ALTOS

486

System Reference

486
System Reference

ACKNOWLEDGEMENTS

CP/M is a registered trademark of Digital Research, Inc.

XENIX is a trademark of Microsoft Inc.

UNIX is a trademark of Bell Laboratories

FEDERAL COMMUNICATIIONS COMMISSION NOTICE

WARNING

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions manual, may cause interference to radio communications.

It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment.

Operation of this equipment in a residential area is likely to cause interference in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

About This Manual

This manual is written for a system programmer or analyst requiring hardware and software reference information for system configuration and program modification.

Contents

ILLUSTRATIONS

TABLE OF CONTENTS

PAGE SUBJECT

TABLES

System Overview 1

FEATURES AND CAPABILITIES

The Altos 486 is a general purpose 16-bit microcomputer system with a 1 Mbyte floppy disk and 2Ø Mbyte hard disk storage capacity in a desktop design. See Figure 1-1. The 486 supports up to four users and a serial printer. The WorkNet port lets you network the 486 system with up to 3Ø other Altos computer systems (such as the 186, 586/986, 586T/986T, and the 68ØØØ).

The 486 combines the low cost data sharing and communications capability of an intelligent terminal with the computer power necessary to tackle sophisticated business and scientific problems. The adjustable, low-glare CRT offers a high-resolution display. The adjustable, low-profile keyboard offers 1Ø5 keys in the format of a dedicated word processor.

The 486 system provides the following features:

- The Intel 8Ø186 microprocessor with expanded instruction set, built-in direct memory access channels and an interrupt controller and operates at 8Mhz.
- Z-8ØA I/O processor for high speed data transfer.
- 512 Kbytes of standard main memory; expandable to 896 Kbytes.
- 5-1/4 inch, double-sided, double density floppy disks with 1 Mbyte storage capacity.
- Network communications using the WorkNet interface.
- Five RS232 serial ports.
- Optional 2Ø Mbyte add-on hard disk for a total of 4Ø Mbytes of hard disk storage capacity.
- Comprehensive system diagnostics support.

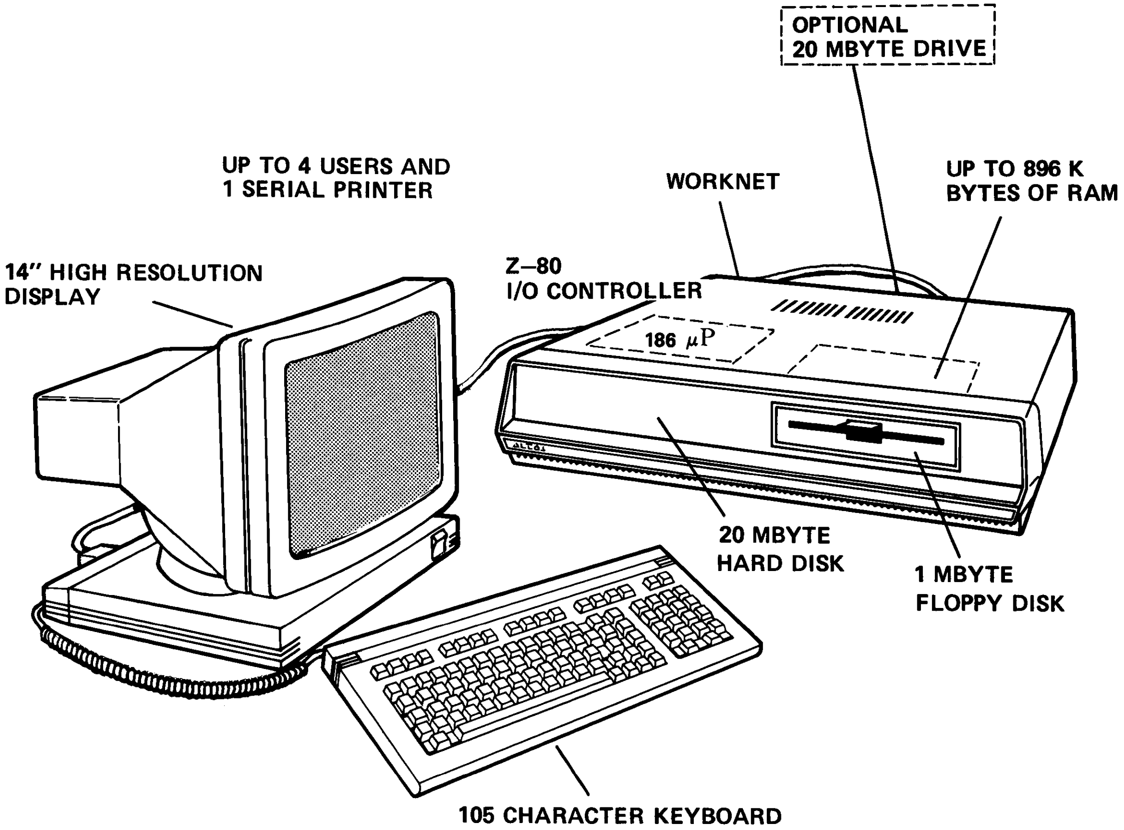


Figure 1-1 Altos Model 486 Computer System

SYSTEM SOFTWARE

The 486 uses several software programs to help run the system. These are the operating system, utilities, diagnostics, and the monitor programs. Applications and languages are not discussed in this manual.

OPERATING SYSTEMS

The 486 system supports two multiuser operating systems: XENIX and Concurrent CP/M. These are described below.

XENIX Operating System

This multiuser system complements a large number of applications such as office automation, database management, and communications. It is the Microsoft 16-bit microcomputer implementation of the UNIX operating system developed by Bell Laboratories.

The XENIX operating system supports program development and many software applications. XENIX features a hierarchial file system, compatibility between file, device, and interprocess I/O, asynchronous processing, a command language interpreter, a large number of subsystems and utilities, excellent file security, user-friendly menus, flexible memory use and compatiblity with a large number of high level languages.

Concurrent CP/M Operating System

Concurrent CP/M (Control Program for Microprocessors) is a multiuser operating system for 16-bit microcomputers. CP/M provides record and file locking with password protection for use in business settings where data base integrity is essential. CP/M also features extensive error handling and reporting, real-time capabilities, and data and time stamps and files.

SYSTEM MONITOR

The system monitor programs and modules reside permanently in the PROMs of the 486 and provide initial system checks and exercise initial coordination and control of the system. The Monitor includes an extensive set of power-up tests used to identify hardware faults in the system, and a debugger program to help find problems in users' programs.

DIAGNOSTICS

The diagnostics programs for the 486 are contained in a separate diskette supplied with the system. These SDX (system diagnostics executive) programs help you isolate problems in system operation. They are described in the Diagnostics manual along with selected system utility programs.

OPTIONS

Listed below are the options for the 486 System which are available through upgrade kits (U/K).

- An additional 384 Kbytes of RAM for a total of 896 Kbytes of main memory.

> NOTE
>
> The standard system is configured with 512 Kbytes of memory. 256 Kbytes are on the CPU board and the other 256 Kbytes are on a pair of memory boards (128 Kbytes each). The memory upgrade kit provides a 512 Kbyte board which is then exchanged with one of the existing 128 Kbyte memory boards.

- An optional 2Ø Mbyte add-on hard disk drive unit.

RELATED DOCUMENTS

The following Altos publications on the Model 486 Computer System may be referenced to supplement your overall understanding.

69Ø-15889-ØØ1	Setting Up Your 486 Workstation
69Ø-15927-ØØ1	Altos 486 SDX Diagnostic Manual
69Ø-15683-ØØ1	486 Maintenance Manual
69Ø-15998-ØØ1	486 Illustrated Parts List Manual

The Setting Up and Diagnostic Manuals are shipped with the system. The Maintenance Manual and Illustrated Parts List Manual can be purchased.

Specifications 2

SYSTEM SPECIFICATIONS

The following specifications describe the overall dimensions, power, environmental, and performance requirements for the 486 Computer System.

DESCRIPTION

Central Processor (CPU)
Floppy Disk Drive
Hard Disk Drive
WorkNet

Rear Panel Connectors and Controls

AC Power ON/OFF Switch
AC Power Input Receptacle
AC Fuse Holder
System Reset Button
Serial RS232 Ports: five 25 pin "D" connectors
WorkNet RS422 Port: 15 pin "D" connector

WorkNet is a local area network capable of supporting up to 30 other Altos computers.

Serial RS232 Ports support asynchronous terminals and printers.

POWER REQUIREMENTS

AC Power

Volts	115	230
Hertz	60	50
Amperes (Maximum)	3	2
Fuse Type	4A - 250V	2A - 250V
Fuse Size	3AG	3AG

Heat Dissipation

Watts	180
BTU per/Hour	614

CPU OVERALL DIMENSIONS

Height (inches)	3.5
Width (inches)	17
Length (inches)	15
Weight (Free standing weight) (pounds)	20

ENVIRONMENTAL REQUIREMENTS

To ensure best performance from the 486 System, avoid extremes in temperature or humidity. These conditions are likely causes of floppy and hard disk drive malfunctions. For further information about environmental constraints, refer to the ENVIRONMENTAL REQUIREMENTS for specific components. It is recommended that you position the 486 System so that disk drives are maintained in a horizontal position.

Ambient Temperature	59 to 90° F (15 to 32° C)
Relative Humidity	20 to 80 (percent)
Temperature (Maximum wet bulb Non-condensing Degree Farenheit)	78

MICROPROCESSOR

Intel - 80186	1
Address/Data Bus (bits)	16
Processor Speed (Megahertz)	8
DMA Channels	2
Programmable Interrupt Controller	1
Programmable 16-bit Timers	3

Zilog - Z80	1
Address Bus (bits)	16
Data Bus (bits)	8
Processor Speed (Megahertz)	4

MEMORY

Standard	512 Kbytes
Maximum	896 KBytes

OTHER SYSTEM SPECIFICATIONS

Electrostatic Discharge Dissipation
10 KV (with no hard errors) 300Pf

Agency Approvals
FCC, Class A, UL, CSA
Designed to meet VDE, IEC

POWER SUPPLY SPECIFICATIONS

o Output Voltages and Maximum Rated Loads

Output 1: +5.0V at 15.0 A
Output 2: +12 V at 4.0 A (surging to 7 A for 10 sec to start disk)
Output 3: -12 V at 0.5 A

Continuous Output Power is 153 watts

o Overall Regulation (from all causes)

Output 1: ± 3% max at 75% load ± 25% load change
Output 2: ± 5% max at 60% load ± 40% load change
Output 3: ± 5% max at 60% load ± 40% load change

o Noise and Ripple

Output 1: 100 mV p-p max
Output 2: 100 mV p-p max
Output 3: 200 mV p-p max

- Input Voltage (at all rated load conditions)

  U.S.: 95 VAC to 130 VAC single phase
  Europe: 190 VAC to 260 VAC single phase

- Efficiency at maximum continuous output power and nominal line input:

  70% min; forced air cooling available, approximately 30 cfm.

- Overvoltage protection threshold

  Output 1: 6.25 V ± 0.75 V

- Temperature Range

  0 to 50 degrees C ambient

- Mean Time Between Failures

  20,000 hours (continuous)

- Safety and Emissions Requirements

  UL and CSA mandatory
  VDE approvable

FLOPPY DISK DRIVE

POWER REQUIREMENTS

DC Power

Volts	+5.Ø
Amperes (typical)	Ø.6
Volts	+12.Ø
Amperes (typical Seeking)	Ø.7

Heat Dissipation

Watts	
Continuous Seek (typical)	11.4
Standby (typical)	9.Ø
Motor off (typical)	5.6

OVERALL DIMENSIONS

(Except for front panel)

Height (inches)	1.62
Width (inches)	5.75
Length (inches)	8.Ø

(Front Panel)

Height (inches)	1.65
Width (inches)	5.83
Total overall Weight (pounds)	2.9

ENVIRONMENTAL REQUIREMENTS

Operating Temperature 41 to 1Ø9.4^{o} F (5 to 43^{o} C)

Relative humidity (percent non-condensing)	2Ø to 8Ø
(Maximum wet bulb temperature)	85oF (29oC)

Non-operating Temperature -4 to 125^{o} F (-2Ø to 51.5^{o} C)

Relative humidity (percent)	5 to 95

PERFORMANCE SPECIFICATIONS

	Double Density
Memory capacity	
Unformatted	
Disk (Megabytes)	1
Track (Kilobytes)	6.25
Number of:	
Cylinders	8Ø
Head/Cylinder	2
Total Tracks (8Ø per side)	16Ø
Transfer Rate (Kilobits/sec)	25Ø
Rotational Latency (milliseconds average)	1ØØ
Access Time	
Track to Track (ms)	3
Average (ms)	94
Motor Starting Time (ms)	25Ø
Settling Time (ms)	15
Head loading Time (ms)	5Ø
Encoding Method	MFM
Index	1
Rotational Speed (rpm)	3ØØ
Disk Diameter (inches)	5.25Ø
Track Density (Tracks per inches)	96

HARD DISK DRIVE

POWER REQUIREMENTS

DC Power

Volts	+12
Steady State: (percent)	±5
Amperes (typical)	1
(Maximum)	1.2
Ripple (Maximum percent allowed with equivalent resistive load)	1
Start Surge: (Maximum percent)	±1Ø
Amperes (Maximum) (1Ø Seconds typical)	2.5

Volts (±5 percent)	+5
Amperes (typical)	Ø.4
(Maximum)	Ø.5
Ripple (Maximum percent allowed with equivalent resistive load)	2

AC Power

None required

Heat Dissipation

Watts (typical)	14
(Maximum)	17

OVERALL DIMENSIONS

Height (inches)	1.63
Width (inches)	5.88
Length (inches)	8.Ø
Weight (pounds)	4.5

ENVIRONMENTAL REQUIREMENTS

Operating Temperature	4Ø to 122 °F (4 to 5Ø °C)
Relative humidity (percent non-condensing)	8 to 8Ø
(Maximum wet bulb temperature)	78oF (26oC)
Non-operating Temperature	-4Ø to 135 °F (-4Ø to 57 °C)
Relative humidity (percent)	8 to 8Ø

PERFORMANCE SPECIFICATIONS

Memory Capacity:	
Unformatted	
Drive (Megabytes)	25.5
Surface (Megabytes)	6.38
Track (Kilobytes)	10.416
Cylinder (Kilobytes)	41.664
Formatted	
Drive (Megabytes)	20
Surface (Megabytes)	10
Track (Kilobytes)	8
Sector (bytes)	512
Sector/track	16
Number of:	
Cylinders	612
Head/Cylinder	4
Data Tracks	2448
Surfaces	4
Disks	2
Transfer rate (Megabits/sec.)	5.0
Access Time	
Average Latency (ms)	8.33
Seek Time (including settling time)	
Single Track (ms)	15
Average (ms)	85
Maximum (ms)	190
Rotational Speed (rpm)	3600
Recording Density	
Areal (TPI x BPI) (Mbits/sq.in.)	5.90
Linear (MFM)	10000
Radial (Tracks per inch)	588
Disk Diameter (inches)	5.250

System Architecture 3

PAGE SUBJECT

ILLUSTRATIONS

INTRODUCTION

This chapter describes the architecture of the 486 Computer System through a discussion of the bus structure and memory arbitration unit.

BUS STRUCTURE

The CPU is made up of various physical data and address buses that determine the general architecture of the system. Refer to Figure 3-1.

The major buses are the XLA Bus, MP (Mapped Address) Bus, AD (Address) Bus, BD (Buffered Data) Bus, ZD0-7 Bus, LA (Logical Address) Bus, MD Bus,and Z80 Address Bus.

The AD bus is a multiplexed 16-bit address/data bus from the 80186 microprocessor and is the main bus of the system. The AD bus "spawns" all the other buses in the system.

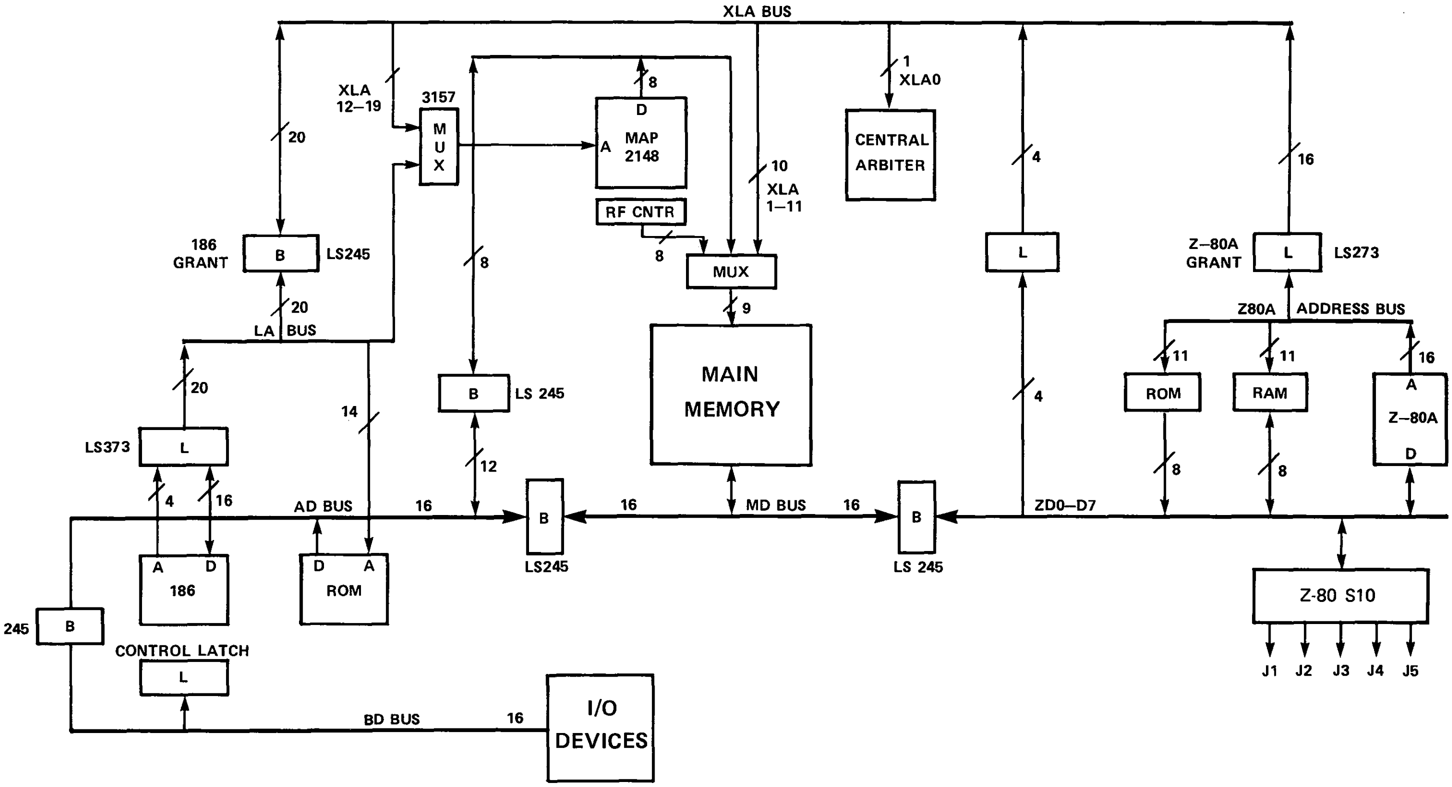


Figure 3-1 Bus Architecture

The 80186 requires a 20-bit address to access all the possible locations in main memory; a 16-bit address can only access 65,536 bytes of memory. The processor creates this 20-bit address by appending four low order zeros (A0 through A3) to the 16-bit segment address (which specifies a 64K segment of memory) and combining this with a 16-bit offset address (which specifies a location within the segment). This 20-bit bus is called the logical address (LA) bus.

The LA Bus is created from the AD Bus when the addresses are latched at ALE time and held for the entire bus cycle. The boot and diagnostic PROM are addressed via the LA bus. When the LA bus is buffered, the XLA bus is developed.

The XLA bus provides virtual memory capability through relocation registers residing between the bus and the main memory. This means that if the data ,during a DMA transfer, is not in physical memory location it will be transferred into memory from the disk by the relocation registers addressing space process.

The CPU board design is divided into two major hardware sections, the 80186 section and the Z80 section. The XLA Bus connects these two major sections together so that either processor can access the main memory.

The BD Bus is within the 80186 section, it is the (Buffered) Bus which is created from the multiplexed AD Bus. The BD Bus is an I/O bus. All devices which are in the I/O space of the 80186 processor send and receive data on this bus.

The Mapped Address Bus is created through the Memory Management Unit (MMU). One function of the MMU is to create page addressing to memory. Another function of the Memory Management Unit is to protect the operating system from normal user access.

Within the Z80 section is the ZD0-7 (Data) Bus which is used by the Z80 Microprocessor to transfer eight data bits from the SIO ports to main memory and the 186 through the MD (Memory Data bus).

Located on the ZD0-7 Bus are the SIO controllers for the five RS232 channels, two timer devices and a small amount of RAM and ROM. The Z80 address bus is used to access locations in the local 2K of RAM, for transferring instruction codes from the local control ROM and to interface the XLA bus for Z80 I/O processing.

MEMORY ARBITRATION

These are all the buses for the 486 system. You will note that both the Z8Ø and 8Ø186 processors require main memory access. To allow both the Z80 and 8Ø186 buses to share main memory, a central arbitration unit is used. This arbitration unit controls access to main memory.

The central arbitration unit is a three way arbiter - arbitrating requests between the 186 and Z8Ø and a refresh cycle. When a memory cycle has been allowed access to the bus, that request will be the only one serviced on the bus. A requesting memory cycle cannot interrupt a currently executing cycle.

A refresh cycle must take place at lease every 15.6 us to recharge the dynamic RAM chips. If all three requests arrive at the central arbiter at the same time, the refresh cycle has highest priority and the 8Ø186 cycle has second priority. Lower priority requests are cued and later executed.

General System Functional Description 4

PAGE SUBJECT

ILLUSTRATION

INTRODUCTION

This chapter provides a general description of the 486 system. Figure 4-1 shows a block diagram of the main subsystems which make up the 486. These include the CPU board and the hard disk controller board which are briefly described below.

CPU BOARD

The CPU board is a multi-functional subsystem that contains the main processor, the I/O processor, the floppy disk drive controller, 256 Kbytes of main memory, the memory management unit, the memory arbitration logic and the I/O ports for all peripheral interfacing.

80186 MICROPROCESSOR

The 486 Computer System is designed around the Intel 80186 microprocessor. It performs most of the central processing unit functions, with the capabilities for Direct Memory Access (DMA), programmable Timers, and Interrupts. It is an integrated circuit housed in a special 68 pin package.

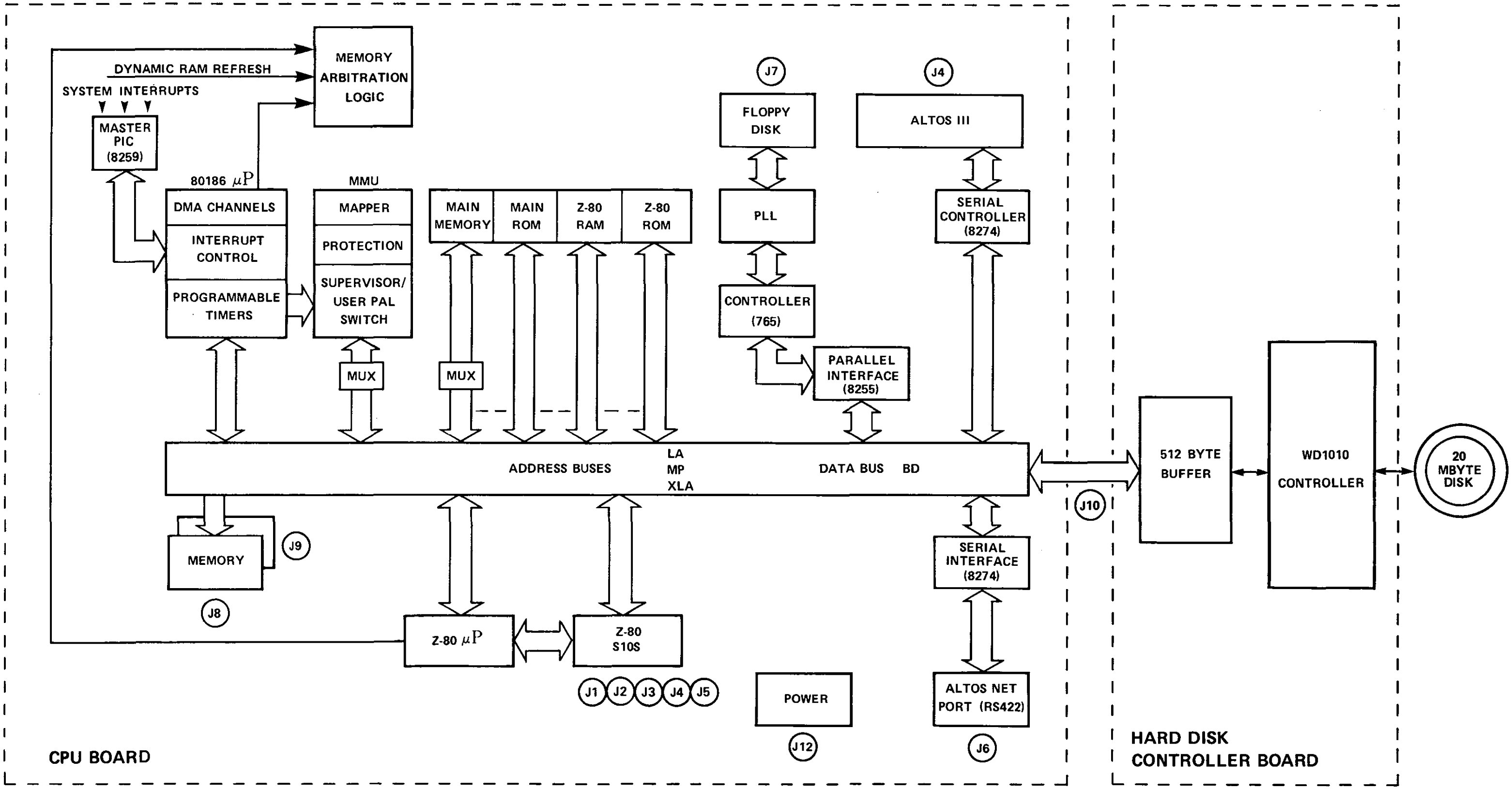


Figure 4-1 System Block Diagram

The basic features of the 8Ø186 Microprocessor are as follows:

- 16-bit address/data bus
- 8 Mhz clock
- Two independent DMA channels
- Programmable Interrupt Controller
- Programmable 16-bit timers

In addition, it can address up to 1 megabyte of memory in banks of 64 Kbytes and over 65,ØØØ I/O ports. The CPU performs 8 and 16-bit arithmetic functions in binary or decimal.

The 8Ø186's internal programmable interrupt control (PIC) acts as a slave to the master external PIC. It is through this master/slave setup that the microprocessor services system and internal interrupts.

The two DMA (Direct Memory Access) channels handle requests from the disk drives and WorkNet port. One channel is for the hard and floppy disks DMA requests; the other is for networking.

MEMORY DEVICES

The system provides four separate memory components to handle the various memory requirements of the system. These are the "main" memory, the system PROM, Z-8ØA RAM, and Z-8ØA ROM.

The standard main memory contains 512 Kbytes of dynamic RAM which is upgradeable to 896 Kbytes. 256 Kbytes are on the CPU board; the remainder are on two memory cards. The PROM contains the power-up tests, the debugger and various boot device drivers.

The Z-8ØA processing circuitry has 4K of ROM and 2K of RAM for I/O. The ROM contains the control program and the static RAM performs temporary storage and data buffering.

MEMORY MANAGEMENT UNIT (MMU)

The MMU creates page memory addressing and contains a mapper and protection circuitry. The MMU circuitry is also responsible for switching the system between user and the protected supervisor mode. In user mode, the system executes the shell or application programs, such as a word processing or an accounting program, for example. In the supervisor mode, monitor programs can be run. In addition, I/O operations can only take place in the supervisor mode, and for operating systems like XENIX, the kernel must run in this state.

FLOPPY DISK DRIVE

The floppy disk circuitry is controlled through the 765 chip and interrupts the microprocessor via the 8255 parallel interface port. Control and read/write signals to and from the disk drive are handled through connector J7 on the CPU. The PLL (Phase Lock Loop) aids the 765 in recovering MFM (Modified Frequency Modulation) data.

WORKNET

WorkNet enables the 486 to be connected to a network of other Altos computers using synchronous data link control (SDLC). WorkNet conforms to an RS422 electrical standard.

To implement WorkNet, the 486 systems dedicates one of its two DMA channels for this function. A circuit then generates 8ØØ KHz for the network baud rate.

Z-8ØA I/O PROCESSOR

The Z-8ØA I/O processor runs a control program that resides in a PROM that is local to the Z-8ØA. It also uses a small amount of static RAM for temporary storage and buffering. The I/O processor communicates with the central processor via system memory locations and interrupts. It supports five RS232 channels.

The Z-8ØA 64K of memory space is divided into two halves. The upper 32K is mapped into the system memory and the lower 32K is mapped into the local PROM and static RAM. The Z-8ØA can access the full 512 Kbytes of memory but must do so in blocks of 32 Kbytes each.

HARD DISK CONTROLLER BOARD

The other major subsystem of the 486 is the Hard Disk Controller unit. The main component on the board is the WD1Ø1Ø Winchester chip. The WD1Ø1Ø Winchester drive controller chip controls the Hard Disk Drive. Read/write instructions from the microprocessor enter the Hard Disk Controller board through the J1Ø connector. The command is then sent to the WD1Ø1Ø Controller which controls and monitors a 512 byte buffer used to hold read or write data. One hard disk sector 512 bytes of information. Data is transferred to and from the 2Ø Mbyte disk drive.

Subsystem Functions 5

ILLUSTRATIONS

TABLES

INTRODUCTION

This chapter discusses the subsystem functions of the 486 system that are of interest to a system programmer. First an overview of the function is provided, then any programmable characteristics are discussed followed by addressing and register information for that function. As identified in the previous chapter, the main subsystems of the 486 are the CPU board and the Hard Disk Controller board.

CENTRAL PROCESSING UNIT (CPU) HARDWARE OVERVIEW

The Central Processing Unit (CPU) printed circuit board performs most of the major functions for the 486 system. Refer to Figure 5-1.

The CPU printed circuit board is designed around two major sections of hardware, the 8Ø186 section and the Z-8ØA section. Within the 186 section are the 186 microprocessor, the floppy disk controller, the boot ROM, the main memory, the memory management unit, the parallel interface controller (8255), and programmable interrupt controller (8259) IC chips.

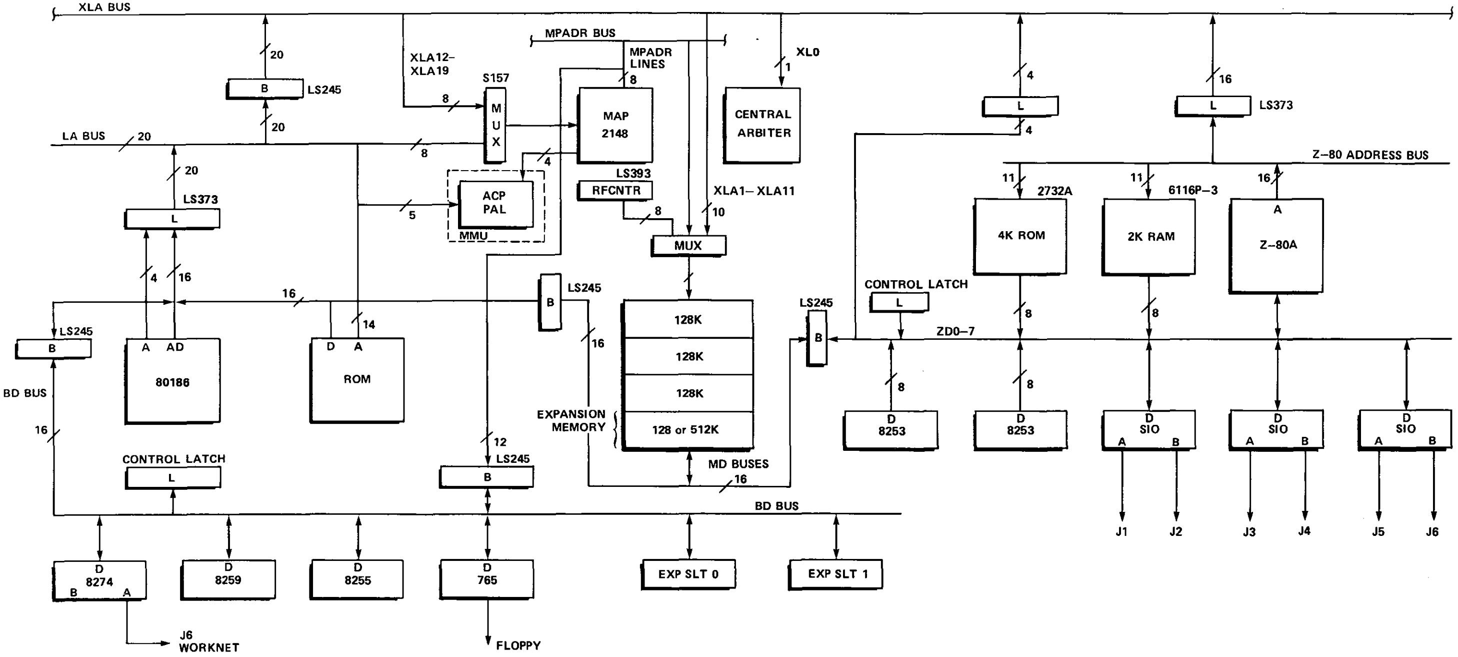


Figure 5-1 CPU Board Block Diagram

The Z-80A section contains an intelligent serial I/O controller circuitry that provides high speed data transfer to and from the user terminals. Within the Z-80A section is the Z-80A microprocessor, three Z-80A SIO controllers for five channels, two timer devices and a small amount of RAM and ROM.

80186 MICROPROCESSOR ARCHITECTURE

This section describes the architecture of the 80186 microprocessor within the 486 system, specifically the addressing modes, the DMA function and programmable timers.

Addressing Modes

There are three addressing schemes done through the 80186 in the 486 system: 1) operand addressing, 2) external port or register addressing, and 3) main memory addressing. Memory addressing is discussed later in this chapter.

Operand Addressing

The 80186 Microprocessor has eight operand addressing modes. Two addressing modes are provided for instructions that operate on internal register or immediate operands and six addressing modes are provided to specify the location of an operand in a memory segment.

A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicity chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset is calculated by summing any combination of the following three address elements.

The displacement (an 8- or 16-bit immediate value contained in the instruction)

The base (contents of either the BX or BP base registers)

The index (contents of either the SI or DI index registers)

For more information on operand addressing modes, refer to the data book.

Port and Register Addressing

The port or register addresses are composed of a programmable base address and an offset. A list of these offset values are provided in Tables 5-1 through 5-19.

To obtain the physical address for a port or an external register, the offset address is added to the base address. The base address refers to a particular address range in the I/O space. For example, if the base address were programmed to start at 1ØØØ and if the floppy status register were being addressed, the offset address, 28Ø (hex), would be added to the base address, and the result would be 128Ø (hex).

NOTE

The default base address in this system is 1ØØØ.

The starting address is defined by the PACS register of the 8Ø186 as programmed by the user. For more information on programming the PACS register, refer to the data book.

Pheripheral chip select lines, PCSØ* through PCS3*, on the 8Ø186 chip are used to select this starting address range block in the I/O space, PCSØ* enables block 1ØØØ, PCS1* enables block 2ØØØ and so forth. See the internal PCS register offsets in Table 5-1. An MPCS register determines the mode of operation of the peripheral chip selects. Refer to the data book for more information on this function.

NOTE

Do not confuse a register or port address offset with a memory offset. A memory offset is used to construct the 2Ø bit address. Refer to discussions on Bus Structure in Chapter 3 and Memory Addressing later in this chapter.

Direct Memory Access (DMA) Operation

Direct Memory Access (DMA) can be instructed, by the Hard Disk Drive Controller printed circuit board, the Floppy Disk Drive Controller chip or the WorkNet, to access the full contents of main memory. Data can be transferred directly from memory to I/O, memory to memory, or I/O to I/O, without the intervention of the microprocessor. This leaves the mircroprocessor free for other tasks.

The 8Ø186 microprocessor provides two independent high-speed DMA channels. Each channel has six registers in the control block that define that channel's specific operation, see Table 5-1. The control registers consist of a 2Ø-bit source pointer, a 2Ø-bit destination pointer, a 16-bit transfer counter and a 16-bit control word. Data may be transferred in eight bit or sixteen bit transfers.

Data transfers may be either source or destination synchronized, that is, either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized, that is, transfers take place continually until the correct number of transfers has occurred.

To perform a DMA channel transfer, the channel program provides information that describes the operation by loading values into channel registers. DMA cycles can run in either Supervisor or User state and must be permitted to run without triggering access protection mechanism.

Table 5-1 Chip Select Register Offset

ADDRESS (HEX)	DESCRIPTION
AØH	UPPER MEM CS
A2H	LOWER MEM CS
A4H	PACS REGISTER
A6H	MID RANGE MEM CS
A8H	MPCS REGISTER

Table 5-2 DMA Register Offsets

ADDRESS (HEX)	DESCRIPTION	CHANNEL
FFCØH	SOURCE POINTER	CH Ø
FFC2H	SOURCE POINTER	CH Ø
FFC4H	DESTINATION POINTER	CH Ø
FFC6H	DESTINATION POINTER	CH Ø
FFC8H	TRANSFER COUNT	CH Ø
FFCAH	CONTROL WORD	CH Ø
FFDØH	SOURCE POINTER	CH 1
FFD2H	SOURCE POINTER	CH 1
FFD4H	DESTINATION POINTER	CH 1
FFD6H	DESTINATION POINTER	CH 1
FFD8H	TRANSFER COUNT	CH 1
FFDAH	CONTROL WORD	CH 1

Programmable Timer Operation

There are three internal 16-bit programmable timers in the 8Ø186 microprocessor, although only two are pinned out. The other timer is for real time applications and generating delayed signals. One of the two pinned out timers is used for generating a baud clock signal at 1.2288 Mhz. The other pinned out timer is used for the system clock.

The timers are controlled by eleven 16-bit registers in the internal peripheral control block, see Table 5-3. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of the register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself.

The timers have several programmable options:

- All three timers can be set tc halt or continue on a terminal count.
- Timers Ø and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

Table 5-3 Timer Control Register Offsets

ADDRESS (HEX)	FUNCTION	TIMER NO.
FF5ØH	COUNT REGISTER	TIMER Ø
FF52H	MAX COUNT A	TIMER Ø
FF54H	MAX COUNT B	TIMER Ø
FF56H	MODE/CONTROL	TIMER Ø
FF58H	COUNT REGISTER	TIMER 1
FF5AH	MAX COUNT A	TIMER 1
FF5CH	MAX COUNT B	TIMER 1
FF5EH	MODE/CONTROL	TIMER Ø
FF6ØH	COUNT REGISTER	TIMER 2
FF62H	MAX COUNT A	TIMER 2
FF66H	MODE/CONTROL	TIMER 2

Interrupt Control Operation

Internal and external interrupts provide a means of stopping the normal processing flow in order to handle unexpected or difficult situations.

The system operates in the RMX Mode and requires an external master Programmable Interrupt Controller (PIC), the Intel 8259, to handle and service interrupts. As stated earlier, the 8Ø186 contains a built-in PIC which acts as a slave to the external PIC. The internal 8Ø186 resources will be monitored through the internal interrupt controller. See Figure 5-2 for a block diagram of the master/slave PIC setup.

When an internal interrupt takes place, the internal slave Interrupt Controller, residing in the 8Ø186 Microprocessor, resolves priorities and generates an internal interrupt request to the external master PIC. Internal interrupts are from Timers and DMA channels and can be disabled by their own control registers or by mask bits within the interrupt controller. There are 15 registers in the internal peripheral control block, see Table 5-4.

The external master PIC resolves priorities among all the external interrupt sources and issues an interrupt to the 8Ø186 Microprocessor. The 8Ø186 Microprocessor then causes an interrupt acknowledge cycle to take place. After the interrupt condition is serviced the system continues with the user program.

Five pins on the 8Ø186 are dedicated for the interrupt function:

NMI - Non-Maskable Interrupt is typically used for parity errors or a reset.

INTØ - Input is used as the 186 CPU interrupt.

INT1 - Input is used as a slave select input.

INT2 - Output is used as an acknowledge output.

INT3 - Output is to the PIC.

There are eight different levels of system interrupts, interpreted by the PIC, (in order of importance to the system.) The lower the Bit Designation Number is, the more important the interrupt. Therefore, all Memory Mangagement Interrupts are handled before all I/O Port Interrupts. Table 5-5 lists the interrupts in numeric order (and order of importance.) Figure 5-3 illustrates the interrupt activity which are discussed briefly in the following paragraphs.

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.

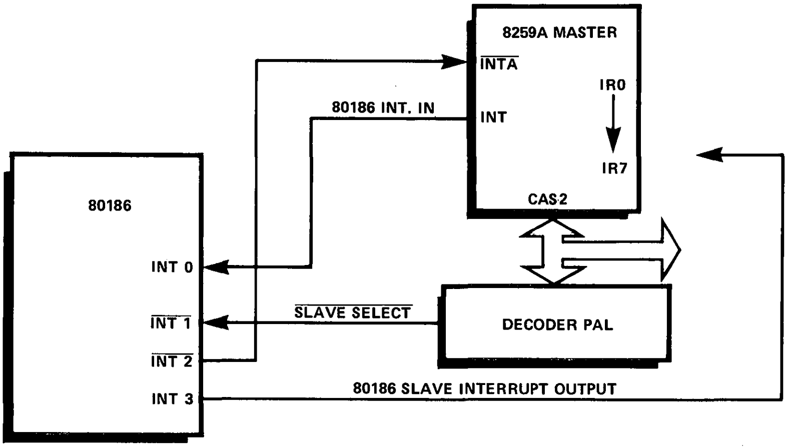


Figure 5-2. Master/Slave PIC Configuration

Table 5-4 Interrupt Controller Register Offsets

ADDRESS (HEX)	FUNCTION
FF22H	EOI
FF24H	POLL
FF26H	POLL STATUS
FF28H	MASK
FF2AH	PRIORITY MASK
FF2CH	IN-SERVICE
FF2EH	INTERRUPT REQUEST
FF3ØH	INTERRUPT CONTROLLER STATUS
FF32H	TIMER CONTROL
FF34H	DMA Ø CONTROL
FF36H	DMA 1 CONTROL
FF38H	INTO CONTROL
FF3AH	INTl CONTROL
FF3CH	INT2 CONTROL
FF3EH	INT3 CONTROL

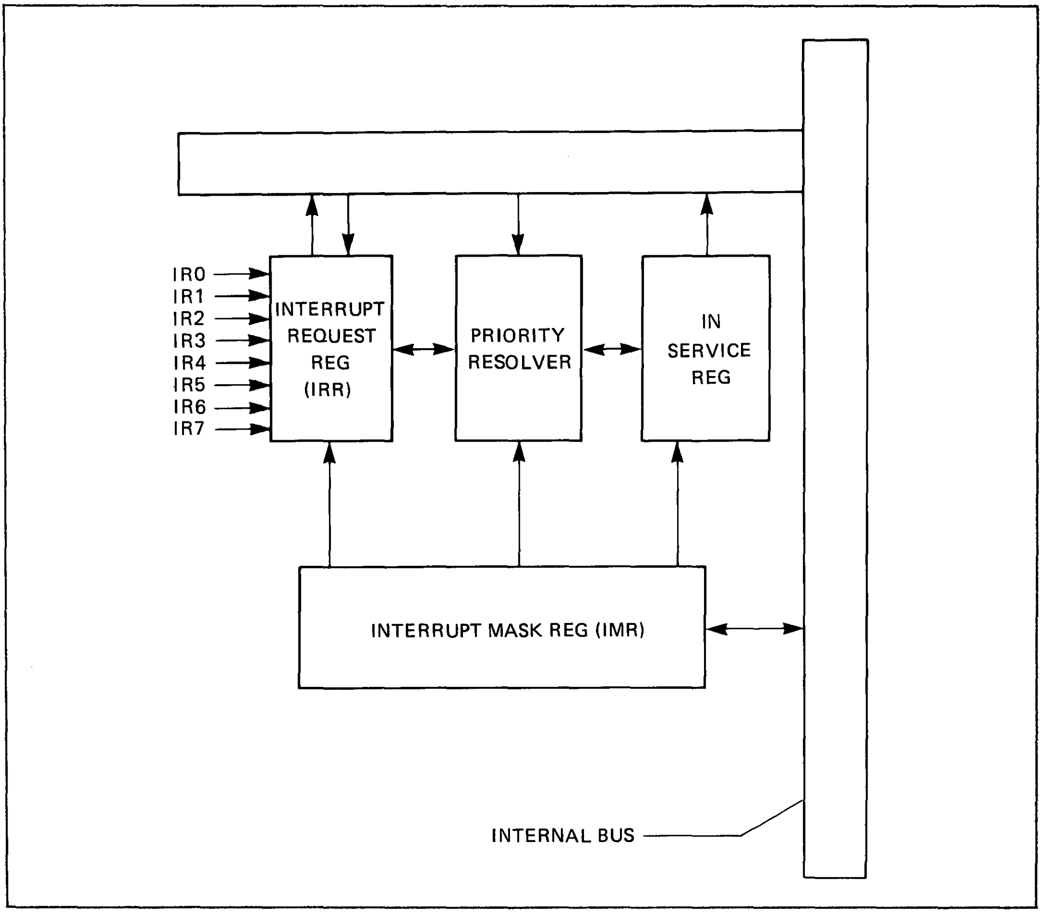


Figure 5-3 Interrupt Request

Priority Resolver

This logic block determines the user-defined priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA* pulse.

The Interrupt Mask Register (IMR) allows the user to prevent (mask) an interrupt. During a particular application, if the parity failure interrupt occurs, it can be ignored. The mask allows the user to state at a given time which of those interrupts, listed in Table 5-5 is going to be accepted.

Table 5-5 System Maskable Interrupts

Mnemonic	**Description**	**Bit** Designation
MMINT	MEMORY MANAGEMENT INTERRUPT	IRØ
	NOT USED	IR1
AINT	ALTOSNET INTERRUPT	IR2
INTRQ	INTERRUPT REQUEST	IR3
INTERNAL INT	CPU INTERNAL INTERRUPT	IR4
	NOT USED	IR5
IOPINT	ZDØ-7 BUS INTERRUPT	IR6
TMRINT	TIMING INTERRUPT	IR7

Master PIC Operation

When the Altos 486 Computer System is powered up, the CPU loads the initialization command word and operation command word into the PIC's internal registers, see Table 5-6. The initialization command word provides the starting address of the service routines, while the operation command word provides the mode of operation. This mode is described below:

Fully Nested Mode - After the initialization sequence IRØ has the highest priority and IR7 the lowest.

This mode permits the PIC to operate as described above. The commands initially entered in the PIC internal registers are instructions which include what mode to enter and use. The initial command instructions can be permanently loaded in the PIC internal registers so that it will always operate in the same mode, or the loading instructions can be modified as conditions change by reprogramming the device.

Every interrupt is assigned a type code that identifies it. A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Table 5-7 shows the 80186 Microprocessor predefined types and default priorities.

The Microprocessor then multiplies the 8-bit vector by four which becomes the address in a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are progammed by writing to the interrupt vector register at offset 20H.

Table 5-6 Master PIC Register Offsets

Address (HEX)	Function	Type
2C0H	READ OR WRITE	CONTROL
2C2H	READ OR WRITE	CONTROL

Table 5-7 80186 Microprocessor Predefined Types and Default Priority

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV,1DIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INTO Detected Overflow Exception	4	*1	INTO
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 interrupt	8	2A****	
Timer 1 interrupt	18	2B****	
Timer 2 interrupt	19	2C****	
Reserved	9	3	
DMA 0 interrupt	10	4	
DMA 1 interrupt	11	5	
INTO interrupt	12	6	
INT1 interrupt	13	7	
INT2 interrupt	14	8	
INT3 interrupt	15	9	

NOTES:

*1. These are generated as the result of an instruction execution.

**2. Generated by the Single-Step flag bit TF in the status word. Once this bit is set, an internal single step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

****3. All three timers constitute one source of request to the interrupt controller. Among the timer interrupts, priority 2A is higher priority than 2B and 2B is higher than 2C. Each Timer interrupt has a separate vector type number.

4. Default priorities for the interrupt sources are used only if you do not program each source into a unique priority level.

***5. The escape opcode (part of the instruction word that designates the function performed by a given instruction) will cause a trap only if the proper bit is set in the peripheral control block relocation register.

MEMORY ADDRESSING

Memory is addressed by a two component address, consisting of a 16-bit base segment and a 16-bit offset value, derived from a combination of pointer registers, instruction pointer, and immediate values.

The 16-bit base values are contained in the one of four internal segment registers. The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 2Ø-bit physical address. This allows the 8Ø186 to access up to 1 Mbyte of physical main memory, if it were available for the system. With expansion memory, the 486 system can hold up to 896 Kbytes of main memory. 512 Kbytes is contained in the standard configuration. Table 5-8 shows a memory map of the system.

Table 5-8 System Memory Map

Memory Locations	Contents
ØØØØØ - ØØ3FF	System's interrupt vector table
ØØ4ØØ - ØØ7FF	Monitor's working area
ØØFFE - Down	Monitor's stack area
Ø1ØØØ - Up	User's memory space
FCØØØ - FFFBF	Monitor program
FFFCØ - FFFDF	Monitor's "copy right" statement
FFFEØ - FFFEB	Monitor's bootstrap program
FFFEC - FFFED	PROM release and version number
FFFEE - FFFEF	PROM checksum
FFFFØ - FFFFF	Power-up jump vector

MEMORY MANAGEMENT UNIT

The MMU (memory management unit) performs memory mapping and management, and provides access protection and violation detection for the mapped locations.

Memory Mapping

The MMU changes the upper eight bits of the XLA bus, generates four attribute protection bits to the BD bus, then combines these bits with the remaining lower bits from the XLA bus to produce the mapped address bus to main memory.

The translation of the XLA bus to the MP bus is controlled by a "mapper" in the MMU circuitry. The mapper is a collection of 256 relocation registers, see Table 5-9. Each register maps a 4K byte page of main memory. This gives a possible total of 1 Mbyte of memory.

Memory mapping is important in a multi-user environment because it keeps different users' programs from interfering with the operating system and with each other. Allocation of memory for each user is performed by the operating system.

Protection Function

Memory protecton allows several different programs ("tasks" or "processes") to use the same physical memory without interfering with the operating system and other user programs. Before running a particular program, the CPU sets up the MMU to map the logical address space of the CPU into the physical memory reserved for that program.

Programs can not interfere with the operating system because each could access only its own portion of the physical memory as determined by the map. This is done through the protection attribute bits discussed later in this section.

Two types of maps reside in the MMU, one for the operating system and one for user programs. In user mode, a user map is selected and the processor may not execute priviledged I/O. These are instructions whose execution could compromise the integrity of the operating system.

The set of priviledged instructions only includes I/O instructions. The operating system always changes the CPU to be in user mode before transferring control to a user program. Therefore, if the MMU maps are accessed by I/O instructions, a user program cannot access the maps. Of course, all other I/O must also be done by the operating system, since only it can execute I/O instructions.

Table 5-9 Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
00	00000-00FFF	000
01	01000-01FFF	002
02	02000-02FFF	004
03	03000-03FFF	006
04	04000-04FFF	008
05	05000-05FFF	00A
06	06000-06FFF	00C
07	07000-07FFF	00E
08	08000-08FFF	010
09	09000-09FFF	012
0A	0A000-0AFFF	014
0B	0B000-0BFFF	016
0C	0C000-0CFFF	018
0D	0D000-0DFFF	01A
0E	0E000-0EFFF	01C
0F	0F000-0FFFF	01E
10	10000-10FFF	020
11	11000-11FFF	022
12	12000-12FFF	024
13	13000-13FFF	026
14	14000-14FFF	028
15	15000-15FFF	02A
16	16000-16FFF	02C
17	17000-17FFF	02E

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
18	18ØØØ-18FFF	Ø3Ø
19	19ØØØ-19FFF	Ø32
1A	1AØØØ-1AFFF	Ø34
1B	1BØØØ-1BFFF	Ø36
1C	1CØØØ-1CFFF	Ø38
1D	1DØØØ-1DFFF	Ø3A
1E	1EØØØ-1EFFF	Ø3B
1F	1FØØØ-1FFFF	Ø3E
2Ø	2ØØØØ-2ØFFF	Ø4Ø
21	21ØØØ-21FFF	Ø42
22	22ØØØ-22FFF	Ø44
23	23ØØØ-23FFF	Ø46
24	24ØØØ-24FFF	Ø48
25	25ØØØ-25FFF	Ø4A
26	26ØØØ-26FFF	Ø4C
27	27ØØØ-27FFF	Ø4E
28	28ØØØ-28FFF	Ø5Ø
29	29ØØØ-29FFF	Ø52
2A	2AØØØ-2AFFF	Ø54
2B	2BØØØ-2BFFF	Ø56
2C	2CØØØ-2CFFF	Ø58
2D	2DØØØ-2DFFF	Ø5A
2E	2EØØØ-2EFFF	Ø5C
2F	2FØØØ-2FFFF	Ø5E
3Ø	3ØØØØ-3ØFFF	Ø6Ø
31	31ØØØ-31FFF	Ø62
32	32ØØØ-32FFF	Ø64
33	33ØØØ-33FFF	Ø66
34	34ØØØ-34FFF	Ø68
35	35ØØØ-35FFF	Ø6A
36	36ØØØ-36FFF	Ø6C
37	37ØØØ-37FFF	Ø6E
38	38ØØØ-38FFF	Ø7Ø
39	39ØØØ-39FFF	Ø72
3A	3AØØØ-3AFFF	Ø74
3B	3BØØØ-3BFFF	Ø76

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
3C	3C000-3CFFF	078
3D	3D000-3DFFF	07A
3E	3E000-3EFFF	07C
3F	3F000-3FFFF	07E
40	40000-40FFF	080
41	41000-41FFF	082
42	42000-42FFF	084
43	43000-43FFF	086
44	44000-44FFF	088
45	45000-45FFF	08A
46	46000-46FFF	08C
47	47000-47FFF	08E
48	48000-48FFF	090
49	49000-49FFF	092
4A	4A000-4AFFF	094
4B	4B000-4BFFF	096
4C	4C000-4CFFF	098
4D	4D000-4DFFF	09A
4E	4E000-4EFFF	09C
4F	4F000-4FFFF	09E
50	50000-50FFF	0A0
51	51000-51FFF	0A2
52	52000-52FFF	0A4
53	53000-53FFF	0A6
54	54000-54FFF	0A8
55	55000-55FFF	0AA
56	56000-56FFF	0AC
57	57000-57FFF	0AE
58	58000-58FFF	0B0
59	59000-59FFF	0B2
5A	5A000-5AFFF	0B4
5B	5B000-5BFFF	0B6
5C	5C000-5CFFF	0B8
5D	5D000-5DFFF	0BA
5E	5E000-5EFFF	0BC
5F	5F000-5FFFF	0BE

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
60	60000-60FFF	0C0
61	61000-61FFF	0C2
62	62000-62FFF	0C4
63	63000-63FFF	0C6
64	64000-64FFF	0C8
65	65000-65FFF	0CA
66	66000-66FFF	0CC
67	67000-67FFF	0CE
68	68000-68FFF	0D0
69	69000-69FFF	0D2
6A	6A000-6AFFF	0D4
6B	6B000-6BFFF	0D6
6C	6C000-6CFFF	0D8
6D	6D000-6DFFF	0DA
6E	6E000-6EFFF	0DC
6F	6F000-6FFFF	0DE
70	70000-70FFF	0E0
71	71000-71FFF	0E2
72	72000-72FFF	0E4
73	73000-73FFF	0E6
74	74000-74FFF	0E8
75	75000-75FFF	0EA
76	76000-76FFF	0EC
77	77000-77FFF	0EE
78	78000-78FFF	0F0
79	79000-79FFF	0F2
7A	7A000-7AFFF	0F4
7B	7B000-7BFFF	0F6
7C	7C000-7CFFF	0F8
7D	7D000-7DFFF	0FA
7E	7E000-7EFFF	0FC
7F	7F000-7FFFF	0FE

Table 5-9 Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
8Ø	8ØØØØ-8ØFFF	1ØØ
81	81ØØØ-81FFF	1Ø2
82	82ØØØ-82FFF	1Ø4
83	83ØØØ-83FFF	1Ø6
84	84ØØØ-84FFF	1Ø8
85	85ØØØ-85FFF	1ØA
86	86ØØØ-86FFF	1ØC
87	87ØØØ-87FFF	1ØE
88	88ØØØ-88FFF	11Ø
89	89ØØØ-89FFF	112
8A	8AØØØ-8AFFF	114
8B	8BØØØ-8BFFF	116
8C	8CØØØ-8CFFF	118
8D	8DØØØ-8DFFF	11A
8E	8EØØØ-8EFFF	11C
8F	8FØØØ-8FFFF	11E
9Ø	9ØØØØ-9ØFFF	12Ø
91	91ØØØ-91FFF	122
92	92ØØØ-92FFF	124
93	93ØØØ-93FFF	126
94	94ØØØ-94FFF	128
95	95ØØØ-95FFF	12A
96	96ØØØ-96FFF	12C
97	97ØØØ-97FFF	12E
98	98ØØØ-98FFF	13Ø
99	99ØØØ-99FFF	132
9A	9AØØØ-9AFFF	134
9B	9BØØØ-9BFFF	136
9C	9CØØØ-9CFFF	138
9D	9DØØØ-9DFFF	13A
9E	9EØØØ-9EFFF	13C
9F	9FØØØ-9FFFF	13E

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
A0	A0000-A0FFF	140
A1	A1000-A1FFF	142
A2	A2000-A2FFF	144
A3	A3000-A3FFF	146
A4	A4000-A4FFF	148
A5	A5000-A5FFF	14A
A6	A6000-A6FFF	14C
A7	A7000-A7FFF	14E
A8	A8000-A8FFF	150
A9	A9000-A9FFF	152
AA	AA000-AAFFF	154
AB	AB000-ABFFF	156
AC	AC000-ACFFF	158
AD	AD000-ADFFF	15A
AE	AE000-AEFFF	15C
AF	AF000-AFFFF	15E
B0	B0000-B0FFF	160
B1	B1000-B1FFF	162
B2	B2000-B2FFF	164
B3	B3000-B3FFF	166
B4	B4000-B4FFF	168
B5	B5000-B5FFF	16A
B6	B6000-B6FFF	16C
B7	B7000-B7FFF	16E
B8	B8000-B8FFF	170
B9	B9000-B9FFF	172
BA	BA000-BAFFF	174
BB	BB000-BBFFF	176
BC	BC000-BCFFF	178
BD	BD000-BDFFF	17A
BE	BE000-BEFFF	17C
BF	BF000-BFFFF	17E

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
C0	C0000-C0FFF	180
C1	C1000-C1FFF	182
C2	C2000-C2FFF	184
C3	C3000-C3FFF	186
C4	C4000-C4FFF	188
C5	C5000-C5FFF	18A
C6	C6000-C6FFF	18C
C7	C7000-C7FFF	18E
C8	C8000-C8FFF	190
C9	C9000-C9FFF	192
CA	CA000-CAFFF	194
CB	CB000-CBFFF	196
CC	CC000-CCFFF	198
CD	CD000-CDFFF	19A
CE	CE000-CEFFF	19C
CF	CF000-CFFFF	19E
D0	D0000-D0FFF	1A0
D1	D1000-D1FFF	1A2
D2	D2000-D2FFF	1A4
D3	D3000-D3FFF	1A6
D4	D4000-D4FFF	1A8
D5	D5000-D5FFF	1AA
D6	D6000-D6FFF	1AC
D7	D7000-D7FFF	1AE
D8	D8000-D8FFF	1B0
D9	D9000-D9FFF	1B2
DA	DA000-DAFFF	1B4
DB	DB000-DBFFF	1B6
DC	DC000-DCFFF	1B6
DD	DD000-DDFFF	1BA
DE	DE000-DEFFF	1BC
DF	DF000-DFFFF	1BE

Table 5-9 (Cont'd): Memory Management Relocation Registers

BLOCK NUMBER	LOGICAL ADDRESS RANGE	ATTRIBUTE AND MAPPING PORT ADDRESS OFFSETS
E0	E0000-E0FFF	1C0
E1	E1000-E1FFF	1C2
E2	E2000-E2FFF	1C4
E3	E3000-E3FFF	1C6
E4	E4000-E4FFF	1C8
E5	E5000-E5FFF	1CA
E6	E6000-E6FFF	1CC
E7	E7000-E7FFF	1CE
E8	E8000-E8FFF	1D0
E9	E9000-E9FFF	1D2
EA	EA000-EAFFF	1D4
EB	EB000-EBFFF	1D6
EC	EC000-ECFFF	1D8
ED	ED000-EDFFF	1DA
EE	EE000-EEFFF	1DC
EF	EF000-EFFFF	1DE
F0	F0000-F0FFF	1E0
F1	F1000-F1FFF	1E2
F2	F2000-F2FFF	1E4
F3	F3000-F3FFF	1E6
F4	F4000-F4FFF	1E8
F5	F5000-F5FFF	1EA
F6	F6000-F6FFF	1EC
F7	F7000-F7FFF	1EE
F8	F8000-F8FFF	1F0
F9	F9000-F9FFF	1F2
FA	FA000-FAFFF	1F4
FB	FB000-FBFFF	1F6
FC	FC000-FCFFF	1F8
FD	FD000-FDFFF	1FA
FE	FE000-FEFFF	1FC
FF	FF000-FFFFF	1FE

Protection Bits

Four bits from the mapper contain the protection code that are sent via the BD Bus to the MMU. These bits are called ALLOWSYWR, ALLOWUSRWR, ALLOWUSRACC and BOSPACE. The first three bits allow system write access, user write access, or user read access to the given 4K page. The last bit is used for a stack warning indicator. If an access takes place within 128 bytes above the bottom of a page so marked then a stack warning is given to the CPU.

The "allow" bits are set by the operating system. When the system is in user mode, the user may be allowed read and write access or neither depending on the accessed page. When the system is in supervisor mode, write access may be denied on some pages.

Violation Detection

The bits in the violation port are defined in Table 5-10. All of these violations are latched and can be read via the Violation Port address offset 2F0 through 2FE. When these violations are detected, they cause an NMI to the processor.

After a violation has occurred, further violations will not produce non-maskable interrupts until the current violation has been cleared. Violations can be cleared by performing an I/O instruction to the control port register 296H.

Normally, only one violation bit will be set in the violation port if an error occurs. However, if another violation occurs before the first violation has been cleared, then the new violation will be detected and latched so that more than one violation bit could be set. No information is given about which violation occurred first. In addition, only the address of the first violation is latched.

Table 5-10 Violation Port Bits

BIT	NAME
0	User Access Violation (read/write/op code fetch)
1	User Write Violation
2	System Write Violation
3	Parity Error
4	End of Stack Warning
5-15	Not Used

SYSTEM CONTROL PORT

The system control decoder port (See Table 5-11) is a decoder PAL (Programable Array Logic) chip which performs some important tasks for the system. This port does the following:

1) Enables the output of an MMU PAL (through the "read violation" command) so that memory cycles can take place.

2) Clocks a flip flop to enable the system to read critical control bits, (See Tables 5-11 and 5-12) at the control bits register.

3) Sets up a timer interrupt to the external PIC, See Table 5-11 for register information.

Table 5-11 Control Port Address Offsets

ADDRESS (HEX)	FUNCTION
292H	READ VIOLATION PORT
296H	CONTROL BITS REG
298H	CLRTIMER INT

Table 5-12 Control Bits

BIT NO.	FUNCTION
3	Clear Parity Error*
4	Request User Mode
7	Clear Violation Port*
14	Clear Memory Management Unit*

*Active low.

8255A PERIPHERAL INTERFACE ADAPTER (PIA)

The 8255A is a three-port general-purpose I/O device that performs the following functions.

1) Interfaces the floppy drive circuitry to the data and LA buses.

2) Provides a non-maskable interrupt (channel attention) signal to the Z-8ØA I/O processor.

3) Supplies maskable interrupts to the 8Ø186 section from various sources.

There are three types of registers within the 8255A, data, control and status along with three 8-bit I/O ports (A, B and C). Port addresses for these registers are supplied in Table 5-13.

Table 5-13 PIA Port Address Offsets

ADDRESS (HEX)	FUNCTION
2DØ	PORT A READ OR WRITE
2D2	PORT B READ OR WRITE
2D4	PORT C READ OR WRITE
2D6	CONTROL/BIT SET REGISTER

Three basic modes of operation may be selected under program control mode Ø--basic I/O; mode 1--strobed I/O; mode 2--bidirectional bus. Port C, under modes 1 and 2 operation, provide control (handshaking) signals for asynchronous operation. Modes 1 and 2 are also supported by interrupt I/O capability.

FLOPPY DISK CONTROLLER OPERATION

The Floppy Disk Controller (FDC) is an LSI chip, the uPD765, which contains the circuitry for interfacing a processor to the disk drive. In this application, it supports a single double-sided, double-density (MFM) 5-1/4 inch drive. The FDC has a built-in data separator.

The FDC shares one of the two DMA channels with the Hard Disk Controller board. Thus the processor need only load the command into the FDC, after which the data transfer occurs under control of the FDC and the 8Ø186. The FDC is capable of multisector transfers.

Fifteen commands can be executed by the FDC, including the basic read, write, scan, format, seek, recalibrate, sense status, and their variations. Details of each command are found in the manufacturer's data handbook.

Programming

Two internal registers in the FDC may be accessed by the 8Ø186 a status register and a data register See Table 5-14 and 5-15. The 8-bit status register may be read at any time. The data register actually consists of a stack of 8-bit registers, only one of which can be latched to the bus at one time.

The track stepping rate, head load time, and head unload time, are not programmable in this application but are hardwired into the controller circuitry.

Table 5-14 Floppy Disk Control Port Address Offsets

ADDRESS (HEX)	FUNCTION
28ØH	MAIN STATUS REG
282H	COMMAND/DATA REG

Table 5-15 Floppy Disk Controller Port Address Offsets

ADDRESS (HEX)	FUNCTION
2EØH	DACK PORT

Z-8ØA I/O CONTROLLER

This section describes the function of the Z-8ØA I/O controller and how it communicates with the 8Ø186 processor.

Functional Description

The Z-8ØA microprocessor performs I/O processing for the five RS232 channels (four user ports and a serial printer port). The I/O processing circuitry includes three Z-8ØA SIO (serial input/ output) devices, two 8253 counters, 4K of RAM and a control ROM for the processor. The 4K of RAM is used for temporary storage and buffering. See Tables 5-16, 5-17 and 5-18.

Table 5-16 Z-8ØA Port Assignments

Z-8Ø ADDRESS	ASSIGNMENT
ØØH - Ø3H	SIOØ
Ø4H - Ø7H	SIO1
Ø8H - ØBH	SIO2
ØCH	RDPERR
1ØH - 13H	TMR1
14H - 17H	TMR2
18H	CONTROL LATCH
1CH	ADR LATCH

Table 5-17 SIOA Port Address Offsets

ADDRESS (HEX)	FUNCTION
2A0H	CH. A DATA
2A2H	CH. B DATA
2A4H	CH. A STATUS/COMMAND
2A6H	CH. B STATUS/COMMAND

Table 5-18 SIOB Port Address Offsets

ADDRESS (HEX)	FUNCTION
2B0H	CH. A DATA
2B2H	CH. B DATA
2B4H	CH. A STATUS/COMMAND
2B6H	CH. B STATUS/COMMAND

Z-80 I/O Processor Memory

The Z-80's 64K of memory space is divided into two halves. The upper 32K is mapped into the system memory and the lower 32K is mapped into the local EPROM and Static RAM. The Z-80 can access the full contents of main memory but must do so in blocks of 32K bytes each. The Z-80 loads the block number register before accessing system memory. When the system memory is accessed, the 5 bit block number is put on the XLA bus address lines 15 through 19. If during the system memory access a parity error occurs, a parity error bit is set and a Z-80A NMI is generated. The NMI should sample the parity bit by reading the PIA and then clear the parity error bit. Table 5-19 defines the Z-80 memory addressing.

Table 5-19 Z-80 Memory Addressing

MEMORY TYPE	Z-80 ADDRESS	
EPROM	00H to 1FFFH	(8K)
STATIC RAM	2000 to 27FFH	(2K)
SYSTEM MEMORY	8000 to FFFFH	(32 blocks)

8Ø186/Z-8ØA Communications

Communication between the 8Ø186 and the Z-8ØA Controller is done through registers. The 24-bit register (initialization register) exists at location 1FFFC (1ØØØ:FFFC). This register contains a pointer to a group of registers. The group of registers may exist anywhere in memory. When the Controller is interrupted, the initialization register is read. From this point, initialization is accomplished according to the parameters in the register groups. The initialization register will be read only after the controller is interrupted, allowing the reserved locations to be reused by the host unless reinitialization is required.

(Note: Multi-byte registers containing 16-bit counts or 24-bit addresses are always stored in reverse order, lo byte first, high byte last.)

Z-8ØA PROGRAMMING

Programming the Z-8ØA I/O Controller is described in nine sections:

Controller Activation
Host/Controller Communication
Controller Initialization
Channel Initialization
Channel Transmit Operation
Channel TTY Receive Operation
Channel Ring Bufferred Receive Operation
Modem Control/Status Operation
Execute from XLA Bus

Controller Activation

After a power-up or controller hardware reset, the controller first initializes all hardware devices, clears internal RAM, and enters a tight loop. To activate the controller, the 8Ø186 must first generate and initialize the system and channel register arrays. (The system register array followed by the eight channel register arrays form a single contiguous block that will hereafter be called the Controller Control Block or CCB.)

The next step is to store the address of the Controller Control Block at XLA bus location 1FFFC hex.

Values must exist at location 1FFFC in the following order:

low mid high xxxx

The first two locations of the Controller Control Block contain the value ØØ hex.

The last step is to generate a "Channel Attention" signal to the controller. This operation will start the controller. When activation is complete, the controller will store the firmware version number (guaranteed not to be zero) in the first location of the Controller Control Block. At this point, the controller is ready to accept commands.

Host/Controller Communication

Commands to the controller are controlled by seven registers within the CCB. The process that reads the command registers is generally in a quiet (not asleep) state. During this time, it will read the "New Command Register". If it finds that this register has not changed, it will exit without checking the system or channel command registers.

When the "New Command Register" has changed, the process will individually test the command registers, and, if there is a command pending, activate the appropriate process to service the request.

This mechanism is very efficient in the general case (when no commands are pending), but does demand the proper sequencing of events. Any time the 80186 issues a command, it must alter the value in the "New Command Register". (The recommended procedure is to increment the value after each command.)

The controller will acknowledge receipt of a command by clearing bit 7 of the command byte. While there is generally no reason to wait for the acknowledgement after issuing a command (see note below), it is essential that bit 7 be tested before issuing a new command. The procedure then becomes the following:

```
Test if last command still pending
<exit or loop if true>
Set new parameters (if required for command)
Issue new command
Increment "New Command Register"
Exit
```

NOTE

> Channel status for a given channel is not valid while a command is pending.

Controller Initialization

After activation, the controller is in an idle state waiting for the system command to enable the controller. Issue the command. At this time, controller interrupts may be enabled.

Channel Initialization

After selecting the desired operating parameters (i.e., parity, word length, etc.), this information must be stored into the channel parameter register. The channel may now be initialized with the "Initialize Channel" command. The channel is now ready to transmit or receive. If any of the interrupt enable bits were set with the command, and their associate conditions true, an interrupt would be generated at this point. At any time, a No-Op command may be issued to effect the interrupt enable bits only.

Channel Transmit Operation

To transmit a block of data, the proper procedure is as follows:

```
Test if last command still pending
<exit or loop if true>
Test if transmitter ready
<exit or loop if false>
Set address and byte count
Issue transmit command
Increment "New Command Register"
Exit
```

Channel TTY Receive Operation

To receive a character in TTY mode, the proper procedure is as follows:

```
Test if last command still pending
<exit or loop if true>
Test if receive character available
<exit or loop if false>
Read status (test for errors)
If Error
Issue Reset Error command
Increment "New Command Register"
Exit
Else
Read character from TTY register
Issue Receiver Ack command
Increment "New Command Register"
Exit
```

Channel Ring Buffered Receive Operation

To receive a character in Ring Bufferred Receive mode, the proper procedure is as follows:

```
Test if last command still pending
<exit or loop if true>
```

```
Test if receive character available
<exit or loop if false>
Read status (test for errors)
If Error
Post status
Issue Reset Error command
Increment "New Command Register"
Exit
Else
Transfer Characters
Issue Receiver Ack command
Increment "New Command Register"
Exit
Transfer Characters:

While (Input Pointer) >< (Output Pointer) Do
{   TTYInputRoutine <- ((BufferAddress)+(OutputPointer));
(OutputPointer) <- (OutputPointer)+1;
If (OutputPointer) = (BufferLength) Then
(OutputPointer) <- Ø;
}
```

Modem Status Operation

Synchronous modem operation is available on Port 3 of the main controller unit rear panel. Asynchronous modem operation may be setup on Ports 1 through Port 5. Refer to the Channel Command Register described in the Register Definition section.

```
Test if last command still pending
<exit or loop if true>
Test if modem status changed
<exit or loop if false>
Read status (test for errors)
Issue Reset Modem Interrupt command
Increment "New Command Register"
Exit

Execute from XLA Bus
```

REGISTER DEFINITIONS

The following provides register definitions for the Z-8ØA I/O controller.

Initialization Register

The Initialization Register is a 24-bit register residing at location 1FFFC (hex). It contains the address of the system and channel.

Control Registers

Firmware Version Register

The firmware version register is normally a read-only register containing the level of the firmware in the Z-8ØA. There is only one case where the host will write into this location. Before system initialization, the host should store a zero into this location (and the System Command register). When initialization is complete, the actual firmware version number will be stored in this location. This location may be tested to identify an uninitialized controller (or initialization in progress). The version numbers will always be in the range of (1-63).(Ø-7) where the version number occupies the five most significant bits, subversion the three least significant bits.

```
|v|v|v|v|v| |s|s|s|
```

Controller Command Register

The command register is an 8-bit register used to pass commands to all channels simultaneously. Handshaking is accomplished by setting bit 7 of the command. The host should insure that the last command has been executed (bit 7 = Ø) before issuing a new command. Commands are:

Ø	disable controller
1	enable controller
2	disable interrupts
3	enable interrupts
4	reset interrupt
5-127	unused

Controller Status Register

The status register is an 8-bit register used to pass status to the host concerning channel-independent information. The status bits are defined as follows:

bit Ø	Controller Enabled
bit 1	Interrupts enabled
bit 2	Interrupt pending (not qualified
by	
	"Interrupts enabled")
bit 3	Bus error
bits 4-7	not used - always Ø

Interrupt Vector Register

The Interrupt Vector Register is a 16-bit register containing data to allow quick response to an interrupting condition. The register is logically divided into three fields.

bits 0-2	interrupting modem channel number
bit 3	modem interrupt
bits 4-6	interrupting receiver channel number
bit 7	receive interrupt
bits 8-10	interrupting transmitter channel number
bit 11	transmit interrupt

New Command Register

The new command register is an eight bit register used to indicate the existence of a new command. This register should be incremented any time a command is written into the system command register, or any of the five channel command registers.

Communication Channel Registers

Channel Parameter Register

The channel parameter register is a 16-bit register containing the following information:

Async mode

bit 0	parity enable		
bit 1	parity even		
bits 2-3	=0		
	=1	1	stop bit
	=2	1.5	stop bits
	=3	2	stop bits
bits 4-5	=0	5	bits per character
	=1	7	bits per character
	=2	6	bits per character
	=3	8	bits per character
bit 6	unused		
bit 7	ring buffer receiver enable/-TTY		
bits 8-11	=0	selectable rate	
	=1	75	bps
	=2	110	bps
	=3	134.5	bps
	=4	150	bps
	=5	300	bps
	=6	600	bps
	=7	1200	bps
	=8	1800	bps
	=9	2000	bps
	=10	2400	bps
	=11	3600	bps

=12	4800	bps
=13	7200	bps
=14	9600	bps
=15	19.2	kbps

bit 12	unused
bit 13	unused
bit 14	CTS control
bit 15	DSR control

Channel Status Register

The channel status register is a 16-bit register containing the following information:

bit 0	transmitter empty
bit 1	Data Terminal Ready
bit 2	not used
bit 3	Request To Send
bit 4	parity error
bit 5	receive data lost (SIO overrun)
bit 6	framing error
bit 7	parity OR overrun OR framing error
bit 8	receive character(s) ready
bit 9	unused - always 0
bit 10	unused - always 0
bit 11	unused - always 0
bit 12	transmitter ready/-active
bit 13	unused - always 0
bit 14	unused - always 0
bit 15	unused - always 0

Channel Command Register

The channel command register is an 8-bit register used to pass commands to the controller. Bit 7 is used to indicate the existence of a valid command. The host should test bit 7 to insure that the last command has been executed. The controller will clear bit 7 after reading the command. At this time, the four least significant bits will also be cleared, while bits 4-6 will retain their values. Bits 4-6 contain the logic level corresponding to the interrupt enable status. The commands available are:

0	no operation
1	initialize channel
2	start transmitter
3	acknowledge receiver
4	abort transmitter
5	reserved
6	"Break" control
7	not used

Channel Command Register (continued)

8	change parameters
9	reset error conditions
1Ø	reset modem interrupt request
11	execute from XLA bus (Port 3 only)
12-15	not used

bit 7	command valid
bit 6	transmit interrupt enable
bit 5	receive interrupt enable
bit 4	modem interrupt enable

Transmit Data Buffer Address Register

The transmit data buffer address register is a 24-bit register containing the address of the data to be transmitted. After tansmit termination, this register will contain the address of the last character transmitted + 1.

Transmit Data Buffer Length Register

The transmit data buffer length register is a 16-bit register containing the number of bytes to be transmitted. After an transmit termination, this register will contain the number of characters in the buffer NOT transmitted. (Note: this value will be Ø unless the transmission was aborted.)

Receive Data Buffer Address Register

The receive data buffer address register is a 24-bit register containing the address of the receive buffer when receiving in ring buffer receive mode.

Receive Data Buffer Length Register

The receive data buffer length register is a 16-bit register containing the length of the receive data buffer. The receive buffer length may be in the range of 2-32768 bytes.

Receive Buffer Input Pointer Register

The receive buffer input pointer register is a 16-bit register containing a pointer (relative to the receive data buffer address) to the first empty location in the receive buffer. As characters are received, the controller will store the character at the specified location, and increment this pointer module buffer length. Bit 15 is the "counter invalid" bit. When this bit is set, the contents of the counter is in the process of being updated. This bit may be ignored if the buffer is =< 255 bytes in length.

Receive Buffer Output Pointer Register

The receive buffer output pointer register is a 16-bit register containing a pointer (relative to the receive data buffer address) to the first full location in the receive buffer. As characters are removed from the buffer, the host will increment this pointer modulo buffer length. When the input pointer is equal to the output pointer, the buffer is empty. Bit 15 is the "counter invalid" bit. When this bit is set, the contents of the counter is in the process of being updated. This bit may be ignored if the buffer is =< 255 bytes in length.

TTY Receive Register

The TTY receive register is an 8-bit register used to receive data in applications not requiring multiple-byte ring-buffering. In TTY receive mode, the byte is simply stored in the TTY receive register.

Selectable Rate Register

The selectable rate register is a 16-bit register used to transfer a 16-bit value directly to the bit rate generator. When the rate is specified through this register, the SIO controller is programmed for clock/16 mode. The value to store in this location can be calculated by the formula:

76800/desired bit rate

Expansion Register

Because some computers/programming languages cannot (or cannot easily) generate/support multiple occurrence arrays containing an odd number of bytes, an 8-bit register is included at the end of each channel register array. This register completes the array of an even number of bytes which may or may not be located on a word boundry. The register is reserved for the purpose of future expansion.

HARD DISK CONTROLLER HARDWARE OVERVIEW

INTRODUCTION

The other major subsystem in the 486 system is the Hard Disk Controller board. This section provides a brief functional description of the circuit board and disk drive and also supplies formatting information.

FUNCTIONAL DESCRIPTION

Figure 5-4 is a block diagram of the hard disk controller. Data enters and leaves the board through connector J1Ø on the CPU board. Bits BDØ through 7 from the external bus are sent to a bus transceiver to create a three-way bus (DØ-7) common to the sector buffers, the WD1Ø1Ø and the external data bus. The sector buffers are a pair of 1K X 4 RAM chips which hold 512 bytes of read or write data. 512 bytes is the size of one sector on the disk.

Disk Drive Description

The 2Ø Mbyte disk drive unit is a half-height random access 5-1/4 inch hard disk drive employing Winchester technology. Data is recorded on two 13Ømm diameter disks through four low force, low mass ferrite heads.

Head Selection

Any of the four heads can be selected by placing the head's binary address on the two Head Select input lines. (Refer to Table 5-2Ø for Head Select 2^0 and Head Select 2^1 pin designations.)

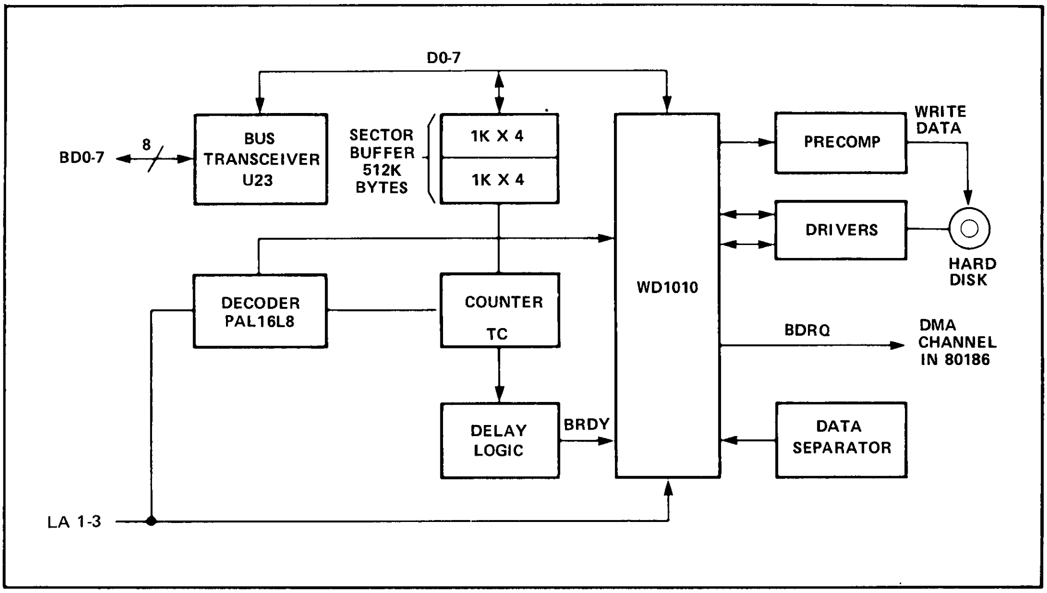


Figure 5-4 Hard Disk Controller Board Circuitry

Table 5-20 Internal Hard Disk Drive Connector

Signal	Ground Return	Signal Name
	BUS	
P1-2	P1-1	Reserved
P1-4	P1-3	Reserved
P1-6	P1-5	-Write Gate
P1-8	P1-7	-Seek Complete
P1-10	P1-9	-Track Zero
P1-12	P1-11	-Write Fault
P1-14	P1-13	-Head Select 2^0
P1-16	P1-15	Reserved
P1-18	P1-17	Head Select 2^1
P1-20	P1-19	-Index
P1-22	P1-21	-Ready
P1-24	P1-23	-Step
P1-26	P1-25	-Drive Select 1
P1-28	P1-27	-Drive Select 2
P1-30	P1-29	-Drive Select 3
P1-32	P1-31	-Drive Select 4
P1-34	P1-33	-Direction In
	RADIAL	
P2-1	P2-2	-Selected
P2-3	P2-4	Reserved
P2-5	P2-6	Spare
P2-7	P2-8	Reserved
P2-9		Spare
P2-10		Spare
P2-11	P2-12	Ground
P2-13		+MFM Write Data
P2-14		-MFM Write Data
P2-15	P2-16	Ground
P2-17		+MFM Read Data
P2-18		-MFM Read Data
P2-19	P2-20	Ground
	POWER	
P3-1		+12 Volts DC
P3-2		+12 Volts DC Return
P3-3		+5 Volts DC Return
P3-4		+5 Volts DC
	STRAP	
P4		Frame Ground

NOTE: These are connectors on the hard disk drive unit not on the Hard Disk Controller board. Refer to Appendix B for Controller board connections.

Formatting Guidelines

The disk drive unit is a soft sectored device that allows the user to define the sector format. When establishing the track format certain rules should be observed to accommodate the physical timing relationships within the drive. Figure 5-5 shows the basic setup of a sector, Table 5-21 shows the disk data format of the WD1Ø1Ø chip, while Table 5-22 provides byte offset conversion information. The Hard Disk port address offset is 32Ø (HEX).

Gap 1

If head switching occurs at index time, Gap 1 must be set to allow the read amplifier to stabilize. The minimum length of Gap 1 is 12 bytes.

Sync

A sync field precedes each addressable record (ID or record). This field should be wide enough to accommoddate the "lock up" characteristics of the phase-lock-loop within the data separator portion of the controller.

Gap 2

Following each sector a gap should be placed to accommodate spindle speed variations between write operations on the same track to prevent overwriting. To accommodate the ±Ø.5% speed tolerance of the disk drive, Gap 2 should be a minimum of 1 byte for each 32 bytes of data within the sector. Additonally, the user should increase the gap to accommoddate the spin speed-asynchronous frequency variation of the controller generated MFM WRITE DATA signals.

Gap 3

This gap is a speed tolerance buffer for the entire track to ensure that the last sector does not overflow beyond the index. Gap 3 precedes index and should be wide enough to accommodate the spin speed variations of the disk drive (±Ø.5%) and the frequency variations of the controller generated MFM WRITE DATA signals. Refer to chapter 2, Specifications for more operating data of the hard disk drive unit.

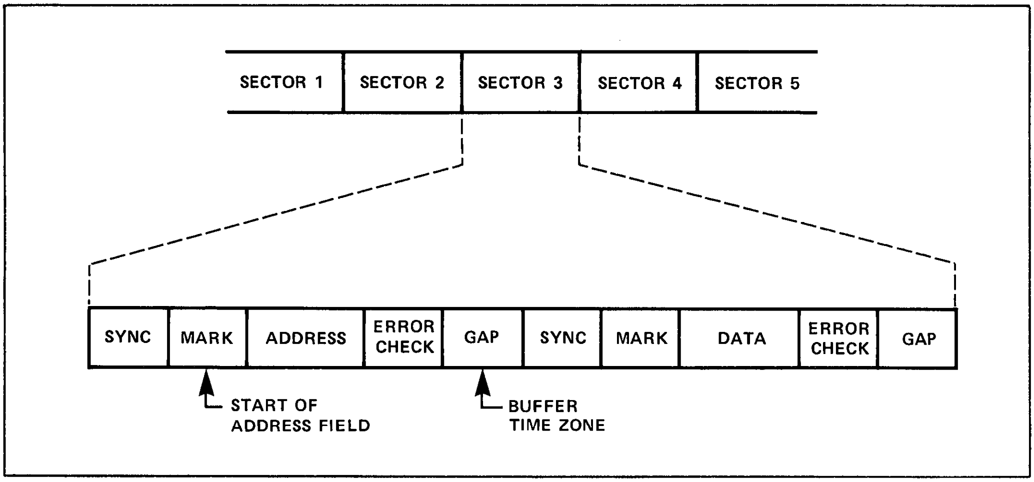


Figure 5-5 Sector Format

Table 5-21 Disk Data Format

No of bytes	Character	
(Start at Index Pulse)		
16	"4E"	
14	"ØØ"	Written out 16 times for 512 byte sectors during format
1	"A1*"	
1	Ident	
1	cylinder byte	
1	head byte	
1	sector byte	
2	CRC bytes	
1Ø	"ØØ"	
12	"ØØ"	Written out during a write operation
1	"A1*"	
1	"FB"	
512	(Data)	
2	CRC bytes	
3	"ØØ"	
(as required)	"4E"	
(Stop at Index Pulse)		

Table 5-22. Conversion from Byte Offset to Sector Number

BYTE OFFSET	ALTOS' SECTOR NUMBER
0- 83	Gap 1 Area
84- 722	Sector 0
723- 1360	Sector 8
1361- 1998	Sector 1
1999- 2636	Sector 9
2637- 3274	Sector 2
3275- 3912	Sector 10
3913- 4550	Sector 3
4551- 5188	Sector 11
5189- 5826	Sector 4
5827- 6464	Sector 12
6465- 7102	Sector 5
7103- 7740	Sector 13
7741- 8378	Sector 6
8379- 9016	Sector 14
9017- 9654	Sector 7
9655-10292	Sector 15

Firmware Interface 6

PAGE SUBJECT

TABLES

INTRODUCTION

This chapter describes the firmware contents of the 486 PROM chips, specifically the system monitor programs and the power-up tests.

SYSTEM MONITOR

The system monitor is a collection of programs and modules that reside in EPROM. The monitor provides initial system checks and exercises the initial coordination and control of the system.

The first function of the monitor is to initialize system hardware. Thereafter, the Monitor loads in (auto-boot) the operating system from the hard disk into the computer's memory. The Monitor then provides the option to interrupt auto-boot in order to select from a menu of other loading sources. These other sources include booting from a floppy disk, booting from WorkNet or entering the Debugger. In addition the Monitor:

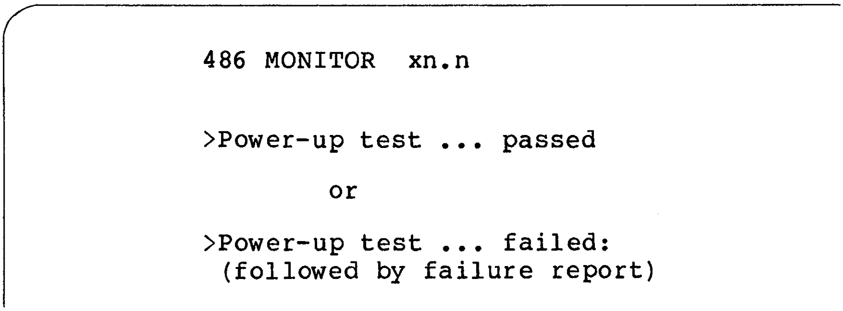
- performs system "power-up" diagnostics tests (described later in this chapter). These tests exercise the major components in the system.

- displays system configuration and power-up test reports.

- provides low-level drivers for floppy diskette, hard disk, WorkNet, and RS232 I/O ports for keyboard and CRT.

- provides a system Debugger program.

The sequence of events in the Monitor program is given below. The results of each step is shown on the following initial display screen.



```
486 MONITOR  xn.n


>Power-up test ... passed

        or

>Power-up test ... failed:
 (followed by failure report)
```

Upon power-up, the 486 Monitor performs the following operations:

- Runs power-up diagnostic tests to verify that all major hardware devices are functioning correctly.

  Under normal circumstances, the system beeps once to indicate that the power-up test has started. A few seconds later a second beep indicates that the system passed the power-up test.

  If the system fails the power-up test, the monitor prints an error code on the screen.

  NOTE

  The error code will not print on the screen if a CRT failure occurs.

- Displays or prints the system configuration (memory size, number of disk drives, and network capabilities).

- Displays a message informing you that auto booting may be interrupted.

  If you do not respond within three seconds, auto-booting begins. The priority sequence for automatic boot is

  1) hard disk
  2) floppy disk
  3) WorkNet

  If you interrupt the automatic boot, a boot menu appears on the screen. From this menu you can boot from any available source or device.

  If either the automatic or user-selected boot fails, the Monitor displays an error message or messages, and passes system control to the Debugger program.

  When boot up is successful, the Monitor passes control to the newly loaded code, which normally is the bootstrap program for the operating system of the diagnostic programs.

Monitor Location

The various Monitor modules are located at fixed memory locations. Table 6-1 shows the address range of memory occupied by each major module.

Table 6-1 Monitor Memory Map

Memory	
Locations	Contents
ØØ4ØØ - ØØ7FF	Monitor Work Area
ØØFFE - Down	Monitor Stack Area
FCØØØ - FFFBF	Monitor's Program
FFFCØ - FFFDF	Monitor's "Copy-Right" Statement
FFFEØ - FFFEB	Monitor's Bootstrap Program

Debugger Program

The Debugger is used to debug application programs using the following available commands (Table 6-2) in conjunction with the proper command syntax (Table 6-3).

Table 6-2 Debugger Support Commands

Commands	Meaning
A	ALTER MEMORY
B	BREAKPOINT
C	CHANGE REGISTER
D	DISPLAY MEMORY
F	FILL MEMORY
G	GO
H	HELP
I	INPUT PORT
J	FLOPPY DEBUG COMMAND
K	HARD DISK DEBUG COMMAND
L	HEX DOWNLOAD
M	MOVE MEMORY
O	OUTPUT PORT
RF/RH	READ FROM FLOPPY/HARD DISK
S	SINGLE STEP
T	SWITCH TERMINAL
	NOTE: This function should not be executed by the user because it will "hang up" the system and require reinitialization. This program command is for manufacturer's use only, as it requires additional cabling not supplied with the system.
WF/WH	READ FROM FLOPPY/HARD DISK
ZØ/Z1	Z8Ø COMMAND (See note above.)
<ESC>	Repeat the previous request
<CONTROL-RESET>	Restart Monitor's power-up sequence

Entering the Debugger Program

The Debugger program is started by interrupting the automatic booting procedure and selecting option 4 from the "boot menu". See screen below:

```
Select [1] to boot from Hard Disk
       [2] to boot from Floppy Disk
       [3] to boot from WorkNet
       [4] to enter Debugger

> Enter option:
```

Upon entering option 4. The following display is shown:

```
>486 DEBUGGER   xn.n
 cs:ip xxxx:yyyy
```

The first two letters cs, ip, refer to the registers of the 80186 microprocessor. See the 80186 data book for more information on these register sets. The letters x and y refer to the hexadecimal contents of the registers.

Using Debugger Commands (An example).

After the debugger program is selected, enter a command (Table 6-1), and the command syntax (Table 6-3) as required. See example below.

```
>g or G <CR>
```

See GO command for definition of this function.

Table 6-3 Debugger's Command Syntax

NOTES

'-' denotes a 'space' in the command line.

Unless indicated otherwise, all values are specified in Hex)

<CR> denotes a "Carriage Return".

Upper or lower case letters are accepted.

All memory addresses are four hex digit segments and a four hex digit offset.

All I/O addresses are four hex digits long.

ALTER MEMORY: This command allows the user to change the memory contents beginning with the given address.

```
Command line syntax:  a_ssss:oooo_yy_yy_. . .<CR>

                   a:  Alter command.
                ssss:  Segment Number
                oooo:  Beginning memory offset address to
                       be altered.
                  yy:  Hex byte values.
               . . .:  Up to 20 bytes at a time.
```

DISPLAY/CHANGE/CLEAR BREAKPOINT: This command allows the user to either view, change, or clear the current breakpoint address.

```
Command line syntax:  b<CR>   [Display breakpoint address]

         b_ssss:oooo<CR>   [Set breakpoint to a memory address]

                   bc<CR>   [Clear all breakpoints 1 and 2]

                  bc1<CR>   [Clear breakpoint 1]

                  bc2<CR>   [Clear breakpoint 2]

                       b:  Breakpoint command.
                    ssss:  Segment Number.
                    oooo:  Breakpoint offset address.
```

Table 6-3 Debugger's Command Syntax

DISPLAY/CHANGE REGISTER CONTENTS: This command allows the user to either view, or change current register contents.

Command line syntax: c<CR> [Display all register contents]

cxx_yyyy [Change particular register]

c: Register command.
xx: Only one register of following is allowed: (ax, bx, cx, dx, sp, bp, si, di, cs, ds, ip, fl)
yyyy: Word value.

DISPLAY MEMORY CONTENTS: This command displays the memory contents starting with the given address. It will display at least sixteen bytes, or one page at a time.

Command line syntax: d_ssss:oooo_yyyy<CR> [Display Memory]

d: Display memory command.
ssss: Segment Number
oooo: Beginning memory offset to be displayed.
yyyy: Byte count. (module 16)

example:

d_ssss:oooo<CR>: Display one line (16 bytes) of memory data.

d_ssss:oooo_ff<CR>: Display one screen full of memory data.

d_ssss:oooo_ffff<CR>: Display the entire 64Kb of memory data in a screen-full matter. The Monitor displays one screen-full of data and pauses. Pressing the "spacebar" will continue to display another screenfull of memory data, however, entering any other keys completes the command and returns the user back to debugger. The display will also continue on to next segment, if the boundary is crossed.

Table 6-3 Debugger's Command Syntax

FILL MEMORY: This command allows the user to fill a range of memory with the desired byte pattern.

Command line syntax: f-xxxxx-yyyy-zz-<CR> [Fill Memory]

```
        f:  Fill memory command.
    xxxxx:  Beginning memory address to be filled.
     yyyy:  Byte count.
       zz:  Hex byte value to be written.
```

GO: This command allows the user to start executing program based on the values in the code segment (cs) and instruction pointer (ip) registers.

Command line syntax: g<CR> [Go/start execution]

```
        g:  Go command.
```

HELP: This command displays the debugger's menu and its required syntaxes.

Command line syntax: h<CR> [Display Debugger Menu]

INPUT PORT: This command allows the user to read in the word value of the port, designated in the given address.

Command line syntax: i_xxxx<CR> [Input from System Port]

```
        i:  Input port command
     xxxx:  Port address
```

FLOPPY I/O: This command allows the user to communicate directly with the floppy controller chip (NEC uD765).

The user is required to enter the right amount of parameter bytes for a particular command operation. (Refer to the NEC uD765 specification for more information).

Upon receiving a carriage return, the Monitor passes the command byte as well as all the parameter to the controller. After the command is finished, the status bytes are displayed on the screen in the same format as in the controller specification. Also, the starting address of the disk buffer during read/write command is defaulted to location 10000 (hex).

Table 6-3 Debugger's Command Syntax

Command line syntax: j_aa_bb_cc-dd_ee-ff-gg-hh-ii<CR> [Floppy]

j: Floppy I/O command.
aa: Command byte required by Floppy Disk Controller.
bb..ii: Fifo parameters (number of parameters required depending on the actual command)

(Default floppy disk buffer area begins at 10000h)

example:

j-08<CR>: 08 Denotes a "sense interrupt status" command with no additional parameter required.

HARD DISK I/O: This command allows the user to communicate directly to the hard disk controller ship (Western Digital 1010). Furthermore, the user is required to enter the right amount of parameter byte for a particular command operation. (refer to Western Digital 1010 specification for more information)

Upon receiving a carriage return, the Monitor passes the command byte as well as all the parameter to the controller. After the command is finished, status bytes will be displayed on the screen in the same format as in the controller specification. Also, the starting address of the disk buffer during read/write command is default to location 04000 (hex).

Command line syntax: k-aa-bb-cc-dd-ee-ff<CR> [Hard Disk]

k: Hard disk I/O command.
aa: Command byte for Hard disk controller (WD 1010).
bb: Sector/Drive/Head (SDH) byte.
cc: Cylinder low byte.
dd: Cylinder low byte.
ee: Beginning sector number.
ff: Sector Count.

(Default hard disk buffer area begins at 04000h)

example:

k-10-20<CR>: 10 Denotes a "recal" command.
20 Denotes a 512K byte sector size: a drive number 0: and head number 0.

Table 6-3 Debugger's Command Syntax

```
RS232 DOWNLOAD:  This command allows the user to download the
                 INTEL hex formatted data from Z8Ø port number 1
                 to port number 5.  Option is provided to start
                 executing the downloaded code as soon as the
                 download operation is completed.  User will be
                 asked which port is to be downloaded from.

Command line syntax:  1<CR>: [Download from Z8Ø port]

                     1X<CR>: [Download from port, then execute]

                          1:  Download from Z8Ø port command.
                          X:  Execute code as soon as the
                              download operation is completed.
```

```
MOVE MEMORY:  This command allows the user to relocate a block of
              memory data to where desired.

Command line syntax:  m-xxxxx-yyyyy-zzzz-<CR>  [Move Memory]

                     m:  Move memory command.
                 xxxxx:  Source address.
                 yyyyy:  Destination address.
                  zzzz:  Byte count.
```

```
OUTPUT PORT:  This command allows the user to output a word value
              to the port designated by the given address.

Command line syntax:  o-xxxx-yyyy<CR>  [Output to Port]

                     o:  Output port command.
                  xxxx:  I/O port address.
                  yyyy:  Word value to be written.
```

```
READ FROM FLOPPY/HARD DISK:  This command allows the user to read
                             in block(s) of data from the floppy
                             disk or hard disk.

Command line syntax:  rf-ssss:oooo-aa-bc-dd-ee<CR>
                      [Read from Floppy]

                  ssss:  Segment number
                  oooo:  Beginning memory offset where disk data
                         is to be stored.
                    rf:  Read floppy command.
                    aa:  Track number. (in hex)
                     b:  Drive number. (Ø--3)
                     c:  Head number. (Ø-1)
                    dd:  Beginning sector number. (note: floppy
```

Table 6-3 Debugger's Command Syntax

```
                         start at sector number 1)
                    ee:  (optional ) Sector Count (default to be
                         1 sector)
Command line syntax:  rh-ssss:oooo-fg-hh-ii-jj-kk<CR>
                      [Read from Hard Disk]

                    rh:  Read hard disk command
                    fg:  Head number (Ø-3)
                    hh:  Cylinder (high byte)
                    ii:  Cylinder (low byte)
                    jj:  Beginning sector number
                    kk:  (optional) Sector Count (default to be
                         1 sector)
-----------------------------------------------------------------

SINGLE STEP:  This command allows the user to execute one
              instruction, pointed by code segment (cs) and
              instruction pointer (ip) registers, then return
              to debugger.

Command line syntax:  s<CR>  [single step]
-----------------------------------------------------------------

SWITCH TERMINAL:  This command allows the user to switch a
                  terminal to or from a port 1 (main user) con-
                  sole to or from a diagnostics console.

                  This function should not be executed by the
                  user because it will hang up the system.  See
                  NOTE on page 6-4 for additional information.

Command line syntax:   t<CR>   [switch terminal]
------------------------------------------------------------------

WRITE FROM DEVICE:  This command allows the user to write in
                    blocks of data to either floppy or hard disk.
                    The command syntax is the same as "Read from
                    device" command.
-----------------------------------------------------------------

Z8Ø DEBUG COMMAND:  This command sends a sign-on message to Z8Ø
                    port number 1 from Diagnostic Console.  This
                    function should not be executed by user
                    because it will hang up the system.  See NOTE
                    on page 6-4.

Command line syntax:  ZØ<CR> [Reset Z8Ø]
                      Z1<CR> [Awake Z8Ø and attempt to send
                              message to port number 1]
-----------------------------------------------------------------
MISC SUPPORTS:  <ESC>  [Repeat the previous debugger command]
      <Control-Reset>  [Restart Monitor's power-up sequence]
```

486 Monitor's System Call Interface

You can call up the Monitor utility programs by entering the system call: "INT 2Ø (dec)" with a Function Code in register <BL>, and if required a Parameter Block Pointer in <DS:AX>. (See Table 6-4).

NOTE

"input" - Refers to the parameters required by the PROMs. You must supply it before typing the INT 2Ø request.

"output"- Is the result returned to you from the PROM.

The following Monitor system calls are supported. See Table 6-4.

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description

Ø1	con_in	Read in an ascii character from the keyboard console
		(Note: This routine waits until a character is typed on the keyboard)
		[input: bl-Ø1; int 2Ø]
		[dx = Ø = Z8Ø's channel Ø;
		dx = 1 = Z8Ø,s channel 1;
		dx = 2 = Z8Ø's channel 2;
		dx = 3 = Z8Ø's channel 3;
		dx = Off = Diag Console (8274)]
		[output: al=ascii character]

Ø2	con_out	Write one character to the console [input: al=ascii character; bl-Ø2; dx has the same assignment as con_in, int 2Ø] [output: none]
Ø3	con_str	Write a character string to the console [input: ds:ax = character array pointer ending with a "Ø"] [bl-Ø3; int 2Ø; dx has the same assignment as con_in] [output: none]

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description
11	con_stat	Read in the keyboard status [input: bl=11; int 20 ;dx has the same assignment as in con_in] [output: al = 0 means no character; al = 1 means character available] (Note: The "con-stat" and "con-in" calls should be prepared together. You should first call "con-stat" to check if there is a character available then call a "con-in" to read this character. See "con-in" above.
12	auto_reboot	Reboots the system from the device requested and also restarts the monitor operation, while skipping over the power-up tests [input: bl=12; int 20 ;bh = device number] [bh = 0 = default auto_boot] [output: none]
13	boot_number	Returns the boot-device number [refer to the power-up test descriptions for device assignment] [input: bl=13; int 20]

Table 6-4 Monitor System Calls

Function Code (dec)	System Call	Function Description
3Ø	general I/O	Performs the peripheral (including Floppy, Hard Disk, Network, tape, etc.) operation specified by the I/O parameter block defined in the following protocol. [input: bl=3Ø; int 2Ø; ds:ax=pointer to parameter block] [output: resultant status return in the pre-defined locations of the same parameter block]

Table 6-5 Peripheral I/O Parameter Block Layout

Relative Byte Address	Byte Content
ØØØ(H)	[Device Number (low byte)]
ØØ1(H)	[Device Number (high byte)]
ØØ2(H)	[Command (low byte)]
ØØ3(H)	[Command (high byte)]
ØØ4(H)	[Result (low byte)]
ØØ5(H)	[Result (high byte)]
ØØ6(H)	[Device Result (Status byte 1)]
ØØ7(H)	[Device Result (Status byte 2)]
ØØ8(H)	[Device Result (Status byte 3)]
ØØ9(H)	[Device Result (Status byte 4)]
ØØA(H)	[DMA Segment (low byte)]
ØØB(H)	[DMA Segment (high byte)]
ØØC(H)	[DMA Offset (low byte)]
ØØD(H)	[DMA Offset (high byte)]
ØØE(H)	[Cylinder (low byte)]
ØØF(H)	[Cylinder (high byte)]
Ø1Ø(H)	[Drive Number]
Ø11(H)	[Head Number]
Ø12(H)	[Beginning Sector Number]
Ø13(H)	[Sector Count Number]
Ø14(H)	[Sector length]
Ø15(H)	[Retries]
Ø16(H)	

Total Block Size = 22 (dec) bytes

Refer to Disk I/O Parameter Block Definition

Table 6-6 Peripheral I/O Parameter Block Definition

		Disk Only			
No.	Device	No.	Command	No.	Status
0	Hard Disk	0	Recal	0	No Error
1	Floppy Disk	1	Write Sector	1	General Error
2	SCSI (Not Used)	2	Read Sector	2	Device not supported
3	Tape	4	Seek	3	Device not present
4	Printer	5	Format	4	Invalid command
5	WorkNet			5	Timeout error
6	EtherNet(Not Used)				
7	Console				
8	Auto Boot				
9	586 Floppy I/O (Not Used)				

Res_1 to Res_4 will have the device's results.

(Refer to individual device driver's specification. However, a non-zero normally indicated a failure condition)

The beginning sector byte is used to specify the gap length size during hard disk and floppy formats. (Note: typically, a value of 50 (hex) is used. In both cases, the format requires the DMA buffer to be filled according to the format command specifications command required by the controller chip. (Note: Refer to individual device driver's specification).

Power-up Test

Upon power-up, or system reset, the Monitor performs a sequence of diagnostic tests (up to eighteen) to verify the system's overall integrity.

Table 6-7 Power-up Test Structure

Power-up Test Number (dec)	Component Tested
Ø	Monitor Prom (Low byte)
1	Monitor Prom (High byte)
2	Map Ram Data Lines
3	Map Ram Address Lines
4	Map Ram Contents
5	Main Ram Data Lines
6	Main Ram Address Lines
7	Parallel (8255) Port
8	Z8Ø Interface test
9	Z8Ø Channel test
1Ø	Main Ram Contents
11	Interrupt (8259) Controller
12	Internal CPU Timer
13	Internal CPU PIC
14	Internal CPU DMA
15	Diag Console (8274) Controller
16	Floppy (765) Controller
17	Memory Parity

Functional verification begins with Test Number Ø and progresses on to Test Number 17 using the following algorithms:

A) The seventeen power-up tests can be divided into group 1 and group 2. Group 1 includes Test Number Ø through Test Number 7. These tests are more critical to the system integrity than those in group 2 (8-17).

B) As soon as a test in group 1 (Ø - 7) fails, the Monitor skips over the remaining tests, continues the normal boot attempt, and reports the error status. (note: since this failure is considered catastrophic, further tests have no significant meaning).

C) Tests in group 2 (8 - 17) are executed regardless of any previous test status.

Power-up test failure reporting is described in the next section.

Power-up Test Error Reporting Scheme

If a failure is detected during power-up diagnostic tests, the ACS 486 Monitor reports error(s) in following ways:

A) The two LED's on the CPU board flash the Test Number that failed.

NOTE

> The left (GREEN) LED, viewing from the front of the CPU board is the synchronizer, while the right (RED) LED pulses the Test Number in a five-pulse binary sequence beginning with the most significant bit.)

example:

Left-Green LED	on	on	on	on	on
Right-Red LED	off	off	on	on	on
Binary Code	Ø	Ø	1	1	1

Test Number = 7 = Memory content test

B) During system configuration reporting, the monitor reports all the failed components(s) on SIO (8274) port number 1 and the Diagnostic Console.

Power-up Test Descriptions

Ø,1) Prom Checksum Test - First, the low byte PROM of the Monitor PROMs is verified with the checksum. Then, the high byte is verified.

By adding all bytes in a single Monitor PROM, a total sum of zero value indicates that the corresponding Monitor PROM has the proper contents.

2) Map Ram Data Lines Test - The test sends a single "1" bit across the data lines to check for a short-circuit among the data lines.

The first location of map ram memory (I/O port Number ØØØØ) is used.

3) Map Ram Address Lines Test - This test sends data across the lower 8 address lines of the map ram memory to check for a short-circuit among the address lines.

The first 8 locations of map ram memory are used. Two data patterns of Number ØØØØ and Number ffff(hex) are also used.

4) Map Ram Content Test - This test writes and reads the entire 256 locations of map ram, using two data patterns (Number ØØØØ and Number ffff) to check for a short circuit to ground or +5 Volt.

5) Main Ram Data Lines Test - The test sends a single "1" bit across the data lines to check for a short circuit among the data lines.

6) Main Ram Address Lines Test - This test sends data across the address lines of main memory to check for a short circuit among the address lines.

The first eight locations of main ram memory are used. Two data patterns of Number ØØØØ and Number ffff (hex) are also used.

6.1) Small-Block of Main Memory Test - The first 4K bytes of memory and locations 1fffc to 1ffff are written with data value of Ø's and check for memory integrities. These areas are used as a communication block to awake the Z8Ø to print out a sign-on message through its port number 1.

7) Parallel Port (8255) Test - Two patterns of ØØØØ and ffff (hex) are written/read back to verify all three ports on 8255.

8) Z8Ø Initialization Test - First, the Z8Ø parameter block is set up with its pointer in memory location 1fffc (a designated area for Z8Ø communication), then generates a channel attention to awakening the Z8Ø controller.

The Z8Ø returns its firmware version number in the first word of the Z8Ø parameter block. If this operation is successful, the Z8Ø's ability to accept initialization command from host CPU is tested. The time-out loop will be set up if Z8Ø fails to clear the command pending flag.

9) Z8Ø Channel Test - Attempts to send a "Channel Initialization" command to Z8Ø's port number 1 after all the required parameters are set up in the Z8Ø's channel control block.

If successful, a wake-up bell and sign-on message will be sent to the Z8Ø's port number 1. This will serve as a wake-up call and to verify the Z8Ø's port number 1 interface.

10) Main Ram Content Test - This test verifies the main memory contents,using two data patterns (Number ØØØØ and Number ffff) to check for a short-circuit to ground or +5 Volt.

The following algorithm is used in this memory test:

A) Write a data pattern of 5555(hex) to the first location of a segment, then read back the data pattern.

   If the data read back is 5555(hex), the segment is valid. The entire segment (64K bytes) is then tested with two data patterns (Number ØØØØ and Number ffff).

   If the data read back is ffff(hex), the entire segment is not there. The test then checks to see if the missing segment is within the basic memory range (segment Ø to 3).

      If so, an error exists since there can not be a gap within the basic memory range.

      If the missing segment is outside the basic memory range, the test continues on.

   If the data read back is anything other than 5555(hex) or ffff(hex), there are bad bits in this segments.

B) Increment the segment counter and check it in the following ways:

   If it is less than segment 7, go to step A.

   If it is larger than segment D, the memory test is completed.

   If it is equal to segment 8, the test checks the wrap-around by writing the first location of segment 8 with a pattern of aaaa(hex) and checks the first location on segment Ø.

      If the first location of segment Ø is aaaa(hex), a wrap-around occured, and the test is completed.

      If the first location remains as 5555(hex), the Monitor returns to step A and continues on to test segments 8 to F (up to the Monitor's area).

11) Interrupt Controller (8259) Test - Two data patterns (Number ØØØØ and Number ffff) are used in a static test to write and read to the system interrupt controller (8259). These two patterns detect possible short-circuit with ground and + 5 Volt.

If no error is detected in the static test, a dynamic test using the actual interrupt capability will be tested in the Internal CPU's Interrupt Test.

12) Internal CPU's Timer Test - Two data patterns (Number ØØØØ and Number ffff) are used in a static test to write and read to all three internal 186's timer registers. These two patterns will detect any possible short-circuits with ground and +5 Volt.

If no error is detected in the static test, a dynamic test will be conducted. In the dynamic test, all three timers are first programmed with a maximum count of 256. Then, all three timers are started and stopped in a very short time. At this point all three count registers are checked to see if they have ever been incremented. If so, the CPU's timers are working.

NOTE

The timer's interrupt capablity is tested in the next test.

13) Internal CPU's Interrupt Test - This test verifies the interrupt capability of the CPU-8Ø186.

First, all CPU interrupt vectors are set to an error code, except timer Ø's interrupt. Next, the CPU's interrupt controller and timer Ø's interrupt service routine are both programmed for timer Ø's interrupt. Then timer Ø is started and the program enters a wait loop. If timer Ø interrupts and the correct service routine is entered, this test is passed, otherwise, if the wait loop is exhausted or there is no response from the service routine, an error is reported.

14) Internal CPU's DMA Test - This test verifies the DMA capability of the CPU-8Ø186.

Since there are as many as 1Ø8 possible combinations of the CPU's DMA transfer modes, only six general tests are performed in this test. They are as follow:

1) Word transfer with both source and destination pointers incremented.*

2) Byte transfer with both source and destination pointers incremented.*

3) Word transfer with both source and destination pointers remaining the same.*

*Refer also to DMA source and destination register offsets in Chapter 5.

4) Byte transfer with both source and destination pointers remaining the same.*

5) Word transfer with both source and destination pointers decremented.*

6) Byte transfer with both source and destination pointers decrement.*

*Refer also to DMA source and destination register offsets in Chapter 5.

In each case, a background pattern is written in the destination area. Also, a wait loop is used to check the CPU's DMA counting ability.

An error is reported if either the wait-time expires or the background pattern remains unchanged.

15) Diag-console Controller (8274) test - Two patterns (number 0000 and number ffff (hex)) are used to check for the RS232 interface. The port address is at serial I/O channel A-12a6 (hex).

16) Floppy Controller (765) Test - Since sending a command to the floppy controller requires several read and write operations, the Monitor verifies the floppy interface by sending a specify command. This specify command tells the controller about the drive's characteristics. If the controller accepts the command, this test is passed.

17) Memory Parity Test - Two patterns of Number 0000 (even parity) and Number fefe (hex) (odd parity) are used to check the memory parity generating logic.

First, the NMI vector, parity service routine, and parity generating circuit are programmed and enabled. Next, location Number 7000(hex) of every segment is written and read back with the data patterns. If there is a parity error, the parity service routine sets the error flag.

Appendix A
486 CPU Jumpering Specifications

APPENDIX A

486 CPU JUMPERING SPECIFICATION

In the 486 jumpers are named by block number and pin numbers. For example the CTS jumper in E1 is named E1-1,2 which means that a strap is placed between pins 1 and 2 of E1.

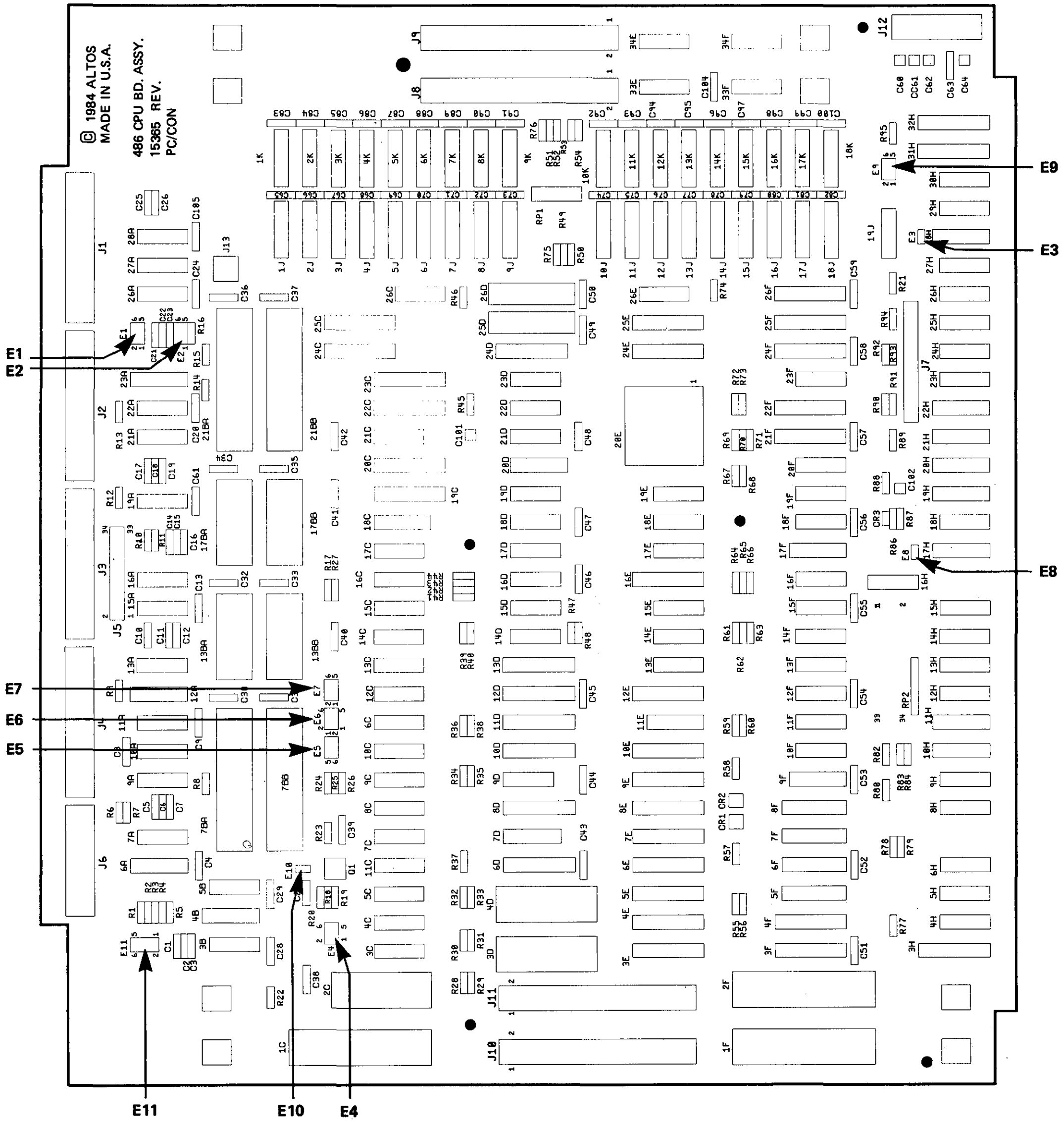


Figure A-1 486 CPU Jumper Locations

E1-1,2 Internally loops the RTS and CTS signals on the SIO port 2.

E1-1,3 Connects CTS on the port header to the SIO port 2.

E1-2,4 Connects RTS on the port header to the SIO port 2.

E2 Functional description is the same as for jumper location E1 but is for SIO port 1.

E3-1,2 Connects 19.66Ø8 Mhz baud rate crystal to divide by sixteen counter. Must be present for proper operation.

E4-1,2 Internally loops the RTS and CTS signals on the diagnostic SIO port.

E4-1,3 Connects CTS on the diagnostic port header to the SIO.

E4-2,4 Connects RTS on the diagnostic port header to the SIO.

E5 Functional description is the same as for jumper location E1 but is for SIO port 4.

E6 Functional description is the same as for jumper location E1 but is for SIO port 3.

E7 Functional description is the same as for jumper location E1 but is for SIO port 5.

E8-1,2 Connects 32 Mhz oscillator to divide by two circuit to run the 186 CPU. Must be present for proper operation.

E9-1,2 Creates a flag for the BANKSEL 1 pal so that either 128K or 256K bit rams can be used in the memory expansion slots J8 and J9.

E1Ø-1,2 Connects the 4 Mhz Z8Ø clock. Must be present for proper operation of the Z8Ø.

Appendix B
Port Connections

APPENDIX B

PORT CONNECTIONS

RS232 SERIAL PORTS J1,J2,J4,J5 PIN OUTS

PIN	SIGNAL
1	PROTGND
2	TXD*
3	RXD
4	RTS
5	CTS
6	DSR
7	GND
8	CARDET
9	NC
10	NC
11	NC
12	NC
13	NC
14	NC
15	NC
16	
17	NC
18	NC
19	NC
20	DTR
21	NC
22	NC
23	NC
24	NC
25	NC

RS232 SERIAL PORT J3 PIN OUTS

PIN	SIGNAL	MODEM PIN OUT
1	PROTGND	1
2	TXD*	3
3	RXD*	2
4	RTS	5
5	CTS	4
6	DSR	20
7	GND	7
8	CARDET	
9	NC	
10	NC	
11	NC	
12	NC	
13	NC	
14	NC	
15	TX TIMING	
16	NC	
17	RX TIMING	
18	NC	
19	NC	
20	DTR	6
21	NC	
22	NC	
23	NC	
24	NC	
25	NC	

RS232 SERIAL PORT J1 AND J2 PIN OUTS

```
CONNECTOR CPU               TI 810 (SERIAL) TERMINAL
CPU J1-J2                     CONNECTOR

1----CHASSIS GND--------------1
2----TXD----------------------2
3----RXD----------------------3
                              6----------|
7----SIGNAL GND---------------7          |
                              8----------|
20---DTR----------------------11         |  JUMPER
                              20---------|
```

```
CONNECTOR CPU             EPSON RX-80 (SERIAL) TERMINAL
CPU J1-J2                     CONNECTOR

1----CHASSIS GND--------------1
3----RXD----------------------3
7----SIGNAL GND---------------7
20---DTR----------------------11
```

```
CONNECTOR CPU             EPSON MX-80 F/T (SERIAL) TERMINAL
CPU J1-J2                     CONNECTOR

1----CHASSIS GND--------------1
2----TXD----------------------2
3----RXD----------------------3
                              4--|
                                 |  JUMPER
                              5--|
                              6----|
7----SIGNAL GND---------------7    |
                              19---|  JUMPER
20----DTR---------------------20
```

WORKNET PORT J6 PORT PIN OUTS

PIN	SIGNAL
1	NC
2	ANETD
3	NC
4	ANETCLK
5	NC
6	NC
7	NC
8	NC
9	NC
1Ø	ANETD*
11	NC
12	ANETCLK*
13	NC
14	NC
15	NC

The Figure B-1 is provided to identify the connecting ports between the Hard Disk Drive , CPU, and the Hard Disk Drive Controller listed on the following pages.

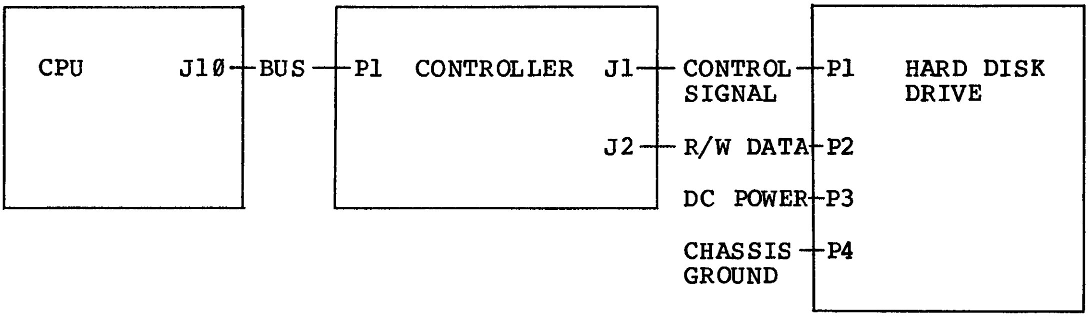


Figure B-1 Hard Disk Drive

HARD DISK DRIVE TO HARD DISK DRIVE CONTROLLER READ/WRITE SIGNALS

```
CONTROLLER 1Ø PIN        TO        HARD DISK DRIVE 2Ø PIN
CONNECTOR J2                       CONNECTOR P2

1     (NOT USED)                   1 (NOT USED)
2                                  2
3---(+MFM WRDATA)------------------3
4---(-MFM WRDATA)------------------4
5                                  5
6                                  6
7---(+MFM RDDATA)------------------7
8---(-MFM RDDATA)------------------8
9                                  9
1Ø    (NOT USED)                   1Ø ----|
                                   11     |
                                   12     |
                                   13     |
                                   14     |
                                   15     |---   (NOT USED)
                                   16     |
                                   17     |
                                   18     |
                                   19     |
                                   2Ø ----|
```

HARD DISK DRIVE TO HARD DISK DRIVE CONTROLLER
CONTROL SIGNALS

CONTROLLER 34 PIN CONNECTOR J1	TO	HARD DISK DRIVE 34 PIN CONNECTOR P1
1	GROUND	1
2	LOW CURRENT*	2
3	GROUND	3
4	HEAD SEL 2*	4
5	GROUND	5
6	WR GATE*	6
7	GROUND	7
8	SEEK COMPLETE*	8
9	GROUND	9
1Ø	TRACK ØØ*	1Ø
11	GROUND	11
12	WR FAULT*	12
13	GROUND	13
14	HEAD SEL Ø*	14
15	RESERVED	15
16	RESERVED	16
17	GROUND	17
18	HEAD SEL 1*	18
19	GROUND	19
2Ø	INDEX*	2Ø
21	GROUND	21
22	READY*	22
23	GROUND	23
24	STEP*	24
25	GROUND	25
26	DRIVE SEL Ø*	26
27	GROUND	27
28		28
29	GROUND	29
3Ø		3Ø
31	GROUND	31
32		32
33	GROUND	33
34	DIR*	34

CPU I/O EXPANSION PORT TO HARD DISK DRIVE CONTROLLER BUS CONNECTIONS

```
I/O EXPANSION                  TO         HARD DISK DRIVE CONTROLLER
CONNECTOR J1Ø                             CONNECTOR P1
1                                         1
2                                         2
3                                         3
4                                         4
5                                         5
6                                         6
7                                         7
8                                         8
9                                         9
1Ø---MPWR*--------------------------------1Ø
11                                        11
12---MPRD*--------------------------------12
13                                        13
14                                        14
15---LA4----------------------------------15
16                                        16
17---LA2----------------------------------17
18                                        18
19---LA1----------------------------------19
2Ø                                        2Ø
21---LA3----------------------------------21
22                                        22
23                                        23
24                                        24
25                                        25
26                                        26
27---BCPUCLK*-----------------------------27
28                                        28
29                                        29
3Ø                                        3Ø
31---BDT/R*-------------------------------31
32---BD7----------------------------------32
33                                        33
34---BD6----------------------------------34
35                                        35
36---BD5----------------------------------36
37---INTRQ*-------------------------------37
38---BD4----------------------------------38
39---SRESET*------------------------------39
4Ø---BD3----------------------------------4Ø
41---SLTSELØ*-----------------------------41
42---BD2----------------------------------42
43                                        43
44---BD1----------------------------------44
45---SLTDMARQ*----------------------------45
46---BDØ----------------------------------46
47                                        47
48                                        48
49                                        49
5Ø                                        5Ø
```

CPU PORT TO FLOPPY DISK DRIVE

```
CPU                     TO         FLOPPY DISK DRIVE
CONNECTOR J7                       CONNECTOR P7

1----GROUND-------------------------1
2----HDLOAD*------------------------2
3----GROUND-------------------------3
4----INUSE*-------------------------4
5----GROUND-------------------------5
6----DRIVE SELECT 3*
7----GROUND-------------------------7
8----INDEX*-------------------------8
9----GROUND-------------------------9
1Ø---DRIVE SELECT Ø*----------------1Ø
11---GROUND-------------------------11
12---DRIVE SELECT 1*----------------12
13---GROUND-------------------------13
14---DRIVE SELECT 2*----------------14
15---GROUND-------------------------15
16---MOTORON*-----------------------16
17---GROUND-------------------------17
18---DIRECTION SELECT*--------------18
19---GROUND-------------------------19
2Ø---STEP*--------------------------2Ø
21---GROUND-------------------------21
22---WRITE DATA*--------------------22
23---GROUND-------------------------23
24---WRITE GATE*--------------------24
25---GROUND-------------------------25
26---TRACK ØØ*----------------------26
27---GROUND-------------------------27
28---WRITE PROTECT*-----------------28
29---GROUND-------------------------29
3Ø---READ DATA*---------------------3Ø
31---GROUND-------------------------31
32---SIDE ONE SELECT*---------------32
33---GROUND-------------------------33
34---READY*-------------------------34
```

Index

ALTOS 486 SYSTEM REFERENCE

READER COMMENT FORM

Altos Computer Systems
2641 Orchard Park Way
San Jose, CA 95134

This document has been prepared for use with your Altos Computer System. Should you find any errors or problems in the manual, or have any suggestions for improvement, please return this form to the ALTOS PUBLICATIONS DEPARTMENT. Do include page numbers or section numbers, where applicable.

____

____

____

____

____

____

____

____

____

____

____

____

System Model Number____________________

Serial Number________________

Document Title____

Revision Number________________________Date______________________

Name____

Company Name____

Address____

____

____

ALTOS

Printed in U.S.A.
P/N 690-15707-001

2641 Orchard Park Way, San Jose, California 95134
(408) 946-6700, Telex 470642 Alto UI

November 1984