAIN

Interface S.A.
Ronda San Pedro 22, 3°
Barcelona 10
Tel: 301 78 51
TWX: 51508

ITT SESA
Miguel Angel 16
Madrid 10
Tel: (1) 4190957
TELEX: 27707/27461

SWEDEN

AB Gosta Backstrom
Box 12009
10221 Stockholm
Tel: (08) 541 080
TELEX: 10135

Nordisk Electronik AB
Box 27301
S-10254 Stockholm
Tel: (08) 635040
TELEX: 10547

SWITZERLAND

Industrade AG
Gemsenstrasse 2
Postcheck 80 - 21190
CH-8021 Zurich
Tel: (01) 60 22 30
TELEX: 56788

TAIWAN

Taiwan Automation Co.*
3d Floor #75, Section 4
Nanking East Road
Taipei
Tel: 771-0940
TELEX: 11942 TAIAUTO

TURKEY

Turkelek Electronics
Apapurk Boulevard 169
Ankara
Tel: 189483

UNITED KINGDOM

Comway Microsystems Ltd.
Market Street
68-Bracknell, Berkshire
Tel: (344) 51654
TELEX: 847201

G.E.C. Semiconductors Ltd.
East Lane
North Wembley
Middlesex HA9 7PP
Tel: (01) 904-9303/908-4111
TELEX: 28817

Jermyn Industries
Vestry Estate
Sevenoaks, Kent
Tel: (0732) 501.44
TELEX: 95142

Rapid Recall, Ltd.
6 Soho Mills Ind. Park
Wooburn Green
Bucks, England
Tel: (6285) 24961
TELEX: 849439

Sintrom Electronics Ltd.*
Arkwright Road 2
Reading, Berkshire RG2 0LS
Tel: (0734) 85464
TELEX: 847395

VENEZUELA

Componentes y Circuitos
Electronicos TTLCA C.A.
Apartado 3223
Caracas 101
Tel: 718-100
TELEX: 21795 TELETIPOS

*Field Application Location

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

INTERNATIONAL SALES AND MARKETING OFFICES

September 1980

INTEL® MARKETING OFFICES

AUSTRALIA

Intel Semiconductor Pty., Ltd.
Suite 2, Level 15, North Point
100 Miller Street
North Sydney, NSW, 2060
Tel: 450-847
TELEX: AA 20097

BELGIUM

Intel Corporation S.A.
Rue du Moulin a Papier 51
Boite 1
B-1160 Brussels
Tel: (02) 660 30 10
TELEX: 24814

DENMARK

Intel Denmark A/S*
Lyngbyvej 32 2nd Floor
DK-2100 Copenhagen East
Tel: (01) 18 20 00
TELEX: 19567

FINLAND

Intel Finland OY
Sentnerikuja 3
SF - 00400 Helsinki 40
Tel: (0) 558531
TELEX: 123 332

FRANCE

Intel Corporation, S.A.R.L.*
5 Place de la Balance
Silic 223
94528 Rungis Cedex
Tel: (01) 687 22 21
TELEX: 270475

GERMANY

Intel Semiconductor GmbH*
Seidlstrasse 27
8000 Muenchen 2
Tel: (089) 53 891
TELEX: 523 177

Intel Semiconductor GmbH
Mainzer Strasse 75
6200 Wiesbaden 1
Tel: (06121) 700874
TELEX: 04186183

Intel Semiconductor GmbH
Wernerstrasse 67
P.O. Box 1460
7012 Fellbach
Tel: (0711) 580082
TELEX: 7254826

Intel Semiconductor GmbH
Hohenzollern Strasse 5
3000 Hannover 1
Tel: (0511) 327081
TELEX: 923625

Intel Semiconductor GmbH
Oberrathstrasse 2
4000 Duesseldorf 30
Tel: (0211) 651054-6
TELEX: 8586977

HONG KONG

Intel Semiconductor Ltd.
99-105 Des Voeux Rd., Central
18F, Unit B
Hong Kong
Tel: 5-450-847
TELEX: 63869

ISRAEL

Intel Semiconductor Ltd.*
P.O. Box 2404
Haifa
Tel: 972/452 4261
TELEX: 92246511

ITALY

Intel Corporation Italia, S.p.A.
Corso Sempione 39
I-20145 Milano
Tel: 2/34.93287
TELEX: 311271

JAPAN

Intel Japan K.K.*
Flower Hill-Shinmachi East Bldg.
1-23-9, Shinmachi, Setagaya-ku
Tokyo 154
Tel: (03) 426-9261
TELEX: 781-28426

NETHERLANDS

Intel Semiconductor B.V.
Cometongebouw
Westblaak 106
3012 Km Rotterdam
Tel: (10) 149122
TELEX: 22283

NORWAY

Intel Norway A/S
P.O. Box 92
Hvamveien 4
N-2013
Skjetten
Tel: (2) 742 420
TELEX: 18018

SWEDEN

Intel Sweden A.B.*
Box 20092
Alpvagen 17
S-16120 Bromma
Tel: (08) 98 53 90
TELEX: 12261

SWITZERLAND

Intel Semiconductor A.G.
Forchstrasse 95
CH 8032 Zurich
Tel: 1-55 45 02
TELEX: 557 89 ich ch

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.*
5 Hospital Street
Nantwich, Cheshire CW5 5RE
Tel: (0270) 62 65 60
TELEX: 36620

Intel Corporation (U.K.) Ltd.
Dorcan House
Eldine Drive
Swindon, Wiltshire SN3 3TU
Tel: (0793) 26101
TELEX: 444447 INT SWN

*Field Application Location

intel
3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN SERVICE OFFICES

September 1980

CALIFORNIA

Intel Corp.
1601 Old Bayshore Hwy.
Suite 345
Burlingame 94010
Tel: (415) 692-4762
TWX: 910-375-3310

Intel Corp.
2000 E. 4th Street
Suite 100
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.
7670 Opportunity Road
San Diego 92111
Tel: (714) 268-3563

Intel Corp.
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086

Intel Corp.
5530 Corbin Avenue
Suite 120
Tarzana 91356
Tel: (213) 986-9510

COLORADO

Intel Corp.
650 South Cherry
Suite 720
Denver, 80222
Tel: (303) 321-8086
TWX: 910-931-2289

FLORIDA

Intel Corp.
1001 N.W. 62nd Street
Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

ILLINOIS

Intel Corp.
2550 Golf Road
Suite 815
Rolling Meadows 60008
Tel: (312) 981-7230
TWX: 910-253-1825

KANSAS

Intel Corp.
9393 W. 110th Street
Suite 265
Overland Park 66210
Tel: (913) 642-8080

MARYLAND

Intel Corp.
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

MASSACHUSETTS

Intel Corp.
27 Industrial Avenue
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333

MICHIGAN

Intel Corp.
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 810-244-4915

MINNESOTA

Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

MISSOURI

Intel Corp.
502 Earth City Plaza
Suite 121
Earth City 63045
Tel: (314) 291-1990

NEW JERSEY

Intel Corp.
2450 Lemoine Avenue
Ft. Lee 07024
Tel: (201) 947-6267
TWX: 710-991-8593

OHIO

Intel Corp.
Chagrin-Brainard Bldg. #210
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736
TWX: 810-427-9298

Intel Corp.
6500 Poe Avenue
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528

OREGON

Intel Corp.
10700 S.W. Beaverton-Hillsdale Hwy.
Bldg. 2, Suite 324
Beaverton 97005
Tel: (503) 641-8086
TWX: 910-467-8741

PENNSYLVANIA

Intel Corp.
275 Commerce Drive
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-651-2077

TEXAS

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628
TWX: 910-874-1347

Intel Corp.
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-2820
TWX: 910-860-5617

Intel Corp.
6420 Richmond Avenue
Suite 280
Houston 77057
Tel: (713) 784-1300
TWX: 910-881-2490

VIRGINIA

Intel Corp.
7700 Leesburg Pike
Suite 412
Falls Church 22043
Tel: (703) 734-9707
TWX: 710-931-0625

WASHINGTON

Intel Corp.
1603 116th Ave. N.E.
Suite 114
Bellevue 98005
Tel: (206) 232-7823
TWX: 910-443-3002

WISCONSIN

Intel Corp.
150 S. Sunnyslope Road
Suite 148
Brookfield 53005
Tel: (414) 784-9060

CANADA

Intel Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W4Y5
Tel: (416) 675-2105
Telex: 069-83574

Intel Corp.
39 Hwy. 7, Bells Corners
Ottawa, Ontario
K2H 8R2
Tel: (613) 829-9714
Telex: 053-4115

*Field Application Location

intel

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Intel Corporation
Rue du Moulin à Papier 51, Boite 1,
B-1160 Brussels, Belgium

Intel Corporation
Flower Hill-Shinmachi East Bldg.
1-23-9, Shinmachi, Setagayu-ku
Tokyo 154, Japan

Printed in U.S.A./980/28K/R.R.D KD